# 3 UX and UX Design

## Syllabus

*The Wheel : UX Processes, Lifecycle, Methods and Techniques, Scope, rigor, complexity and Project perspective, Agile lifecycle Processes and the Funnel model of Agile UX.*

## Contents

## 3.1 UX and UX Design

### 3.1.1 Expanding Concept of Interaction

- Interaction was only beginning to be more than how people used computers in the first version of this book. In the context of UX, the concept of interaction has evolved from a human and a computer working together to achieve a goal to, as shown in Fig. 3.1.1, a very broad phrase referring to a wide range of communication and collaboration between a human and an artifact in an ecology.
- **Artifact of interaction :** Interaction artifact is a gadget, service, instrument, mechanism, item, or environment that can communicate with humans in both directions. The building or room you are in, the chair you are sitting in, kitchens, an ATM, an elevator, equipment like refrigerators, cars and other vehicles, most types of signs, dwellings, the DMV's workflow, and voting machines are all examples of artifacts.

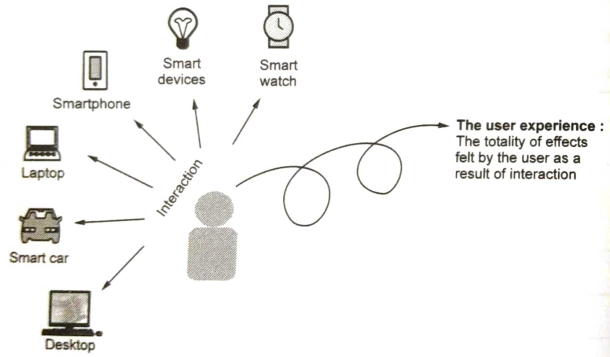


**Fig. 3.1.1 Human-artifact interaction and user experience**

- The nature of engagement has evolved as well, not simply the variants of gadgets. For example, 25 years ago, most interaction took place on desktop computers, either at home or at work. The primary means of interaction was through a keyboard, mouse, and monitor. Interaction was then expanded to handheld devices such as cell phones and pagers. The former enabled rudimentary tasks to be completed via highly modal interaction. When someone was paged, the latter enabled an interrupt-driven engagement.





- With the introduction of PDAs, this progressed to utilizing a stylus. Learning a new input format using the PDA glyph language or using an on-screen small keyboard that could be tapped with the stylus was required for the interaction. Then there were touch cellphones and tablets, which rendered the stylus obsolete. Now, there's a smart watch that combines touch and a scroll wheel to change the nature of interaction. Then there are smart glasses, virtual reality goggles, and smart living environments, which allow users to interact in new ways.
- Good UX design will inevitably have a significant positive impact on our lives because interaction is so broad and ubiquitous in our world.

### 3.1.2 Definition of UX

#### 3.1.2.1 Distinction from "UI"

- People used to talk about "UI" when they meant "user interface," which usually meant "user interface software." In general terms, the User Interface (UI) is the software medium that underpins interaction, and therefore is of little interest in our context.
- UX design encompasses interface design as well as much more, but not UI software.
- According to some sources, UI design is more about visual design, whereas UX design is more about interaction design. You could argue that the user interface is one of the gateways through which users engage, and that developing it entails a variety of sub-disciplines.
- The appearance, feel, and emotional components of a specific UI are frequently seen as the responsibility of a visual designer in the literature. An interaction designer is responsible for the organization of tasks on the UI and how they are supported in connection to other tasks supported by the UIs of other devices in the ecosystem.
- Software engineers are incharge of the software that implements those specifications. In other words, UIs are various ecological portals.
- However, in the general public, the phrases UI, HCI, and UX are used interchangeably.

#### 3.1.2.2 Distinction from "HCI"

- Along similar lines, the term "HCI" refers to the entire field of study and stands for "human-computer interaction." This phrase is now largely used in academic contexts, such as research and development, whereas "UX" is the more common term for HCI practise in the field.

### 3.1.2.3 What Does "UX" Mean ?

- We need to go right to the point regarding what UX is because this book is clearly about UX. The two letters "UX" are a popular abbreviation that stands for "user experience". Those two letters represent the entire practise, all of the work done in this field, and the final user experience that results from it.
- The multidisciplinary nature of UX has led to many definitions from multiple angles, including UX as theory, UX as a phenomena, UX as a field of study, and UX as a practise.

### 3.1.2.4 Rise of UX

- Early large mainframe computers of the preconsumer era were used to run large enterprise software systems, and users were trained to use a system for specific business purposes. "Interaction" was via punch cards, paper tape, and paper printouts, so there really were no system-development considerations of usability or UX.
- The personal computer then brought computing to the desk of a corporate user, while the consumer movement brought computers into people's homes. Customer service and support were the first to notice that opening up the market to "mere mortals" without a thorough understanding of how the product was used had a significant influence on support expenses.
- Smart devices and the Internet have made computing accessible to everyone.
- It is now possible for businesses to communicate directly with their customers. The paradigm shift users no longer need training to use a system; instead, the system must match their needs. The path to usability, HCI, and UX was unavoidable due to user expectations.
- Digital natives today regard computing as something that simply exists; they are unaware that the product is designed. They just assume it will work.

### 3.1.2.5 What is User Experience ?

- Of course, user experience is a type of experience, and "experience is a very dynamic, complicated, and subjective phenomenon", with the context of the related activity playing a big role.
- The sum of the impacts perceived by a user before, during, and after engagement with a product or system in an ecology is referred to as user experience.
- As UX designers, it is our goal to design that interaction in such a way that the user experience is productive, fulfilling, satisfactory, and even enjoyable.





- The following are key qualities of a user experience that are reflected in the definition above :
  1. It is a result of interaction, whether direct or indirect.
  2. It is about the totality of the effects.
  3. It is felt internally by a user.
  4. It includes usage context and ecology.

#### Interaction, direct or indirect

- The interaction between a human and a designed artifact can be direct (e.g., controlling a device and receiving feedback) or indirect (e.g., interacting with a device and receiving feedback).

#### Totality of effects

- Following up on the second characteristic of a user experience, the effects of interaction include the user's entire "stream of perceptions, interpretations of those perceptions, and resulting emotions during an encounter with a system," according to the Dagstuhl report.
- The entirety of interaction effects includes :
  - The impact of usability, utility, and emotional impact on physical interactions.
  - The complete progression of effects over time.
- Consider a potential user investigating a product or system, viewing advertisements and reviews, and anticipating ownership as an example of effects perceived over time.
- Product packaging and the "out of the box" experience; seeing, touching, and thinking about the object; admiring the product, using it, and retaining and savouring (or not) the joy of usage are all effects that occur after the product is purchased.
- Finally, the user experience can encompass an individual's feelings about the company that created the product or system, as well as its reputation and branding, as well as pride of ownership and how the product has come to mean something in the user's life, extending into a broader cultural and personal experience.

**User experience is felt internally by the user**

- Obviously, the user is the one who has the experience. As a result, even when users interact under the identical settings, their experiences can differ.

**Context and ecology are crucial to user experience**

- An ecosystem refers to the entire usage context, which includes all parts of the world with which the user interacts. Multiple ecologies are available to the user (e.g., work versus home). Multiple unique usage contexts may exist within an ecological (e.g., stressful work conditions or pleasurable play conditions). And each of these circumstances has an impact on the user's experience.

## 3.1.3 UX Design

### 3.1.3.1 Can a User Experience be Designed ?

- A keen reader may have already picked up on a minor inconsistency. We have used the terms "UX design" and "creating a user experience," while others refer to "developing a user experience." However, you cannot design something that happens inside a user's head. So terms like "UX design" are a little confusing, but we trust you will recognize that this entails creating a positive user experience.

### 3.1.3.2 Importance of UX Design

- The importance of UX is becoming more widely recognized and UX design has taken center stage. As a senior VP of IBM said, "There's no longer any real distinction between business strategy and the design of the user experience". Knemeyer agrees, by saying, "User Experience (UX) has become a mission-critical consideration for companies in every industry, and of every shape and size."
- The high cost of improper UX design is one technique to emphasise the necessity of effective UX design. Poor UXdesign in the architecture of buildings and living environments, for example, might result in long-term expenses.
- "Too often, those who plan and create buildings and parks are unconcerned about whether or not they will function effectively or how much they will cost to operate. They can go on to the next project after the current one is completed. However, the public is forced to live with poorly constructed and designed buildings and spaces, and taxpayers are frequently forced to foot the tab for putting them right."
- Bad user interface / user experience design costs a lot of money and, more significantly, lives. Traffic accidents, injuries, and even death can occur as a result of distractions caused by poor UX designs for driving cars.





- The same caution applies to the design of user interfaces for flying planes and ships at sea. The crash of Egypt Air Flight 990 in 1999 was an example of this. Poor usability in the design of cockpit controls was determined to be the culprit. And According to reports, the collision of the USS McCain was caused by poor guidance console UX design.
- The requirement for strong UX design in the medical domain is arguably even more pressing in terms of the effects of safety in day-to-day operations. "A field research revealed 22 instances in which automated hospital systems can result in the erroneous medication being delivered to patients," according to Nielsen 3. The majority of these faults are standard usability issues that have been known for decades."

## 3.1.4 Components of UX

- User experience, as shown in Fig. 3.1.2, is a mix of the following factors :
  1. Usability.
  2. Usefulness.
  3. Emotional impact.
  4. Meaningfulness.

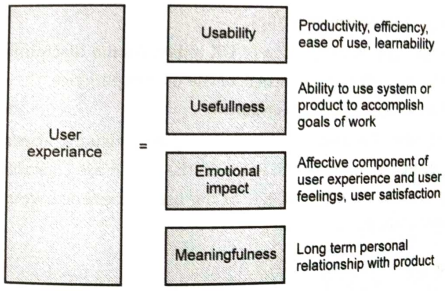


**Fig. 3.1.2 Components of UX**

### 3.1.4.1 An Analogy with Fine Dining

- We utilize the realm of fine dining to show the many components of user experience. A meal's utility might be measured in terms of nutritional value, or it can be felt in terms of health benefits.
- Practical issues can also play a role in determining usability in a dining experience.

For example, if the food given is harsh and difficult to cut or chew, or if it takes too long to prepare and serve, the usability aspect of the eating experience would undoubtedly suffer. Dining usefulness will also be hampered by a lack of basic eating utensils.

- However, for many of us, eating is primarily an emotional experience. Perhaps it all begins with the thrill of anticipation. The dining environment, lighting, background music, and décor, as well as the quality of service, aesthetics of food presentation, friendliness of the staff, and how well the food tasted, will all affect the diners' perceptions and emotional responses.

### 3.1.4.2 Usability

- Long ago, usability was all that mattered in the subject of Human-Computer Interaction (HCI), the umbrella academic discipline for UX, which included :
  - Ease of use.
  - User performance and productivity.
  - Efficiency.
  - Error avoidance.
  - Learnability.
  - Retainability (ease of remembering).
- Usability is still a critical component of UX today. As the discipline has been more focused on the more glamorous aspects of the user experience, the core component of usability has sometimes been overlooked.
- The trendy so-called flat design style, for example, is visually appealing but lacks a key affordance that indicates which items on the screen are clickable and which are not. It's unlikely that the other aspects of the user experience would be addressed without decent usability.

### 3.1.4.3 Usefulness

- The second factor is usefulness, which is often overlooked in the user experience. Utility is usefulness.
- The strength and functionality of the backend software that allows you to get work done is referred to as usefulness. It is the true motivation behind a product or system. Usability and usefulness are defined as meeting a user's do goals, such as checking email or leaving a comment on Facebook.





### 3.1.4.4 Emotional Impact

- Emotional impact is the third component, which is an affective aspect of the user experience. Emotional impact, as the name implies, encompasses how users feel emotionally about an interaction, including user pleasure.
- Despite the fact that there have been previous scholarly publications on emotion in the user experience, Norman (2002) was one of the first to bring the topic to the attention of the general public. It is on a large size, and it is related to his topic of everyday things.
- There are now conferences dedicated to the subject, such as the biennial Conference on Design and Emotion, whose objective is to promote a cross-disciplinary approach to design and emotion.
- While every aspect of the user experience is emotional because it is felt internally by the user, there are a few aspects of the user experience that are purely emotional, factors that are felt up close and personal during the use of technology, factors that take the user beyond simple satisfaction to fun, enjoyment, and self-expression, with sometimes strong emotional consequences.
- Emotional impact can manifest itself in a variety of ways, including :
  - Joy of usage.
  - Pleasure.
  - Excitement.
  - Fun.
  - Curiosity.
  - Aesthetics.
  - Novelty.
  - Surprise.
  - Delight.
  - Play.
  - Exploration.
  - Coolness.
  - Appeal.
  - A sense of identity.
  - Happiness.
  - Enthusiasm.

