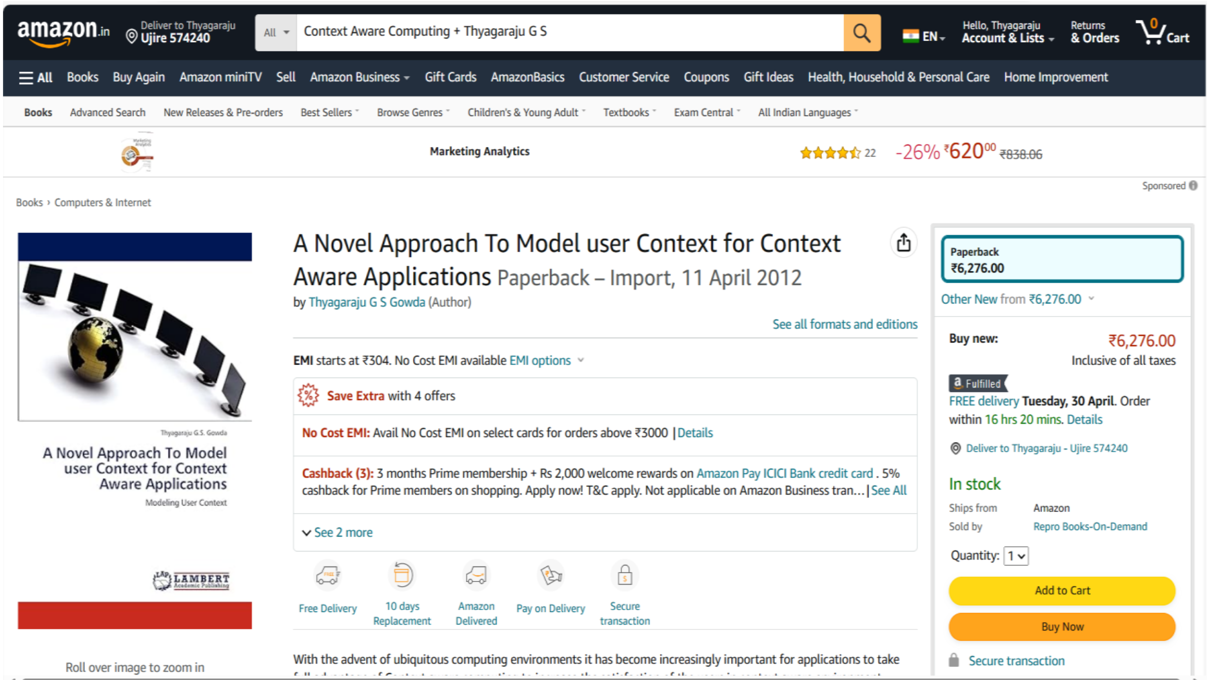
Module1_Chapter1 : Introduction

1.1 What is UI/UX?

UI/UX stands for User Interface (UI) and User Experience (UX), which are critical aspects of designing digital products like websites, mobile apps, software, and more.