- Enticement.
- Engagement.
- Pride of ownership.
- Affinity, attractiveness, identifying with a product.
- "Wow" in UX design.

**Why include emotional impact ?**

- Users are no longer happy with merely the efficiency and effectiveness of usability; they are also searching for emotional fulfillment. Norman puts it like way : "beautiful things make people feel happy."
- Users now seek enjoyment in product use and aesthetics in product design, and the items we possess and use can evoke significant sentiments of importance and social status, especially if the object is high-tech and esoteric.
- Emotional impact in interactions can have a good effect on economics and job performance; happy emotions can lead to improved job satisfaction, decision-making, and other behaviours (Zhang & Li, 2005). Positive emotions, as Norman (2004) demonstrates, can have a significant impact on learning, curiosity, and creative ideas.

**Deeper emotions**

- While the majority of the emotional impact elements are related to pleasure, they can also be related to other types of emotions, such as love, hate, fear, grief, and reminiscing about shared memories.
- Social engagement and interaction for cultural problem solving are examples of applications where emotional impact is significant.
- Emotional components of social and cultural relationships include trustworthiness and credibility. In sites like CaringBridge and CarePages, for example, design for emotional impact can also mean encouraging human compassion.

**Joy, excitement, and fun**

- "The humanistic concept that happiness is vital to existence is the most important reason for considering joy of use."
- We've taken an example to show how emotional impact affects the outcome. A mountain bike is depicted in Fig. 3.1.3.





- This bike is waiting for you to climb on and embark on a thrilling adventure. However, this image does not depict the adventure, which is the user experience.
- Compare that to the next illustration in Fig. 3.1.4 doesnot even depict the entire user or even the entire bike.

**Fig. 3.1.3 Mountain bike**

**Fig. 3.1.4 Mountain bike experience**

- However, it succeeds in capturing the thrill of the user experience. The dynamic water spray portrays the fun and excitement of the game. As you careen over the uneven rocks, your blood and adrenaline are rushing, and the panorama rushes past in a whirl. That is what you are purchasing: the exhilarating excitement of riding the bike.

**Attractive designs somehow work better**

- For many users, a pleasing design simply works better and makes them feel better. It's similar to getting your new automobile washed and cleaned - it seems to run better afterward.

**Engagement and enticement**

- Churchill (2010) defines engagement as "flow," "fascination," and "involvement." The psychological idea of flow requires complete participation, intense focus, and the exclusion of all activities except the central one.
- Engagement can extend beyond usage episodes and contribute to long-term value. Enticement is a trait that attracts the user's attention.

**Coolness and "wow" in UX design**

- Consumers nowadays are accustomed to, and even expect, high-quality items. In the user experience, coolness and "wow" are becoming "necessary" characteristics of emotional impact.

**Role of branding, marketing, and corporate culture**

- In some circumstances, the impacts of usability, usefulness, and joy of use outweigh the effects of usability, usefulness, and joy of use. Users can become engrossed in the entire environment of the company, their political affiliations, how the product is presented, and so forth.
- What image does a product's brand represent ? Is it a company that employs ecologically friendly manufacturing methods ? Is it true that they recycle ? As a result, what does the fact that someone uses a particular brand's product say about them ? These variables are difficult to define in the abstract and even more difficult to locate in the real world.
- Take Apple in the late 2000s and early 2010s, for example. Everything they produced had a stamp of subtle elegance and amazing design because the culture of creating for user experience was so firmly ingrained in their corporate culture.
- Apple's obsession with providing a great user experience extends beyond their products and into other aspects of the firm. When they presented a job offer to a new employee, for example, the package arrived in a carefully designed envelope that set the tone for the company's values.
- This atmosphere infiltrated Apple's retail outlets as well. "Not only has the corporation made many of its stores feel like gathering places, but the bright lighting and similarly bright acoustics create a buzz that makes shoppers feel more like they are at an event than a retail store," according to a New York Times article.
- One new business in Manhattan aimed to be "the most personalised store ever established." This well-thought-out user experience has resulted in increased sales, repeat visits, and even tourist pilgrimages.





### 3.1.4.5 Meaningfulness

- While usability and, in some cases, emotional effect are normally concerned with a single usage incidence, meaningfulness is concerned with how a product or artifact becomes meaningful in the life of a user, as seen in Fig. 3.1.2. Meaning is derived from a long-term personal interaction between the product and its human user.
- It is exemplified by many people's feelings of attachment to their cellphones, to the point where they become physically uncomfortable if they are separated from them. A hiker's sense of comfort and safety in reaction to a hand-held GPS is an example of meaningfulness.
- The academic idea of phenomenology is strongly tied to the concept of meaning.

### 3.1.5 What UX is Not ?

- While user experience is becoming a more established aspect of the technological sector, there are still certain misconceptions and mischaracterizations.

#### 3.1.5.1 Not Dummy Proofing or User Friendliness

- Usability and UX are not the same as dummy or idiot testing. While these expressions may have been slightly amusing the first time they were used by people who did not know anything about usability, they are today derogatory and humiliating to both users and designers.
- Usability and UX, on the other hand, aren't about being "user-friendly." This is a misnomer that reduces UX design to a cliche. Users are not searching for friendliness; they're looking for a tool that will help them achieve their objectives that is efficient, effective, safe, and possibly beautiful and enjoyable.

#### 3.1.5.2 Not Just About Dressing Things Up in a Pretty Skin

- Another common misperception regarding early usability and human factors experts was that they were the ones who dressed up and "made it pretty" the design at the end. "Design means veneer in most people's vocabularies," as Steve Jobs put it. It's all about interior design. The curtains and sofa are made of the same fabric. But nothing, in my opinion, could be further from the definition of design.
- Adding UX design as a "spread-on" layer at the end is what we call the "peanut butter theory" of UX, because it appears to be founded on the idea that, once the product has been produced, you can spread this wonderful thin layer of UX all over it. It does not take much knowledge of UX or software these days to figure out that this will not work.

### 3.1.5.3 Not Just a Diagnostic View

- Many businesses had big software engineering teams, as well as a small pool of human factors experts, who were loaned out to project teams for a short period of time, generally near the end of the project, and were expected to do "usability testing." Many people mistakenly believe that "doing usability" is the same as "doing usability testing," which we term the "priest in a parachute" approach.
- The human factors people were supposed to drop down into the project once the team had pretty much committed to the design, give it their blessing, and then leave! There is not enough time to remedy noncosmetic issues discovered at this point.
- Before the product must be shipped, there are no resources left to invest in it.

## 3.1.6 Kinds of Interaction and UX

- Not every interaction between a user and a GUI is for a specific task, such as adding an item to a calendar. Some interactions persist throughout a variety of temporal and spatial states, as well as distinct contexts.
- An interaction can be about a transaction that spans a system and extends into a series of exchanges and encounters over a long period of time, rather than just a single transaction.
- We've identified a few different types of interaction that we can link to various types of user experiences.
  1. Localized interaction.
  2. Activity - based interaction.
  3. System - spanning interaction.

### 3.1.6.1 Localized Interaction

- Localized interaction is localized with respect to both time and system. It's a basic contact with a single "product," a single gadget in the user's ecology that surrounds them.
- It's task-oriented, confined, and limited, and it happens in a very short amount of time within a single interaction setting and with a particular aim, such as checking your email on your laptop or withdrawing cash from an ATM.
- As a result, interactivity is at the forefront of design.





### 3.1.6.2 Activity-Based Interaction

- Norman coined the term "activity-based design" to characterise interactions that are more complex than simple tasks. One or more task threads make up an activity, which is a collection of multiple, overlapping, and linked tasks. It may entail :
  - Interaction with one device to do a set of related tasks.
  - Interaction across devices in the user's ecology.
- Interaction with one device to do a set of related tasks
  - Let's say you are looking for a small digital camera on the internet. You may click on links to reviews, choose one, and add it to your "cart," then click on connections to related products.
  - You can also click on additional links to find accessories. Despite the fact that this comprises a variety of tasks, people see it as a single activity.
- Norman defines communication devices as "mobile phones that integrate appointment books, diaries, and calendars, note-taking capabilities, text messaging, and cameras." "This one device can do everything: look up phone numbers, dial, converse, take notes, check one's diary or calendar, and exchange photos, text messages, and emails." The many tasks can then be integrated into a single overall activity.
- Interaction across devices in the user's ecology.

### 3.1.6.3 System-Spanning Interaction

- Interaction across devices in the user's ecology
  - System-spanning engagement is a type of activity-based interaction that frequently involves several people in different roles, multiple devices, and multiple places.
  - **Example : Power Lines are Down**
- Here's an example of a transaction with a straightforward goal : to restore electric power service to a user who discovers that his home's electricity has gone out.
- Our user starts the activity by calling his neighbour (Fig. 3.1.5). The neighbour claims that his electricity is out as well, and that a power line is down in the area. Our hero then contacts the power company's customer care department to see if the problem can be resolved. The customer care representative enters an account number, client name, phone number, and address into a queue in a central database.

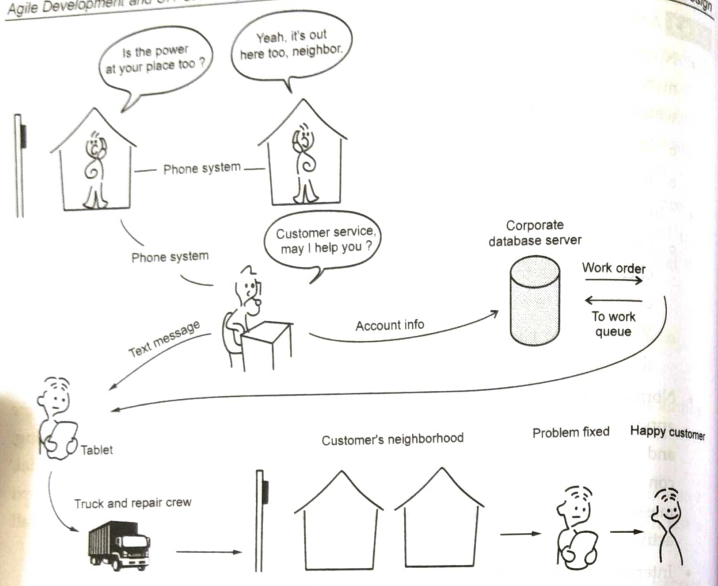


**Fig. 3.1.5 System - spanning activity**

- Customer support also sends a text message to a field technician, who checks the queue on his tablet, picks up the work order, jumps in his truck, and travels to that neighbourhood. He and his workers repair the electricity lines, and he documents the work on his tablet. The consumer is satisfied.
- You can see an ecology involved in one user activity from this one simple example, which includes the user as customer, the power company, its customer service, customer accounts, the power company's central database, a work order queue, a work order, power company field technicians, and the power lines.
- The telephone system, a neighbour on his phone, a text message, the technician's portable tablet, the technician's crew, and a fleet of power company trucks are all part of the ecology of this activity.
- The user's workflow is continuous throughout the ecology, and information pervasiveness, or the capacity to share and access information throughout the entire ecology, is the glue that holds the many subsystems together.





### 3.1.6.4 The Dagstuhl Framework of Interaction and UX

- Participants in the Dagstuhl seminar (Roto et al., 2011, p. 8) modelled different types of UX (and different types of interaction) in terms of time. They begin with the earliest and include :
  - **Before usage :** Anticipated UX.
  - **During usage :** Momentary UX, one-off encounters.
  - **Post-usage :** Episodic UX, with periods of usage mixed with periods of non-usage.
  - **Over time :** Cumulative UX, or overall views of a system after using it for a while.
- These various types of UX overlap and validate our definition of the totality of effects, depending on temporal durations. Feelings created by researching a product, reading reviews, and so on are examples of anticipated UX. Our localized, activity-based, system-spanning, and long-term interactions cross in varied ways with their immediate, episodic, and cumulative UX.
- Users' assessments of systems they use frequently, such as a laptop, desktop PC, operating system, or word processor, were given weight in their overall UX. Our long-term relationship spans and extends beyond their total user experience. We term it meaningfulness if the total UX is positive.

### 3.1.7 Service Experience

- UX has particular applications in service experience and service design. According to Forlizzi, UX design for a user or customer experience within a "transactional path" is service design.
- Although Forlizzi attempts to differentiate the two, we believe her description only affirms that service design is UX design applied to a customer journey : it is "transactional" and aids "a consumer in achieving a goal."
- Applying the principles we've discussed in this book to a customer's experience of buying something or receiving assistance is what service experience is all about. It's about the customer's user experience journey's touch points.
- It usually entails a storytelling tale of a UX experience that is spread out over time and often across multiple locales. A patient's service experience at a hospital for elective surgery, for example, could include their arrival, checking in and going through lines, being processed, and so on.
- The customer journey is a simplified version of the primary path. Deviations, edge cases, malfunctions, pinch spots, and other issues are also encountered by users.

- **Example :** Presurgery hospital visits
- Consider the following description of how one local health institution arranged the workflow of its surgery patient service.
- It all started with a visit to the family doctor, where the symptoms were investigated and a provisional diagnosis was made. Further tests were ordered by the doctor in order to properly diagnose the problem.
- The next trip was the local hospital, where X-rays and an MRI were taken, with the final conclusion being that surgery was required.
- The patient then proceeded to the nearest large city's hospital, where the surgery would take place. The patient had to follow driving directions he had received via email to get there. There was a pretty complicated set of parking, walking, and admittance directions when I arrived.
- The patient was then processed by a group of professionals who were experts in various aspects of preadmission and preparation. There was also a preop appointment at a later time.
- The surgery was simple for the patient, and no instructions were required. However, postsurgical care necessitated a series of prescriptions and trips to the pharmacy to fill them, as well as phone calls to the hospital's surgery team with queries concerning recuperation. Several follow-up appointments with the surgeon were required, which were gradually reduced to follow-up appointments with the family physician.

## 3.2 Wheel : UX Processes, Lifecycles, Methods and Techniques

### 3.2.1 INTRODUCTION

#### 3.2.1.1 Where are we Heading ?

- It's vital to remember that, this is all about UX design, not at all about software. We create a user experience design, which is usually presented as a prototype. Developers, software engineers, and programmers will implement these designs in software using a software engineering lifecycle.

#### 3.2.1.2 Need for Process

- Long ago, software engineers realised that having a process is essential for designing complex systems, and they put a lot of effort into defining, verifying, and following it.





- Wixon and Whiteside, who worked at Digital Equipment Corp in the 1980s, were ahead of their time when they said :

"Building usability into a system takes more than a basic understanding of what works. For finding problems and answers, more than an empirical method is required. It will take more than senior management's backing and openness on the part of all system developers. It necessitates more than just money and time. An explicit engineering method is required to include usability into a product. That engineering procedure is not logically distinct from any other procedure. It entails empirical definition, level specification, proper methodologies, early delivery of a functional system, and the readiness to update that system. These concepts, when combined, transform usability from a "last-minute add-on" to an essential component of product development. We can only hope to regularly deliver products in which usability is more than an advertising claim if usability engineering is as much a part of software development as scheduling."

- Practitioners are forced to make it up as they go along without the assistance of a UX design approach. You are not alone if this has happened in your initiatives. Without a method, an approach will be idiosyncratic. Practitioners' actions will be guided and constrained by their personal experiences.
- They will focus on their own preferred methods of doing things, while other critical process activities will fall through the cracks. Finally, as Holtzblatt puts it, adopting a product development process is a good way to avoid "an organization's unrelenting push to ship 'something' by a particular date."

#### 3.2.1.3 What do You Get by having a Process ?

- **A guiding structure is a process**
  - A process is a framework that guides both beginners and specialists through the complexities of a project. A process imposes a systematic approach, providing order to what could otherwise be a chaotic situation, particularly in a large and complex undertaking.
  - Process serves as a guide to guarantee that inexperienced designers produce high-quality work and progress toward becoming specialists. A process serves as a checklist for professionals, ensuring that they do not overlook any critical components of the problem in the midst of their work. It assists designers in addressing issues such as "Where are we now ?" and "What can / should we do next ?"

- **Process offers reliability and consistency**
  - A written method allows you to take the same approach from project to project and team member to team member.
- **Process provides scaffolding for learning**
  - Learning is at the heart of design. A process provides a framework on which you can construct a knowledge base of what you've learned, using organizational memory from previous similar attempts to incorporate past learning. This fabric, in turn, aids in the training of new designers in the techniques of UX at that business or in the field as a whole.
- **Process provides a shared conception of what you are doing.**
  - A documented process explains how a product or system is created. Process also aids team coordination and communication by externalizing the stage of development for observation, measurement, analysis, and control; otherwise, communication among project roles about what they are doing is difficult due to a lack of shared understanding of what they should be doing.

## 3.2.2 Basic Process Components for UX

### 3.2.2.1 UX Design Lifecycle

- A lifecycle is exactly what it sounds like : it's a cycle of a UX design's life, from conception to implementation and beyond.

### 3.2.2.2 UX Lifecycle Activities

- The high-level things you accomplish during a lifecycle (Fig. 3.2.1) are called lifecycle activities :
  - Understand Needs (of users).
  - Design Solutions.
  - Prototype Candidates (for promising designs).
  - Evaluate UX.

### 3.2.2.3 UX Design Lifecycle Process

- The UX lifecycle process is a depiction of how you organize the lifecycle activities in order over time and how the lifecycle activities - the boxes in Fig. 3.2.1 are related in the process flow, which is commonly represented as a flowchart diagram. We do not make fine distinctions here, therefore we use the phrases "process," "lifecycle," and "lifecycle process" interchangeably.





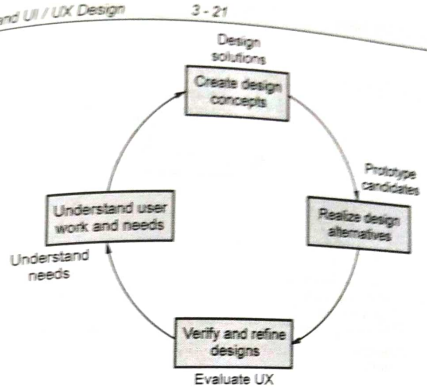


**Fig. 3.2.1 UX design lifecycle**

### 3.2.2.4 Wheel : A Model of the UX Lifecycle

- We get a kind of UX lifecycle template of Fig. 3.2.2 if we expand this abstract cycle a little to include feedback and iteration, which we term "the Wheel" as an analogy. This is because it rotates in circles, bringing you closer to your destination with each round.
- This fundamental diagram is the template for practically any design process; it applies whether the scope of the design is a small piece of a product or the entire system.

### 3.2.2.5 Lifecycle Subactivities

- Each action in the lifecycle is large enough to be broken down into its own set of sub-activities. The things you undertake during a single lifecycle activity are called lifecycle subactivities.
- The following are some examples of sub actions for the understand needs lifecycle activity :
  - Data elicitation.
  - Data analysis.
  - Data modeling.
  - Requirements extraction.

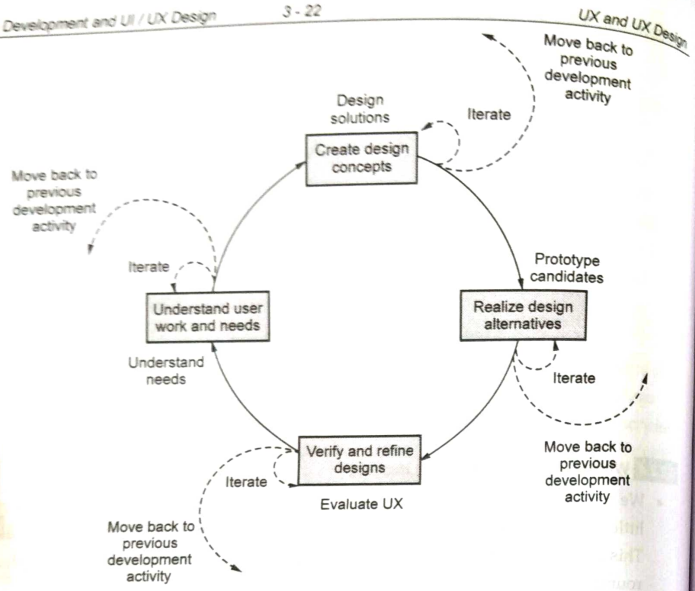


**Fig. 3.2.2 The wheel - A lifecycle template**

### 3.2.2.6 UX Methods

- A method, in our lexicon, is a manner of carrying out all or part of a lifecycle activity or subactivity. Usage Research is an example of a method for the Understand Needs lifecycle activity.

### 3.2.2.7 UX Techniques

- Finally, a UX technique is a specific comprehensive practise that you can apply to complete a step within an activity, subactivity, or method in our usage. A given UX design strategy is not limited to a certain UX method and can be used in a range of various lifecycle activities. Within the usage research UX method, examples of UX techniques for data elicitation include :
  - User interviews.
  - Observation of users at work.





### 3.2.2.8 A Hierarchy of Terms

- We end up with a form of vocabulary hierarchy to separate the levels of explanations of ways things can be done in UX design :
  - Process, or UX lifecycle process.
  - UX lifecycle activities and subactivities.
  - UX methods.
  - UX techniques.
- These terms, despite their common appearance in the literature, are usually vague and ill-defined within the discipline. We have chosen to shade the definitions of these closely related terms to reflect the meanings we think are most commonly understood within this loose kind of hierarchical relationship, examples of which are shown in Table 3.2.1.

| Lifecycle process | Traditional waterfall process |
|---|---|
| Lifecycle activity | Understand Needs |
| Subactivity | Elicit usage information, analyze usage information, model system or product usage, codify needs |
| Method | Usage research, surveys, and competitive analyses (for elicit information subactivity); usage research analysis (for analyze information subactivity); flow, sequence, task models (for model usage subactivity); formal requirements (for codify needs subactivity) |
| Technique | Interviews, observations, affinity diagramming, etc. |

**Table 3.2.1 Informal hierarchy of process, methods, and techniques**

## 3.2.3 Fundamental UX Lifecycle Activities

- The distinct UX design lifecycle activities and subactivities are discussed in greater detail in this section. This book devotes a significant portion of its content to these issues.
- Fig. 3.2.1 and Fig. 3.2.2 show the four basic UX lifecycle activities :
  - Recognize needs in order to comprehend users, work practises, usage, the subject matter domain, and, ultimately, design requirements.
  - Design solutions, which entails creating designs that serve as solutions.
  - Candidates for prototyping (promising solutions) to materialise and imagine promising candidates for design.

- Evaluate UX to confirm and improve designs in terms of the user experience they provide.

- Each box in the figure indicates a method for carrying out the associated lifecycle action for a given iteration. The method to be employed will be determined by the design circumstance.
- The use of separate boxes to represent UX lifecycle events is a simple approach to highlight each action. However, in fact, these tasks may not always have such distinct borders; there can be significant tangling and overlap.

### 3.2.3.1 Understand Needs UX Lifecycle Activity

- Understanding the business domain, users, work practise, usage, and the overall subject-matter domain is accomplished through the Understand Needs lifecycle activity. The most popular method is some variation of usage research, and the most rigorous version comprises the following sub activities :
  - Data elicitation : Interview and observe users at work and gather data about work practice, users, usage, and needs.
  - Data analysis : Distill and organize usage research data.
  - Data modeling : Create representations of user characteristics, information flow, tasks, and work environments.
  - Requirements extraction : Codify needs and requirements.
- In Fig. 3.2.3, we show how the data elicitation subactivity is carried out using various approaches and strategies.

### 3.2.3.2 The Design Solutions UX Lifecycle Activity

- The most significant lifecycle activity, and the one with the widest scope, is design solutions. As the project and the product expand and mature through these basic "stages" (Fig. 3.2.4), typical sub activities of this activity alter substantially over time :
  - **Generative design :** Ideation and drawing low-fidelity prototype, and critiquing for design exploration
  - **Conceptual design :** Making conceptual design candidates' mental models, system models, storyboards, and low-fidelity prototypes.
  - **Intermediate design :** Creating illustrated scenarios, wireframes, medium fidelity mockups of design forerunners, and identifying design tradeoffs to compare design candidates and developing ecological, interaction, and emotional design plans for the most promising candidates.





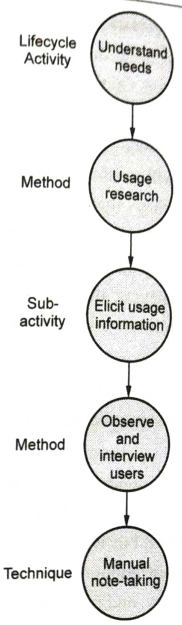


**Fig. 3.2.3 Data elicitation sub activity**

  - **Design production :** Specifying specific design plans for the new design choice's implementation
- The relative relevance of each of these subactivities varies according to the design environment, particularly the type of product or system being developed.