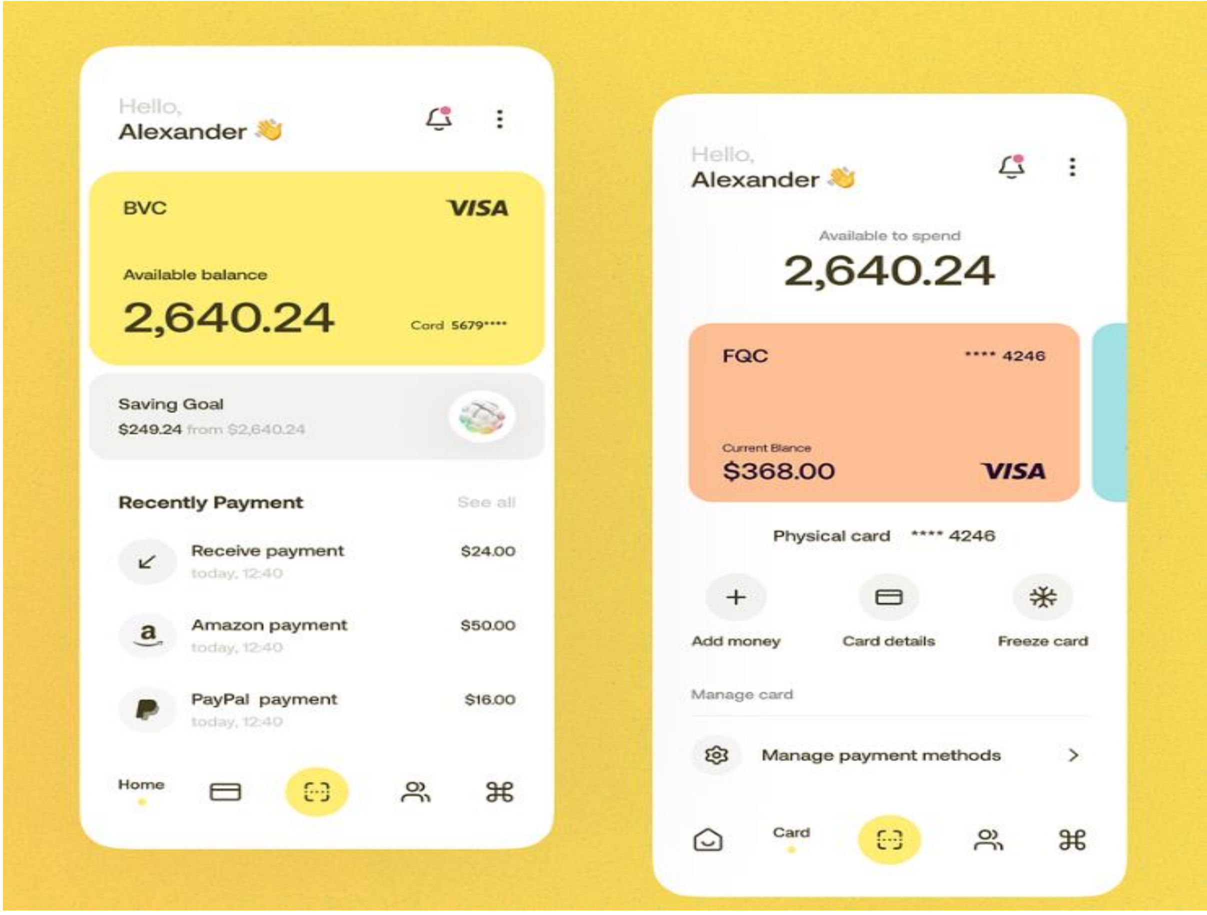
- **UI (User Interface)** refers to the visual elements and components that allow users to interact with a digital product. It encompasses the layout, typography, icons, buttons, menus, and other visual elements that users see and interact with directly. The UI plays a crucial role in determining how a product looks and how users navigate through its features.
- **UX (User Experience),** on the other hand, is a broader concept that encompasses the overall experience a user has while interacting with a product. It encompasses not only the visual elements (UI) but also the usability, accessibility, performance, and overall satisfaction that the product provides. UX design aims to create products that are intuitive, efficient, and enjoyable to use, considering the user's needs, goals, and behaviors.

Example1: E-commerce website like Amazon



- The **UI elements** would include the search bar, navigation menus, product images, buttons (like "Add to Cart" or "Buy Now"), and the overall layout of the website. These visual elements allow users to browse products, search for items, and complete purchases.
- **The UX,** however, encompasses the entire journey a user goes through when using the Amazon website. It includes factors like the **ease of navigation**, the **clarity of product information**, the **speed of loading pages**, the efficiency of the **checkout process**, and even the overall satisfaction and trust users have in the brand. A good UX design ensures that users can find what they're **looking for quickly, complete their purchases smoothly**, and have an overall positive experience that encourages them to return to the website in the future.

Example2: Mobile banking app



- The UI design includes the layout of the login screen, the arrangement of buttons for different banking services (like checking balance, transferring funds), the color scheme used for different sections (green for successful transactions, red for errors), and the overall aesthetic appeal of the app.
- UX design focuses on how easy it is for users to navigate through different sections, how intuitive the interface feels (e.g., using familiar icons for actions), ensuring security features are reassuringly integrated without being intrusive, and addressing any pain points users might encounter (such as a clear error message and guidance if a transaction fails).

1.2 Relationship Between UI and UX:

UI is part of UX: The UI design contributes to the overall UX by ensuring that the interface is visually appealing, easy to use, and aligns with user expectations.

UX guides UI: UX research and analysis help designers understand user needs, which in turn informs the UI design decisions to create an interface that is functional and enjoyable.

In summary, UI/UX design is about creating digital experiences that are visually appealing, intuitive to use, and aligned with users' needs and expectations. Good UI/UX design leads to higher user satisfaction, increased engagement, and better overall performance of digital products.

1.3 Ubiquitous Interaction

UI/UX (User Interface/User Experience) and Ubiquitous Interaction are related concepts, but they are not entirely synonymous. UI/UX primarily focuses on the design and interaction between a user and a specific digital product or system, while Ubiquitous Interaction is a broader concept that encompasses the idea of seamless and natural interaction between humans and various computing devices and services embedded in our everyday environments.

UI/UX principles and practices are essential for designing effective and intuitive interfaces that enable seamless and natural interactions between users and computing devices or services embedded in their everyday environments, which is the core goal of Ubiquitous Interaction.

Following section discusses how the user interfaces, concept of computing and concept of interaction are changing towards ubiquitous interaction:

1. **Desktops, Graphical User Interfaces, and the Web Are Still Here and Growing**:

The “old-fashioned” desktop, laptop, and network-based computing systems are alive and well and seem to be everywhere, an expanding presence in our lives. And domain-complex systems are still the bread and butter of many business, industry, and government operations. Most businesses are, sometimes precariously, dependent on these well-established kinds of computing.