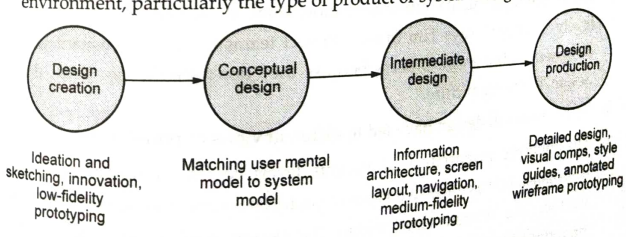


**Fig. 3.2.4 Design solution subactivities**

**Interpretation of "design": broad versus narrow**

- The ambiguity in the way the term "design" is employed in UX and other domains is a possible source of confusion. On the one hand, the entire Fig. 3.2.1 is referred to as the UX design lifecycle. As a result, one could conclude that this entire picture shows the solution to the question, "What is UX design ?"
- However, the astute reader will see that there is also a box at the top of the lifecycle named Design Solutions. Perhaps that is what UX design entails. Both uses of the term are correct, but we lack the vocabulary to distinguish between them.
- We avoid the ambiguity trap by adhering to the field's flexible traditions and using "design" in both senses, believing that context will provide clarification. If the context does not clarify our meaning, we will state it explicitly.
- **Broad interpretation :**
  - In its broadest sense, "design" refers to the entire UX design lifecycle process. The UX process is a UX design process, to put it simply.
- **Narrow interpretation :**
  - However, the astute reader will see that there is also a box at the top of the lifecycle named Design Solutions. Perhaps that is what UX design entails. Both uses of the term are correct, but we lack the vocabulary to distinguish between them. Design is significant enough as a lifecycle activity to warrant its own definitions, activities, theory, and practise.
- **Narrow view has led to misconceptions :**
  - When evaluated in isolation, the narrow perspective may have contributed to a long-standing misconception about the function of UX designers in projects.
  - "Historically, design has been considered as a downstream stage in the development process—the point where designers, who have played no previous role in the fundamental job of invention, come along and wrap the concept in a lovely wrapper," says Tim Brown. Project teams have been hesitant to include designers in the entire process because they don't see design, particularly UX design, in the big picture."
- **Different views of design have led to different views of prototyping :**
  - The multiple perspectives at various levels of UX design make it easy to distinguish between the two types of prototyping and assessment mentioned by Buxton. The limited view within the design creation subactivity includes prototyping, evaluation, and iteration to acquire the right design.





  - The purpose of prototyping, evaluation, and iteration as part of the overall UX lifecycle process is to get the design right. Prototyping is employed as drawing and speedy low-fidelity prototypes in the design-creation subactivity, whereas higher-fidelity prototypes are created in the Prototype Candidates activity in the entire UX design lifecycle.

### 3.2.3.3 Prototype Candidates UX Lifecycle Activity

- Prototyping is a full-fledged lifecycle activity for realising and envisioning promising design candidates in this context. The major subactivity is to produce high-fidelity design representations in the form of :
  - Paper prototypes.
  - Wireframes and wireflows.
  - Click-through wireframe prototypes.
  - Physical prototypes.
- Prototyping, like sketching, is frequently done in tandem with and in conjunction with design; a prototype is a developed version of a sketch. Designers create several types of prototypes as exterior design representations as their designs emerge in their heads. Because prototypes are created for a variety of goals, there are many different sorts of prototypes, each with their own set of processes and techniques.
- Low fidelity, medium fidelity, high fidelity, and "visual comps" for pixel-perfect look and feel are all examples of prototypes.

### 3.2.3.4 Evaluate UX Lifecycle Activity

- This task is about double-checking and fine-tuning the UX design to make sure we're on the correct track. Sub activities and alternative approaches for assessing, verifying, and refining designs for the Evaluate UX action could include :
  - **Collect evaluation data :** To mimic or comprehend actual usage and provide evaluation data, evaluate designs using empirical or analytic methodologies.
  - Analyze evaluation data
  - Propose redesign solutions.
  - Report results.
- There are numerous methods for carrying out the tasks and subactivities of UX evaluation. Depending on the design circumstances, inspections can range from light and quick to detailed and time-consuming, from full empirical research to quick and dirty inspections. There are also various ways and approaches for analysing the various aspects of UX, such as usability, utility, emotional effect, and meaningfulness.

### 3.2.4 UX Design Techniques as Life Skills

- UX approaches are used in UX design settings. We consider of these strategies as life skills since they are employed for problem-solving in our everyday lives. They are basic general skills for solving problems that enable designers and nondesigners survive as human beings.
- Furthermore, several of the approaches described in our process chapters are extremely useful. UX processes are unique. Here are a couple such examples :
  - **Card sorting :** A method for organising data obtained in user research or UX evaluation such that it can be understood and interpreted.
  - **Think-aloud :** Participants are urged to articulate their thoughts and plans while they interact with a design prototype or system, as part of the lab-based evaluation method for the Evaluate UX lifecycle activity.
  - **Note taking :** In the elicit information subactivity of the Understand Needs lifecycle activity, a technique for acquiring raw user data that encompasses variations such as voice recording, video recording, handwritten notes, and notes typed on a laptop.
- There are numerous UX design strategies available, with more being published in the literature and implemented in practise throughout time.
- According to our expertise as researchers and practitioners in this field, the most significant procedures, considered as general talents, are described here. Most of them should be known to you because you've come across or used them before. You can think of them as a sneak peek of what will be discussed in later chapters.

#### 3.2.4.1 Observation

- The practise of seeing an ongoing action with the goal of comprehending the underlying phenomenon is known as Observation.
- Exceptions, surprises, generalities, patterns, routines, sequencing, what works and what doesn't, issues and hurdles, and how people react to challenges are all things to look for. The ability to observe effectively can be tricky, but it supplies the inputs for thinking and deduction.
- Developing the ability to notice takes time and effort. UX experts must teach themselves to recognize vital information and not let it pass them by because it does not register. "Sight is a faculty; seeing is an art," wrote George Perkins Marsh.

**Example :** Observation of a Car Wash Workflow





- Here's an example of using the observation technique in a real-world situation: watching a car wash process issue. On a recent trip to Buster's Auto Spa, our neighbourhood car wash, I had to wait over half an hour to get in.
- To pass the time, I imagine most individuals would read the newspaper or listen to music. But, as someone who is blessed/cursed with UX designer abilities, I was intrigued as to what was causing the delay.
- It's worth noting that this is a practical demonstration of at least two other techniques: abstraction and drawing.
- The car wash works well most of the time, although there are times when it has a work flow issue. When a customer requests that the interior of their vehicle be cleaned, attendants use a huge vacuum cleaner that is installed at the entrance.
- Vacuuming can take up to 10 minutes, during which time no automobiles are moving through the car wash, which is terrible for both the car wash and the consumers who have to wait in line behind the vacuumed car. Waiting for customers is particularly inconvenient because getting your car washed is something you generally do on your way somewhere else. Furthermore, clients will find it nearly impossible to leave due to the tight path going up to the entrance. Naturally, I proceeded to consider various facility designs that would solve the problem. What would you suggest ?

#### 3.2.4.2 Abstraction

- Abstraction is the process of deleting details that are not relevant to a specific goal. "In computer science (CS) and software engineering (SE), abstraction is considered to be a critical ability underlying most activities". As a result, you can see what's vital without being distracted by irrelevant information. To put it another way, abstraction is the process of separating the wheat from the chaff.
- The ability to generalize from an example is also part of abstraction. You must be able to grasp and extract the substance of a specific observable incident or phenomenon as an example of a larger case or principle.
- Consider the following scenario: you are a designer and you are interviewing users of a house you are intending to build. Various users have expressed a range of needs :
  - User A : I would like a fan in the kitchen to get rid of any cooking odors.
  - User B : I prefer to let fresh air into my study by opening the windows.
  - User C : I do a lot of glue in my workshop and need a lot of windows and doors to keep the chemical odors at bay.

- All of these specific situations can be abstracted under the umbrella of a general concept of house ventilation, resulting in a design that addresses all of the difficulties listed.

### 3.2.4.3 Note Taking

- The practise of efficiently capturing descriptions of observations is known as note taking. It covers a variety of approaches for gathering qualitative data.
- Making handwritten notes, typing notes on a laptop, recording the essence on audio, or filming on video are all methods for taking notes. Note taking, whichever you do it, should be an almost subconscious action that does not divert your cognitive processes from the observation process.
- This usually entails utilizing the most basic methods available, such as handwritten notes or notes typed on a laptop. I kept a small digital audio recorder with me at all times and wrote my notes about the car wash difficulty when I arrived home.
- To be efficient, you must use abstraction when taking notes in order to record the most important ideas with the least amount of language. Sketches and / or models, analogies, or any other descriptive mechanism can be used in notes, adding to the mix of techniques.

### 3.2.4.4 Data / Idea Organization

- The activity of categorizing data by category to make raw data intelligible is known as data organization. The following are some examples of data organization techniques :
  - Card sorting.
  - Affinity diagrams.
  - Mind-mapping : "A mind map is a visual representation of information. A mind map is often built around a single notion, which is drawn as an image in the middle of a blank landscape page and to which linked representations of ideas, such as images, words, and parts of words, are added. Other concepts branch out from the fundamental thought, and major ideas are linked directly to it."
  - Concept mapping : "A concept map, also known as a conceptual diagram, is a graphic that shows possible connections between concepts. Instructional designers, engineers, technical writers, and others utilize it to organize and structure knowledge."
- In desktop applications, card sorting is widely used to organize menu structures.





Designers create a list of all the actions that the system should support, which are then printed on cards. After that, users are prompted to sort them into groups.

- Mind-mapping and concept-mapping are approaches for externalizing loosely organized and related ideas and data.

### 3.2.4.5 Modeling

- Modeling is the process of simplifying and aiding comprehension by depicting complicated and abstract phenomena along specific dimensions. It is a means to categorize or describe components of the problem area.
- Modeling is a type of abstraction that is used to represent and identify objects, relationships, actions, processes, variables, and dependencies.
- Modeling is a technique for organizing and presenting data in order to gain a better understanding of it. It is a technique for extracting generalizations and correlations from unstructured data.
- Consider the personal research you conduct before purchasing a car as an example of modelling as a life skill. You will come across a variety of car-related facts. You must organize this information using some kind of model in order to compare one car to another.
- Your model may incorporate dimensions such as style, aesthetics, and technological parameters to consider. Convertibles, SUVs, and sedans are examples of different styles. You might think about body shapes, aerodynamic appearance, and colour options when it comes to aesthetics. Cost, MPG, horsepower, torque, and all-wheel drive versus front-wheel drive are all factors to consider when looking at technical characteristics.

### 3.2.4.6 Storytelling

- The process of utilizing narrative to describe parts of a phenomenon or design with the goal of immersing the audience in the phenomenon is known as storytelling.
- In the world of advertising, storytelling is a popular tactic. Telling stories of people who use a product and gain joy and/or utility from it in their lives can be more interesting than simply listing the product's benefits.
- Storytelling is an important life skill that may be applied to a variety of situations. More engaging than simply discussing square footage and other amenities is a real estate salesperson narrating the storey of a house, including when it was built, who lived there before, and who lives there today.

- The stories allow us to imagine ourselves living in this place and creating our own memories there.
- When it came to introducing themselves or doing design portfolio walkthroughs, we discovered that job candidates that used storytelling were always more intriguing and relatable.
- Storytelling enabled these candidates to express the background of a project in a rich and compelling way, including the design brief, the challenge, the politics, and the culture.
- Candidates who did a "presentation" of their portfolio one design slide at a time, on the other hand, were less effective since there was no glue to bind the disparate design snapshots in each slide into a cohesive narrative we could relate to.

### 3.2.4.7 Immersion

- Immersion is a type of thorough examination and thought about the topic at hand - to "live" within the context of a problem and discover connections between its various parts.
- As in a war room, immersion entails surrounding oneself in your UX work area with creative design artifacts. You block out all outside distractions, and what you see serves as a kind of cognitive scaffolding and trigger for creative ideas.
- The artifacts operate as stimulants, triggering framings and bringing links and interconnections to the surface. Everything you observe serves as a kind of cognitive scaffolding and catalyst, assisting in the emergence of new ideas.
- Consider the following scenario of immersion in a non-UX environment. When a police investigator is working on a difficult case, she may shut herself away in a "war room" away from other distractions on the job or in the area.
- She will immerse herself in the subject and surround herself with artifacts like crime scene images and sketches, as well as police and witness records. She might also hang a timeline on the wall, along with sketches of the life story of people involved. She spends so much time studying the problem that she "becomes" the perpetrator.
- Finally, while much of your immersion will have to take place in your UX studio, immersion on-site can be a useful supplement for analysis and design.

### 3.2.4.8 Brainstorming

- Brainstorming is an interactive group discussion technique for examining various ideas, challenges, and solutions :





- This is a group activity that must be completed. The input and conversation of each person encourages, motivates, and inspires the others.
- Highlighting diverse perspectives and generating different framings of a phenomenon or an issue is a major talent in the Design Solutions lifecycle activity.
- Can be used to create solutions to recognised UX problems throughout the Evaluate UX lifecycle activity.
- It can be employed in any case where there is an open-ended problem. Who are the system's potential users, for example ? Where can we find evaluation participants ?

### 3.2.4.9 Sketching and Drawing

- Sketching is the technique of sketching basic sketches and diagrams to express the essence of problems and solutions in the field of user experience.
- It is a technique of externalizing object study and investigation, as well as an evolving awareness of problems and solutions. The most important thing to remember about sketching is that it is not about aesthetics or art.
- It is all about getting your message through. Do not be concerned if your sketches aren't properly proportioned or artistic. A discussion of how sketching is used as an integral aspect of generative design may be found in next section.
- A prototype is similar to a drawing. It employs an abstract representation to improve visualization by emphasizing key elements. Sketching genuinely aids your thinking by utilizing the hand-eye connection to cognition in the brain.
- This can help with creativity by improving cognition. Ideation must always be followed by sketching. "If you are doing design, you are sketching," explains Buxton.
- Sketching is a life skill that can be used in a variety of situations. Are you rearranging the furniture in your home ? Make a rough drawing of the desired setup or layout first. Make a model of your workflows for that area and immerse yourself in the room's usage environment to back it up.

### 3.2.4.10 Framing and Reframing

- The practise of framing and reframing a situation within a specific perspective is known as framing and reframing.
- Framing creates a framework for the topic that organises it and highlights the aspects you will investigate. A framing is a pattern or a certain topic through which we view everything as we search for solutions. We can ask "what if ?" and "why do

not we do this ?" in the framework of a framing. Reusable framings improve strength with each use.

- To develop a frame, you must return to the problem's fundamental ingredients, the underlying abstract phenomena, determine what is truly going on, the problem's essence, and dismiss the rest as noise. Framing is best done as a group exercise because it is a specialized type of brainstorming.
- Here is a non-UX illustration of how different framings result in different answers in practically any problem-solving situation. Assume that fields along the flood plains of a river near a town are occasionally flooded.
- To develop a frame, you must return to the problem's fundamental ingredients, the underlying abstract phenomena, determine what is truly going on, the problem's essence, and dismiss the rest as noise. Framing is best done as a group exercise because it is a specialized type of brainstorming.
- If you frame it as a problem of the river overflowing its banks, you can consider constructing a system of dikes to keep the water contained. Alternatively, if you frame it as a natural phenomenon to work with rather than against, landowners will be required to build dwellings only above the flood plain.

#### 3.2.4.11 Reasoning and Deduction

- The method of utilizing logic to analyse observed facts, put them together, and arrive at a logical conclusion is known as reasoning and deduction. The logic's predicates are the observations, and the deductions are the conclusions. Reasoning and deduction are methods for synthesizing new facts by applying logic to existing information.
- To arrive at user needs based on usage research, design features based on needs, tradeoffs and restrictions based on insights from the work domain, reasoning and deduction are frequently employed in UX.

#### 3.2.4.12 Prototyping and Envisioning

- Prototyping is the process of creating or constructing a model or mockup of a design that can be altered and used to imitate or manifest a user experience that can be evaluated.
- Sketching is expanded upon by prototyping. A prototype is a platform for imagining and testing the efficacy of a design as a problem solution, and it is the primary output of UX design.





#### 3.2.4.13 Critical Thinking

- The activity of "objective investigation of data to create a judgement" is known as critical thinking. The subject is complicated, and there are various definitions, but the most common ones include "the reasonable, sceptical, and unbiased investigation or evaluation of factual evidence."
- For testing, analyzing, diagnosing, verifying, or validating a prospective design solution, critical thinking is a key component of UX evaluation. This type of assessment necessitates observation, abstraction, data gathering, note-taking, reasoning, and deduction skills, as well as the capacity to generate judgments, ranks, and ratings.

#### 3.2.4.14 Iteration

- Iteration is the process of refining a concept or improving a design as a problem solution by repeating a cycle of analysis, design, prototype, and evaluation.
- The rereading and reediting of a paper or report that an author might conduct before submitting it is a simple non-UX example of iteration.

#### 3.2.4.15 UX Techniques are used in Combination

- These strategies are frequently incorporated within methods when utilized in UX design or as life skills. A police investigator, for example, must combine talents such as observation, note-taking, narrative, immersion, brainstorming, sketching, framing, and reasoning and deduction to solve crimes.

## 3.3 Scope, Rigor, Complexity and Project Perspectives

### 3.3.1 Introduction

- Let us go through some of the criteria that define a design situation and how they affect lifecycle decisions.

#### 3.3.1.1 Rigor and Scope : Project Parameters that Determine Process Choices

- You must make decisions in any design situation. Even within an agile UX process, there is no such thing as a one-size-fits-all UX lifecycle activity or method. Your goal is to adapt UX lifecycle activities, processes, and strategies to the needs of specific projects. The majority of high-level process decisions are based on two primary factors : rigour and breadth.

## 3.3.2 Rigor in a UX Method or Process

### 3.3.2.1 What Is Rigor ?

- The degree of formality, completeness, precision, and correctness with which you perform all of the processes determines the rigour of a UX design lifecycle activity, method, or approach. It is also about how diligently you keep and document the completeness and purity of the data you acquire, particularly data from usage study and UX review.
- **Completeness**
  - The term "completeness" refers to the thoroughness with which methods are used, which means that each phase is covered completely. Completeness also applies to data from usage studies and evaluations.
  - Designers benefit from this attention to detail since it allows them to cover all the bases and fill in all the gaps, ensuring that no functions or features are overlooked.
- **Purity**
  - Purity entails being as accurate as possible with your data, as well as avoiding allowing additional bogus "data" to enter. For instance, if you have a lot of data, Designer insights or conjectures should be marked with information so that they can be identified and distinguished from actual user input.
- The approaches for ensuring data completeness and purity are referred to as rigorous methods.

### 3.3.2.2 Complexity as an Influence on the Need for Rigor

**The system complexity space**

- One of the main reasons why there isn't a single set of methods for designing all systems is that there is a wide range of system and product types, each with a different level of risk versus the need for rigour in lifecycle activities and methods.
- As shown in Fig. 3.3.1, we demonstrate a complexity space for systems defined by the dimensions of interaction complexity and domain complexity. While there are definitely other ways to divide the area, this method works well for us.





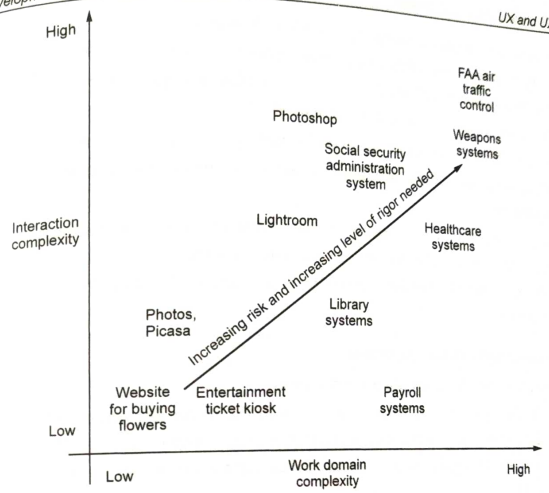


**Fig. 3.3.1 System complexity space**

**Interaction complexity**

- The intricacy or elaborateness of user activities, including the difficulty of cognitive operations, required to complete tasks with the system is represented on the vertical axis by interaction complexity.
- Low interaction complexity is commonly associated with systems that facilitate simple, minor operations, such as ordering flowers from a website.
- Larger and more difficult jobs, such as altering a colour image with Adobe Photoshop, are frequently associated with high interaction complexity.

**Domain complexity**

- We present work domain complexity on the horizontal axis, which is about the complexity of the work domain. the level of difficulty and the technical, if not esoteric, character of the problem a field of work that is related Mechanisms that are complex and intricate Within the ecology of the system, elements of the system work and communicate, add to the complexity of the domain.

- In domain-complex systems, user work is frequently mediated and collaborative, with numerous "hand offs" in a complex workflow involving multiple dependencies and communication channels, as well as compliance requirements, regulations, and exceptions in how work instances are handled.
- Systems for geological fault analysis for earthquake prediction, worldwide weather forecasting, and entire healthcare systems are examples of high work-domain complexity.
- Low work-domain complexity indicates that the system's operation within its ecosystem is relatively straightforward. The same website for ordering flowers and a basic personal calendar management tool are both examples of low domain complexity.

**System complexity space quadrants**

- **Simple interaction, simple work domains**
  - In Fig. 3.3.1, the simplest quadrant is in the lower left corner, where both the interaction and work domains are the least complicated. Smaller web pages, interactive applications, and numerous commercial items can be found in this area.
  - The fact that this is the simple-simple quadrant does not imply that the items are extremely simple; in fact, the products in this quadrant can be sophisticated.
  - There are a lot of relatively simple systems out there. Some commercial software products, but not all, are domain- and interaction-simple, at least in comparison to big systems of other types.
  - This quadrant is exemplified once again by the web page for ordering flowers. Interaction is straightforward; there is only one major task with a few options, and the job is done. Because there is only one user at a time and the process is practically straightforward, the work domain complexity is similarly low.
  - Many mobile apps, which account for a sizable portion of the commercial product market, are simple - interaction and simple - domain.
  - Although emotional impact elements do not apply to every system or product in this quadrant, they can be highly essential, especially when it comes to factors like aesthetics or pleasure or joy of use.
  - Commercial products in this quadrant that have emotional impact issues, such as meaningfulness, include the smartphone and personal mp3 music player. Because the systems and commercial goods in this sector are less complicated, they demand less rigour in their development.





- **Complex interactions, complex work environment**
  - As you proceed over the diagonal in Fig. 3.3.1, you will reach the upper right quadrant, which contains interaction-complex and domain - complex systems, both of which are typically huge and sophisticated.
  - **Serious system**
    - Serious systems are found in the system complexity space's upper right corner. An air traffic control system, for example, is used to determine landing instructions for an inbound airliner.
    - With workflow and collaboration across a large number of work roles and user types, an air traffic control system has considerable domain and interaction complexity. A huge system for the social security administration is another distinguishing example for this area.
    - Resistance to innovation is most likely to be seen in large domain-complex systems, such as military weapons systems. Conformity can be believed to be more important than radical designs in some cases.
    - Even though there are superior alternatives, users and operators may commit operations to habit and accomplish tasks with learned behaviour. Changes in work practices must be undertaken with caution.
    - This subset of serious systems in the system complexity space usually has nothing to do with emotional impact elements like beauty, entertainment, or ease of use, if at all.
  - **Enterprise - level system**
    - The so-called enterprise systems, massive information systems used within businesses that have generally been overlooked in debates about usability and user experience, are among the types of systems seen in the rest of this quadrant. Most significant firms globally are hooked into some type of corporate technology.
    - Unfortunately, such systems are not only difficult to implement and maintain, but they are also tough for employees to utilize. Why is it that the user experience of corporate technologies is so poor when the stakes are so high ?
  - **Utmost level of rigour is required**
    - The domain - complex and interaction-complex quadrants include the most hazards and, as a result, the greatest requirement for risk management rigour. The following are the project's requirements :

- When you have the most stringent regulatory compliance obligations.
- When the necessity of avoiding errors in usage is greatest.
- When you can't afford to make a mistake and the cost of failure is too high.
- When there is a contractual requirement for high rigour.
  - These systems are often among the most difficult and expensive to design and develop due to their sheer magnitude and the necessity for rigorous processes.

- **Simple work domain, complex interaction :**
  - Located in the upper left quadrant of the Fig. 3.3.1, one of the "in-between" quadrants, we may see interaction-complex and Domain-simple systems are systems that have only one domain. A huge volume of functionality, resulting in a big number and broad variety of complex user activities, is typical of an interaction-complex system.
  - A simple yet effective illustration is an older-style digital watch. A broad array of modal settings with overloaded and unlabeled push buttons contribute to the interaction complexity. The domain, on the other hand, remains straightforward, focusing on the question of "what time is it ?" Workflow is basic; in a simple system ecology, there is just one work position.
  - Interaction design - a wide range of jobs, screen layouts, user behaviours, and even metaphors - requires attention in this sector. For the consistency and usefulness of the conceptual design and the detailed interaction design, rigorous formative evaluation may be required.
  - Modeling will be focused on tasks - task structure and task interaction models - and possibly the artifact model, with little attention paid to work roles, workflow, or the majority of other models.
  - Usage research rarely produces fresh information that may be used to inform design in simple work domains, regardless of interaction complexity. As a result, less rigour will suffice prior to design creation. Complex interaction, on the other hand, necessitates thorough and methodical brainstorming, ideation, and sketching, as well as iterative review and refinement and consideration of emotional effect variables.
- **Interaction is simple, work domain is complex.**
  - We observe interaction-simple and domain-complex systems in the lower right quadrant of Fig. 3.3.1, the other "in-between" quadrant.





  - Because the user tasks in this quadrant are very simple and straightforward, less attention to task descriptions is required. Understanding the domain and its often obscure work practices is the primary endeavour for users in this sector. To focus on conceptual design and easy-to-understand user models of how the system works, designers need to conduct extensive usage research.
  - Once that is understood, people will find the interaction to be relatively simple. The underlying domain of tax preparation software for average homes is an excellent illustration because the data entering into forms can be simplified to a step - by - step process.
  - Libraries management systems, depicted in the middle of the work domain complexity scale near the bottom of Fig. 3.3.1, occasionally fall into the simple - interaction, complex-work-domain quadrant.
  - Because the range of tasks and activities available to each user is typically limited and straightforward, and the difficulty of each user work is modest, typical library systems have low interaction complexity. As a result, modelling jobs for a library system, for example, does not require as much rigour.
  - A full library system, on the other hand, has a lot of domain complexity. The work practices of library systems can be esoteric, and most UX designers will be unfamiliar with them.
  - To handle the surprisingly significant and highly controlled minor details in cataloguing procedures, for example, additional training is required. As a result, a methodical approach to user research may be required.
- **Gradation within the system complexity space**
  - Some systems or products clearly fall into one of the quadrants, while others may have a hazy quadrant boundary.
  - Websites, for example, might fall into various quadrants depending on whether they are for a huge corporation's intranet system, a massive e-commerce site, or a simple photo sharing site. A printer or a camera, for example, have a low domain complexity but a medium interaction complexity.
  - Healthcare systems frequently traverse the four quadrants of system complexity. Some aspects of a tiny doctor's office may be straightforward. Large healthcare systems, on the other hand, have complicated work domains that integrate medical instruments, health record databases, and patient accounting.
  - Machines in a patient's hospital room, likewise, can perform a wide range of technical jobs and activities, resulting in a high level of interface complexity.

- Regulation, paperwork, and compliance difficulties, as well as legal and ethical constraints, all contribute to high work domain complexity and a high demand for rigour in UX lifecycle activities and techniques in the healthcare sector.

### 3.3.2.3 Domain Familiarity as an Influence on the Need for Rigor

- Even if a domain isn't particularly complicated in absolute terms, it will appear complex to the UX designer, at least at first. The designer's cognitive scaffolding for understanding the design problem includes familiarity with the target domain. The best approach to obtain this domain familiarity is to start with rigorous UX methods.
- Our house - building example is in a domain that we are all at least vaguely familiar with. Consider the world of financial portfolio analytics and management, which is highly specialised and esoteric.
- If the designer is unfamiliar with the domain, there is a higher requirement to record as much information as possible from users and usage so that they may refer back to it for clarification or education.
- Users from various portfolio management businesses explained distinct investment techniques and ideologies in a variety of ways, making it more difficult for me to retain a consistent grasp of the domain.
- To preserve rigour, metadata such as who said what and where they worked was routinely included during usage research data collecting sessions with customers, users, and Subject Matter Experts (SMEs) because such detail was critical for later analyses to contextualise what was said.

### 3.3.2.4 Risk Aversion Influences the Need for Rigor

- Risk refers to the danger or chance of things going wrong, of features or requirements being overlooked, or of the end result failing to meet the expectations of users; it also refers to the possibility of failing to accommodate legacy requirements or failing to adhere to legal or safety limits.
- One significant goal - related factor in deciding which path to take is Aversion to risk is the level of rigour in the UX design lifecycle activities and processes. In some circumstances, failing to get the UX design right poses a significant risk, often due to requirements for legacy support, regulatory limits, or safety issues.
- A legacy system is an ancient method, technology, computer system, or application programme of, relating to, or being a previous or obsolete computer system with long-term maintenance issues.





- The lower the risk tolerance, the more rigour in the lifecycle activities, procedures, and approaches is required. High rigour throughout the lifespan process, on the other hand, will add cost and time to the process.

#### Risk of data loss

- The loss of completeness is the most severe data loss in the UX design process. The data is simplified, summarized, and generally abridged for the sake of speed and economy. By documenting (audio or video) every aspect of user interviews and observations, you may avoid data loss. However, trancribing the recordings is time-consuming and expensive, and the end result is a large amount of text to sort through in order to isolate the crucial information from all the noise and irrelevant information; it's nearly never worth it.
- As a result, many usage researchers just take notes to remind themselves of the essential aspects. Although this results in a loss of completeness, it does not necessarily affect quality or utility of the data.
- In any case, raw data usually requires a step of abstraction to remove unnecessary detail and highlight what is important. However, this abstraction can be taken too far. You start to lose data that could be vital later in the process if you are sluggish, negligent, or don't take enough time.

#### Risks associated with legal, safety, and compliance constraints

- Legal requirements for public safety concerns must be addressed extremely seriously in systems such as air traffic control, healthcare and medical records systems, and financial systems.
- Designing a financial product necessitates a procedure that incorporates compliance with complicated federal rules into the commercial process of using the product. Your lifecycle activities and techniques must be rigorous enough to assure (and potentially even prove) that the system passes all of the compliance tests.

### 3.3.2.5 Stage of Development within Your Project as an Influence on the Need for Rigor

- Another factor that influences the need for rigour is the level of development of your project. Over time, all projects go through many "stages." The appropriateness of a level of rigour and corresponding choices of UX methods and techniques for lifecycle tasks will change as a project advances through these stages of

development, regardless of method selections based on other project factors. Early phases, for example, may necessitate a heavy focus on rigorous usage research in order to collect the most user, usage, and domain knowledge possible.

- However, there may be a lack of emphasis on rigorous review in the early stages. Because the design is still changing, investing a lot of money in early evaluation could be a waste. As a result, lightweight, quick, and frequent evaluation approaches may be preferable in the early phases. A rapid design review, for example, could be used to assess an early conceptual design.
- You might progress to UX inspection of a low-fidelity prototype in subsequent phases of evaluation to enhance a now-stable design, and then to more rigorous lab-based testing using a higher-fidelity prototype, increasing the amount and quality of data you need to keep at each step. Keeping track of this additional data necessitates more diligence.

### 3.3.2.6 Project Resources

- High rigour is costly and time-consuming. Budget constraints and tight deadlines are practical realities that can limit your lifecycle activity and method options, as well as the amount of rigour you can give.
- Person power is another essential type of resource. What project team responsibilities are they capable of filling, and what UX skills, experience, and training do they offer to the project ?
- Formal components of rigorous procedures, such as substantial usage research or explicit UX evaluation goals and targets, are likely to be less necessary for UX specialists with extensive experience and maturity. Following the method in detail does not add much to what these seasoned UX specialists can achieve with their already absorbed knowledge and refined intuition.

### 3.3.2.7 Being Rapid in Lifecycle Activities, Methods and Techniques

- While certain procedures are inherently more rigorous than others, it's crucial to remember that every method can be applied to a range of rigour levels.
- More rigour in the use of an evaluation method can result in more thorough and accurate results, but it takes longer and costs more money. Similarly, you can usually speed up and lower the cost of practically any UX evaluation approach by employing shortcuts, but at the sacrifice of rigour.





#### Not every project needs rigorous UX methods

- High rigour is not necessary, is not worth the cost, or just is not practicable for many projects, especially in the commercial product space and often in the enterprise system space.
- As a result, less stringent UX methodologies and strategies have emerged in the literature and practise, allowing you to acquire good outcomes from your time and resources while being faster and less expensive.

#### Rapid methods are a natural result

- The goal of working at a lesser level of rigour when higher rigour is not required is to reduce lifetime cost and time.
- You can usually be less rigorous by abbreviating the "normal" rigorous methods and approaches, which include taking shortcuts, skipping needless stages, and keeping only the most critical data. Alternatively, you can apply less strict rapid alternative procedures.

#### Over time our need for rigor has diminished

- Another factor is that UX practise has grown and we have improved our art. As a result, in most ordinary UX circumstances, we don't need to be as strict and detailed. As an anomaly, the arduous and time-consuming fully rigorous method has been abandoned.
- A rare job that necessitates complete rigour will come with detailed instructions on what to do and how to accomplish it. The "standard" technique for each lifecycle activity is now simply a "regular" process, which is a mix of practical middle-of-the-road rigour and some clear ways to save time.

#### Rapidness principle

- Regardless of other considerations, you should always choose the quickest solution. All lifecycle activity, method, and technique decisions are guided by the principle : Go as fast as you can, within the limits of project goals and the necessity for rigour.
- It is a common misperception that working slowly and thoughtfully will result in a superior product and the ability to resolve problems as they arise. This myth has been disproved by the agile approach to both UX and SE.
- Being cost-effective and light is now a designer's way of life, and it's at the heart of agile processes. Even when a rigorous approach is required, you should just do what comes easily in terms of making wise decisions and not spending time or money on tasks that will not contribute to your final design.

- If one or more of the types of models are not required for the project, an otherwise rigorous technique could be used without them.

### 3.3.3 Scope of Delivery

- The term "scope" refers to the way the goal system or product is "chunked" for delivery in each iteration or sprint for agile implementation. The UX roles offer their designs to the SE persons in chunks of a certain size, and the SE role consumes the chunks of UX design as code specifications, which it then implements and presents to clients and users as chunks of possibly any other size of functional software.
- In a broad sense, chunks can be made up of several features or even a vast number of them. A part of the system chunks are frequently made up of one feature at a time in a narrow scope, which is synonymous with agility. In today's agile software development, even huge and sophisticated business systems are being produced utilizing small-scope methodologies. Within an agile UX process, we, the UX designers, will always end up delivering our designs in little portions.
- Some early UX design effort, such as establishing the conceptual design, may, nevertheless, necessitate a greater scope. A broad scope, of course, provides a useful framework for discussing the UX lifecycle activities in this book.

### 3.3.4 Commercial Product Perspective and Enterprise System Perspective

- Process parameters such as rigour, scope, and complexity have occupied the most of this chapter. Another element that effects your UX design strategy is whether you're working on a commercial product or an enterprise solution. Obviously, designing a smartphone with the same strategy as, say, a massive corporate resource management system isn't appropriate.

#### 3.3.4.1 Commercial Product Perspective

- We use the term "commercial product viewpoint" to characterize scenarios in which our designs are aimed towards a personal object, such as a device or software app, that a customer purchases for personal use. A consumer's perspective on a product is a consumer's perspective on a product.
- Multiple users and multiple activities can still be involved in a product. Within the product viewpoint, these scenarios can be viewed of as miniature systems; for example, a network of instructional and entertainment devices connected to the cloud.





**Single - user products**

- A single-user product design perspective's usage context is usually quite narrow and straightforward. Games built for a single player, personal calendar software, portable music players, and cellphones are all examples.
- A single-user product, on the other hand, does not have to be a standalone application; it can be part of a network of communicating applications that offer activity - based consumption. A coordinated use of a calendar, a contacts list, and email is an example.

**Multiuser collaborative products**

- Multiuser products are comparable to enterprise systems in that they involve several users and actors participating in different activities, and there may be a flow of information or even usage artifacts.
- However, the scene is set in a different manner. In contrast to an organization's enterprise system perspective setting and organizational goals, a multiuser product perspective setting is more akin to a user community with its own aims, which could be cooperative or competing.
- Families exchanging music on a set of smartphones and family interactions with living room devices like Amazon Echo are two examples.

#### 3.3.4.2 Enterprise System Perspective

- The term "enterprise system perspective" refers to circumstances in which several users and actors are involved in different activities, typically in an organizational environment. In this case, the goal of work practise is to help the organization achieve its business objectives.
- Many and often disparate system usage responsibilities exist in organizations, each contributing to a portion of the overall work that is completed. These items are often not held by a single user, and their manifestation to the consumer is not as concrete or self - contained. They can appear abstract and impersonal to the people who utilize them.
- Of course, some projects will have design goals that are in between the two points of view.
- Now that we've covered the topics of scope, rigour, and complexity, let us move on to the next step. We're now ready to talk about what it takes for the process to succeed. In order for the UX lifecycle to work in an agile environment, it must be flexible. We use the funnel model to do this.

## 3.4 Agile Lifecycle Processes and the Funnel Model of Agile UX

### 3.4.1 Challenges in Building Systems

#### 3.4.1.1 Change Happens During a Project

**Evolution of project requirements and parameters**

- A lot of the talk these days about lifecycle processes is about how well they can adapt to change. Why is it so vital to be able to adapt to change ? First and first, may be the most important lesson in the history of software engineering (and UX design) is that change is unavoidable. The majority of project parameters vary throughout time, including :
  - Requirements.
  - Product concept, vision.
  - System architecture.
  - Design ideas.
  - Available technology.
- This set of project parameters will be referred to as the "requirements" for the sake of simplicity.
- Here are some things we know :
  - Technology is always changing; it improves and becomes faster, and there are technological paradigm shifts.
  - Because change happens over time, the longer it takes to deliver a release that can be assessed, the more the initial needs and evaluation comments can diverge.
  - Because a wider scope equals a longer time to produce, the scope must be limited to minimize the impact of change.

**External changes**

- Things in the outside world can also alter during a project, for example :
  - At the time, technology was available.
  - The client's goals and objectives.
- We have limited control over these changes because they are external to the process, but we must be responsive.





#### 3.4.1.2 Two Views of These Changes

**Reality**

- As the project progresses, the "reality" view of change reflects true changes and reveals "real requirements."

**Designer's understanding of these changes**

- In this context, we will use the term "designer" to refer to both the UX designer and the entire team, which includes important SE responsibilities. The designer's perspective on change is the consequence of evaluation input and reflects the designer's comprehension, awareness, or perception of the changes.

#### 3.4.1.3 Gap Between Views

- There is a disconnect between reality and the designer's impression of requirements. The designer's vision of requirements usually lags behind reality in terms of time and substance, but a successful project requires that this gap be kept minimal.

#### 3.4.1.4 Responding to Change

- Closing the gap between reality and the designer's vision of reality - that is, how closely the designer's understanding reflects the real requirements - is critical to the designer's and the entire team's ability to respond to changes during the project. This is all about selecting lifecycle processes, which is a major theme in this book.

#### 3.4.1.5 Closing the Gap

- Closing the gap between real requirements and the designer's understanding of real requirements necessitates the designer's understanding of real requirements being updated as changes occur. The designer's consciousness is updated through the process of learning and receiving feedback.

#### 3.4.1.6 True Usage is the Only Ascertainer of Requirements

- The process of imagining usage is used to describe perceived requirements. On the other hand, learning from real-world feedback is essential for tracking changes and determining real-world requirements.
- We won't know what the genuine needs are until we've used the system. However, there is a problem : you cannot create the system to test it out unless you know the criteria.

- This problem requires a lifecycle process that can handle a continual state of flux. No, that isn't sufficient. Your lifecycle process must not only handle change; it must welcome ongoing change as a critical method for learning throughout the process.
- This necessitates a lifecycle approach that provides input on needs prior to the entire system being developed, as well as feedback from real-world usage at each step, which is critical for learning about how conditions are changing.

### 3.4.2 Old Waterfall SE Lifecycle Process

- The waterfall process was created in the pre-consumer period, when the majority of systems were massive enterprise systems and users were taught to utilize them for specialized business goals. There were no usability or user experience considerations during the development of the system.

#### 3.4.2.1 Waterfall Process was an Early SE Attempt to Get Organized

- The waterfall process was one of the first formal software engineering lifecycle procedures. The waterfall method is one of the easiest ways to combine lifecycle activities to create a SE lifecycle process. Because it was described as an organized, largely linear sequence of stages, each of which flowed into the next like the cascading tiers of a waterfall, the process was given that name (Fig. 3.4.1).

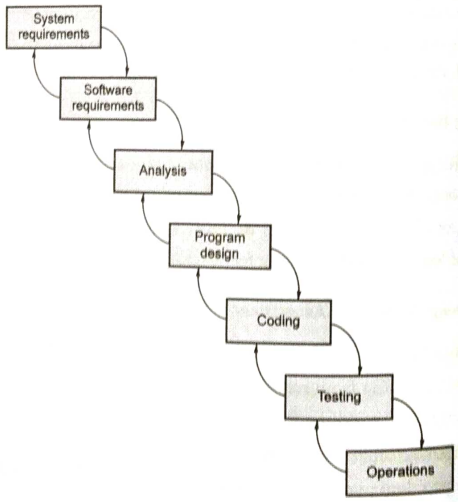


Fig. 3.4.1





- The waterfall technique was designed to supply the entire system at once. This technique was thorough and tended to be tough because it was an attempt to overcome the prior "Wild West" approach to development. It is also helpful to remember that the systems being constructed at the time were largely for huge corporations or government agencies.
- The waterfall process, however, was slow, unwieldy, unmanageable, and not very responsive to change because it operated with such a broad scope.

#### 3.4.2.2 Waterfall Process did have Some Feedback, but Not the Right Kind

**Verification and validation of phase work products**

- Before the entire system was deployed, the waterfall approach did not completely lack input. Software developers introduced formal evaluation in the form of verification and validation at the end of each phase to include a formal evaluation component.
- Verification ensured that the software met standards, while validation ensured that it accomplished its goal. Verification and validation aided a little by checking in with users on a regular basis to verify if their demands were still "valid" and on track.

**Verification and validation was not enough**

- While verification and validation allowed for some client input after each phase, it hampered the ability to track actual changes since feedback :
  - The scope of the project was extensive.
  - Only happened at the end of each big phase.
  - It was based on analytic reality checks rather than actual practical usage data.

**Change discovered was too expensive to address**

- It was costly to go back and fix the entire system when issues were eventually discovered at the conclusion of each step. New requirements identified later in the process necessitated costly rework because they rendered the prior phases' deliverables obsolete.
- A problem discovered during implementation was many times more expensive to remedy than a fault discovered during requirements, according to studies conducted at the time.

- Despite the fact that the phase deliverables were not the real system, a significant amount of effort was put into fixing them because it was the only thing they could change. The waterfall process failed to achieve prerequisite feedback must be based on real usage, even though everything was adjusted with the new insights.

**Feedback was not communicated well with respect to user needs**

- In terms of the feedback they did receive during the waterfall process, it frequently failed to meet prerequisite of communicating user feedback appropriately, because the focus was on the system rather than the user at the time. Verification and validation addressed some communication issues by allowing users to review the evolving system design, but SE people took a system approach and focused on "under the hood" issues. And, in many cases, these new system design artifacts were too complex and abstract for consumers to comprehend.

**The bottom line**

- Because the waterfall lifecycle method represented the pinnacle of batching with a very vast scope rather than incremental results, it did not produce nearly enough feedback along the way to effectively handle change.
- Because of the project's broad scope, customers and clients did not see much of the real product or system until it had completed all of the lifecycle stages.
- Many things had changed by that time, but there had been no mechanism to learn about them and adjust to them inside the waterfall model's lifecycle. As a result, UX and SE teams had to work extremely hard and diligently in each lifecycle activity to reduce the difficulties and errors that always arose, making the process long and laborious.
- This era's culture emphasized a long product or system shelf life because it was too tough to change. The team, on the other hand, was about to breakup and go on to new assignments, so none of this was a concern.
- But it was a different story for the client and users : goals were missed, certain requirements were incorrect, others were not completed, the system had serious usability issues, and the final deliverable failed to meet its intended purpose.

### 3.4.3 Embracing an Agile Lifecycle Process

- An agile lifecycle process (UX or SE) is a small - scope strategy in which all lifecycle activities are completed for one product or system feature before the cycle is repeated for the next item.





- An agile approach is defined by modest and frequent releases to obtain early usage-based feedback and is driven by demands expressed as user stories of capabilities rather than abstract system requirements.
- A search for alternate SE procedures led to the idea of agile SE as a result of the concerns with the waterfall method outlined in the preceding section. Each lifecycle activity for the entire product or system is completed using the waterfall method.
- In an agile lifecycle process, you complete all of the lifecycle tasks for one product or system feature before moving on to the next.
- Agile methods are known for being quick, iterative, and adaptable to change. Because the word "agile" denotes "nimble" or "sensitive to change," this makes sense. The following issues are addressed by agile processes :
  - A good project must have a small scope so that the time it takes to provide a release is constrained by delivering the first chunk reasonably soon because it is easier to implement.
  - The gap between reality and the designer's understanding of the same must remain small in a successful project and feedback from actual usage is the only way to know real requirements by delivering a small chunk that customers can use, bridging the gap between perceived and real needs.
  - Feedback on requirements must be provided successfully in a successful project by :
    - Instead of abstract system requirements, the needs should be expressed as user stories of capabilities.
    - Telling stories about discrete, manageable features rather than the entire system.
- Furthermore, by the time agile processes was on the scene, the world of system development had begun to embrace smaller, less complex systems. While some of the major commercial or government systems of the waterfall days are still being developed, consumer apps are becoming smaller and less sophisticated. This was a trend that benefited agile methods.

#### 3.4.3.1 Scope and Chunking are Key to Real Usage Feedback

- Chunking is the division of a product's or system's requirements into small groups, each corresponding to a release, which is usually based on features and consists of a set of tasks connected to that feature.

- The chunking of features into a manageable scope for each iteration of the process was crucial to receiving the feedback needed to track changes in understanding that occurred with real-world usage during the software engineering transition from the waterfall process.
- Fig. 3.4.2 shows how tiny the scope of an agile approach really is. The waterfall method, as shown in the diagram, makes a single large pass through the lifecycle activities for the entire system at once.

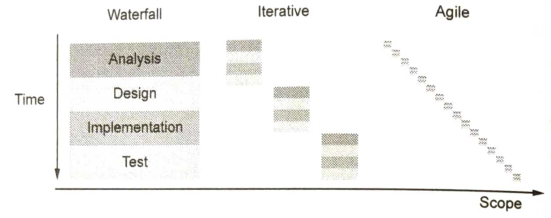


**Fig. 3.4.2 Scope of the waterfall, Iterative and Agile processes**

- A step toward smaller, but still very large, chunks is taken in an iterative technique, which results in many passes. We don't see the true nature of an agile approach - agility - until we get to high numbers of frequent iterations for very small portions.
- It's not so much about smart ideas, technology, foresight, good timing, or even chance that makes the difference. You should instead have a "method for adapting to events as they emerge." Lean UX is a type of agile UX that focuses on delivering a minimum viable product (MVP) at the end of each sprint.
- A sprint is a very brief period in an agile software engineering process during which "a usable and potentially releasable product increment is created". "Each sprint has a goal for what has to be produced, as well as a design and flexible plan to guide the work, and the resulting product increment".
- In an agile SE setting, a sprint is the basic unit of work. In a nutshell, it is an iteration that occurs in conjunction with a release.
- Every iteration, Agile SE procedures deliver relevant small-scope product or system chunks of capabilities and working software features to the client and / or users.
- That way, we will never get too far down a dead end before we can correct our course. Agile procedures meet, the gap between reality and the designer's understanding of the same must remain small in a successful project by constantly narrowing the gap between perceived and actual needs.





### 3.4.3.2 Having Measure of Agility without Chunking

- Iteration on a regular basis, feedback, learning, and adaptability to change have always been fundamental goals in UX. In reality, these are the core principles of the UX design field. We didn't need to chunk to achieve them in UX.
- We use quick prototypes to meet the scope of a successful project should be short, so that the time it takes to deliver a release is kept to a minimum by reducing the time it takes from conception to feedback.
- To achieve the distance between reality and the designer's understanding of the same must be kept short in a successful project, there have been numerous strategies to maintain the gap between user and designer as short as possible. The client reviews all of our models and work items. Users are invited to contribute in design brainstorming and sketching. We use low-fidelity prototypes to conduct early and frequent evaluations.
- In other words, we can imitate the user's experience at a huge scale and learn from it without having to develop the actual system. User feedback is not normally sought one feature at a time in UX. A top - down strategy with a broad scope is preferable for achieving and evaluating a high - quality user experience.
- The only method to determine true requirements is to get feedback from actual users and it was met by using prototypes to simulate real - world usage. Finally, we used participatory design to meet prerequisite that feedback on requirements must be communicated effectively in order for a project to succeed. Furthermore, all of our artifacts are about user problems, not system concerns.

### 3.4.3.3 SE Team and User-facing Prototypes

- In some ways, the UX team has had it easier. On the SE side of things, the most of the issues are system-related. As a result, their artifacts tend to be abstract and technical representations of the inner workings of the system.
- We're dealing with what the user sees and feels on the UX side of things. As a result, mocking up these user-facing components with prototypes makes it simple for us to give the user a sample of a particular design approach. The SE team was unable to provide relevant low - fidelity mockups. They had to develop the entire operational system to show something to people in the preagile days.

### 3.4.3.4 SE Team not interested in Users

- SEs were not recognised for interacting with users about envisioned behaviour with simply a sketch or mockup prior to agile procedures. They didn't have any user touchpoints that were focused on the product side of things to help users understand where they fit into the design solution domain.
- Rather, SE professionals have traditionally concentrated on the process side of things, as well as technical issues like code structure, comprehension, and reusability.
- They were more interested in making things better for the programmer than for the end user. They could change their strategy to delivering usable chunks and receiving product-oriented feedback.

### 3.4.3.5 Why UX followed SE into an Agile Approach ?

- There is a practical reason why UX designs may still need to be chunked for delivery to SE personnel : to keep up with agile SE sprints. The people who work for SE are the ones who create the systems.
- Even if we give them the entire system's design at once, they'll build it in pieces. As a result, agile UX has evolved to meet that development style. We now give UX design portions to the client, customers, product owners, and other stakeholders for immediate feedback, as well as to the SE team for implementation.

## 3.4.4 The Funnel Model of Agile UX

- We designed the funnel model of agile UX to visualize UX design activities before synchronizing with agile SE sprints and after syncing with SE sprints due to confusion about how the UX designer should work in conjunction with an agile SE process.

### 3.4.4.1 Why a New Model was Needed ?

- Agile UX became a means to include user experience design into agile SE. However, there were issues with the way agile UX was tackled at first:
  - Being nimble was seen as being able to move quickly.
  - There is a fundamental misalignment with UX issues while following agile SE flow in sprints.





**Speed kills : Rapidness and agility are not the same**

- People frequently conflate agility with speed. While an agile process is frequently quick, agility is not characterized by speed. Being agile involves chunking design and delivery so we can react to new lessons gained through usage, not just working faster.
- While agile procedures are virtually always linked with speed, warns that a single-minded concentration on speed is usually always detrimental to the quality of the ensuing user experience.
- Putting speed above all else is a risky reaction to time constraints, as it's "proved to produce Frankenstein UIs in just two to three iterations, That's what we call speed". To use an illustration, if you're driving a car and you're getting lost, going faster would not help.

**The single biggest problem : UX was expected to follow the agile SE flow completely**

- The SE world has become almost entirely agile, but the UX sector has failed to keep up. Many individuals believed that in order to stay up with rapid SE development, agile UX workflow should exactly follow agile SE flow. As a result, the UX teams began to produce portions of UX design.
- A successful UX design, on the other hand, is holistic, cohesive, and self-consistent, and these new agile UX practitioners hadn't done the essential groundwork to develop an unified perspective. By the time they caught up with the SE sprints, the "design" had become jumbled.
- Agile techniques are typically used in design-hostile workplaces. And UX is all about design. Agile methodology, which embodies the polar opposite of a holistic perspective, can readily create fragmentation.
- What was needed was a means for UX to do large - scale user research and conceptual design before settling into a small-scale SE routine. This problem was mentioned in the literature, although it was primarily considered as an exception or unique circumstance. We illustrate how to include some upfront usage research and design in a mainstream concept of agile UX in the next section's "funnel model of agile UX."

### 3.4.4.2 Introducing the Funnel Model of Agile UX

- Fig. 3.4.3 depicts the agile UX funnel model, which is divided into two parts : the early funnel on the left and the late funnel on the right.

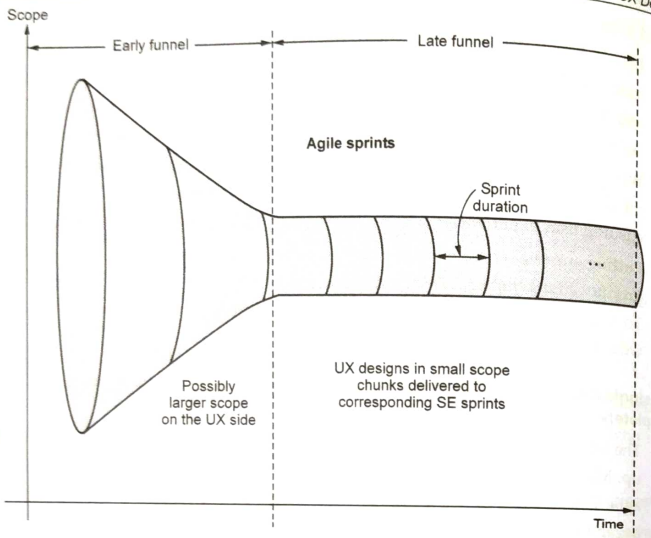


Fig. 3.4.3 Funnel model of agile UX

**Scope in the funnel model**

- The diagram's vertical dimension is called scope. At any point on the funnel, a larger funnel diameter indicates a larger scope. At that moment, a tiny diameter equals a smaller scope.
- Fig. 3.4.3 depicts a typical situation in which the early funnel's scope is greater than the late funnel's scope.

**Speed and rigor in the funnel model**

- The horizontal dimension of the diagram is time, which represents the length of time it takes for operations in the funnel to complete. The stripes or segments on the funnel visually indicate iterations or sprints, and the length of a segment symbolizes the duration of that sprint in time, and, by extension, the speed of procedures and techniques that must be applied in that iteration.
- Longer sprints are usually associated with greater rigour, necessitating more comprehensive and meticulous methodologies and techniques for that iteration (Fig. 3.4.4).





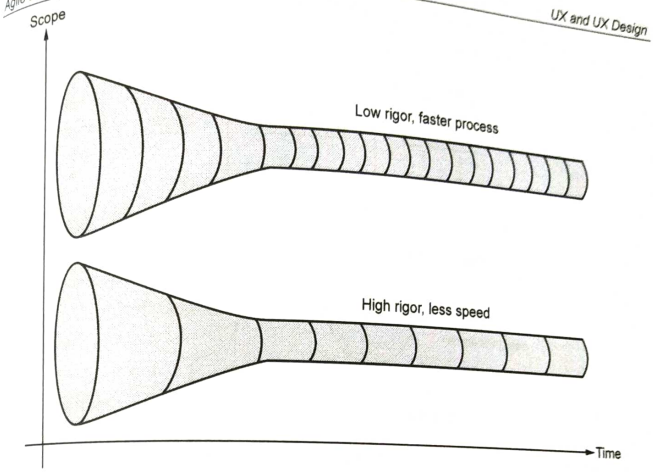


Fig. 3.4.4 Low rigor and high speed Vs high rigor and low speed

### 3.4.4.3 Late Funnel Activities

- The late funnel, or "spout" on the right side of Fig. 3.4.3, is where the agile UX and agile SE processes coexist. The goal of both the UX and SE sides is generally defined in terms of little portions supplied inside a small scope in a short amount of time.
- Each provided software chunk is tested until it works and then integrated with the rest, at least in theory. As a result, regression testing is performed until everything works. The goal is to have something that works at the start and conclusion of each iteration. Continuous feedback is generated by constant and intimate communication.

**Syncing agile UX with agile SE sprints**

- The UX team delivers a design chunk, which the SE team implements in a series of sprints alongside its design of the related functionality (Fig. 3.4.5).

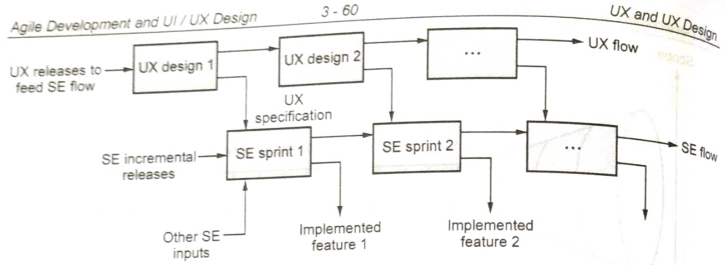


**Fig. 3.4.5 Syncing agile UX with agile SE**

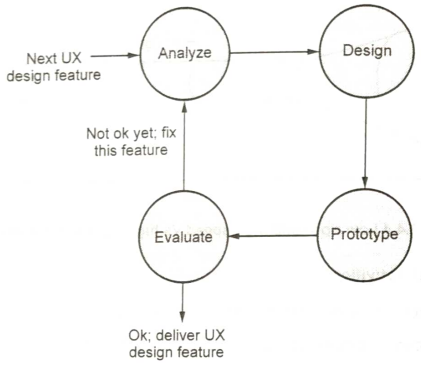


**Fig. 3.4.6 Mini agile UX lifecycle process**

- The idea of agile UX is to sync with agile SE by providing UX design for each feature in turn, as shown in Fig. 3.4.6. Once agile SE gets into the rhythm of going through sprints to produce implementations of chunks as features, the idea of agile UX is to sync with agile SE by providing UX design for each feature in turn.
- A tiny UX lifecycle, such as the one shown in Fig. 3.4.6, is included in each UX design box.

### 3.4.4.4 Early Funnel Activities

- We have to start with a full-scope analysis and design up front in the early funnel before we can execute small-scope incremental releases in the synchronized late-funnel flow of Fig. 3.4.3. To understand the ecology and demands, and to build a conceptual design, UX must begin the design of a new system from the top down.





- Because of the nature of UX design, this is a requirement. UX design, unlike code, is not hidden from the user. Every release, code can be restructured and refactored, making it adaptable. Users will be enraged if the UI is redesigned in this manner.
- This upfront UX activity, which addresses the second challenge and is frequently referred to as "sprint 0" because it occurs before the first sprint in the late funnel, aims to establish :
  - An overview, a skeleton on which the features can be attached.
  - A well - thought - out conceptual design that will guide the development of the features.
  - A top - down design at first.
- This will include as many of the following actions as are required to understand demands and workflow :
  - Usage research.
  - Analysis.
  - Modeling.
- This early funnel effort is what moves "UX strategic design decisions up front, where they belong" to the front of the funnel.

**Need to establish a conceptual design**

- A high - level model or theme of how the system functions is referred to as a conceptual design. It serves as a platform for users to build their own mental models of how the system works.
- If you're working on a project to design and create a whole new system or a new version of an existing one that's perplexing, you'll need to start with a clear conceptual design that will serve as the foundation for a cohesive overall UX design. By definition, conceptual design is a broad concept. At this phase, a limited scope is likely to result in a disjointed conceptual design.
- After establishing the basic design, the UX team can go to a small scope in the late funnel, delivering specific UX designs of individual elements to the SE side in modest increments.
- **Example :** Developing a new smartphone design from scratch
- Assume you're working on a project to develop a completely new smartphone to compete with the present market leaders. It has been concluded that this will necessitate a completely new and original conceptual design - one that is better and more intriguing to customers than present market possibilities.

- This situation may necessitate a large - scale work in the early funnel to produce a thorough conceptual design, establish the smartphone's general ecology, and create a cohesive design.
- It is just impossible to build a phone operating system or a brand-new consumer-facing app without starting with a large - scale UX design. After you have developed a consistent conceptual design, the project can move into the late funnel with a smaller scope to provide the UX design.
- In this situation, it is worth noting that even after the UX and SE responsibilities align in the late funnel and release pieces of the smartphone operating system, end customers may not see those chunks. Because you can't release a new smartphone to end customers in pieces, the learn - through - frequent - customer - releases strategy won't work in this scenario.

**Small systems with low complexity**

- Small systems with low complexity are an example of using a broad scope, but for a different reason : the system is simple and can be handled in one go.
- Small systems with little complexity does not usually benefit from a UX approach with a limited scope. As a result, you can use a broad scope across the funnel.
- If the system isn't large enough, the UX team may not have enough of a "pipeline" of features or complexity to chunk the ideas into small - scope increments.
- In other words, the system's size and complexity may be readily managed with a wide scope across the entire funnel and supplied to the SE side in large scope, allowing them to breakdown it into features for their own small - scope implementation if needed.

**SE needs a funnel model, too**

- On the SE side, some upfront study and design is also required, at the very least to establish the system architecture. So it is possible that there are two overlapping funnels.

**Nexus of early and late parts of the funnel**

- The move to the late funnel is a critical phase in a project; it marks the start of "crunch time," when you get your act together and align with the agile SE team. Now is the time for user stories to drive your design iterations, but the design is informed by the deep understanding that you gained through usage research and modelling.





## 3.5 Review Questions

1. *Explain UX and UX design.* **(Refer sections 3.1 and 3.1.3)**
2. *Explain components of UX.* **(Refer section 3.1.4)**
3. *What UX is NOT ?* **(Refer section 3.1.5)**
4. *Explain kind of interaction and UX.* **(Refer section 3.1.6)**
5. *Explain basic process components of UX.* **(Refer section 3.2.2)**
6. *Explain fundamental UX lifecycle activities.* **(Refer section 3.2.3)**
7. *Explain UX design techniques as life skills.* **(Refer section 3.2.4)**
8. *What is rigor in a UX method or process ? Explain in detail.* **(Refer sections 3.3.2)**
9. *Explain scope of delivery in UX process.* **(Refer section 3.3.3)**
10. *Differentiate commercial product perspective vs enterprise system perspective in UX design.* **(Refer section 3.3.4)**
11. *What are the challenges in building systems ?* **(Refer section 3.4.1)**
12. *Explain an agile lifecycle process.* **(Refer section 3.4.3)**
13. *Explain the funnel model of agile UX.* **(Refer section 3.4.4)**

❑❑❑