Web addresses are commonplace in advertisements on television and in magazines. The foreseeable future is still full of tasks associated with “doing computing,” for example, word processing, database management, storing and retrieving information, spreadsheet management. Although it is exciting to think about all the new computing systems and interaction styles, we will need to use processes for creating and refining basic computing applications and interaction styles for years to come.

2. **The Changing Concept of Computing**:

The realm of computing has transcended conventional desktops and laptops, graphical user interfaces, and the Web, expanding into diverse environments. Nowadays, computer systems are integrated into clothing, appliances, homes, offices, entertainment systems, vehicles, and infrastructure like roads. Computation and interaction are extending to walls, furniture, and everyday objects such as briefcases, purses, wallets, watches, PDAs, and cellphones.

In projects like 2Wear (Lalis, Karypidis, & Savidis, 2005) and eGadget (Kameas & Mavrommati, 2005), mobile computing components connect via short-range wireless communication, adapting system behavior to various user devices and contexts. Wearable computers, like those experimented with at MIT involving soldiers outfitted with sensors to monitor health metrics (Zieniewicz et al., 2002), exemplify the integration of computation into personal attire and accessories.

Innovations like "Smart-its" (Gellersen, 2005) embed microprocessors, sensors, actuators, and wireless connectivity into everyday objects, enhancing their functionality in human activities. For instance, car keys equipped with tracking technology assist in locating misplaced items, and low-cost machine-readable identifiers can monitor changes in products like milk and groceries. This technology enables remote inventory checks, like querying a fridge via a mobile phone (Ye & Qiu, 2003), prompting users to replenish items.

This pervasive computing largely occurs without traditional interfaces like keyboards or monitors, as emphasized by Cooper (2004), demonstrating that interaction doesn't always necessitate standard input methods.

Commercially, such technology is revolutionizing industries. For example, smart railcars (Gershman & Fano, 2005) autonomously manage their status, location, maintenance, and security, promising significant efficiency gains over current manual methods.

Research initiatives, such as those at MIT Media Lab (Paradiso, 2005), are exploring embedded computing's potential in fields like robotics, telemedicine, and prosthetics. Projects like Tribble (Tactile Reactive Interface Built By Linked Elements) use dense sensor networks akin to biological skin, capable of sensing and responding to stimuli.

Notably, robotics is expanding beyond household tasks (Scholtz, 2005) into healthcare, eldercare, museum curation, urban rescue, and space exploration, illustrating the increasing integration of advanced computing technologies into diverse applications.

3. The Changing Concept of Interaction:

When we use a desktop or laptop, we know we're doing computing—like exchanging information, working, learning, playing, or exploring. But when we drive a car with its built-in computer and maybe a GPS, we don't think of it as "computing." Tscheligi (2005) echoes Mark Weiser, saying "the world is not a desktop."

One prominent example of computing away from traditional setups is seen in mobile communications. Mobile devices are rapidly growing in popularity, emphasizing user experience quality. The way these devices are named—like "cellphone" versus "mobile phone"—reflects attitudes towards users.

Weiser (1991) suggested that the best technologies are the ones that fade into the background. Russell, Streitz, and Winograd (2005) discuss the concept of "disappearing computers," which become seamless and unobtrusive like electric motors in machines.

In the realm of ambient intelligence, computers blend into our living spaces. Philips Research's HomeLab envisions technology becoming part of our daily social interactions seamlessly.

Even though computers seem to vanish into walls and objects, the challenge remains to make interactions natural and user-friendly. Poorly designed embedded systems could leave users lost without familiar interfaces like menus and icons.

Some propose using smell for human-computer interaction (HCI), arguing that smell is underused in technology despite its importance in daily life. Interaction in HCI involves mutual influence, not just with computers but also with the environment and other systems.

For example, a "smart wall" can sense and respond to users without explicit actions, controlling inputs based on user presence or identification. Systems can also react to their environment independently, like adjusting temperature based on sensors, without direct user involvement.

Road signs are another form of interaction—they guide drivers within the larger highway system. This broader view of interaction encompasses human-machine interactions with devices like telephones and ATMs, as well as interactions with the physical world, like navigating a museum's layout.

1.4 Designing for a Quality User Experience in 3DApplications

- **Motion controls** and **freehand gestures** are prevalent in modern interfaces.
- 3D interaction is gaining popularity across g***aming, home theaters, and mobile applications***.

Designing for a quality user experience in 3D applications involves exploring various forms of interaction, such as motion controls and freehand gestures, which are becoming prevalent in modern interfaces. These interactions aim to mimic real-world actions, leading to "natural" or "high-fidelity" 3D user interfaces (3DUIs). For instance, using physical movements like turning and walking to navigate virtual environments can enhance spatial understanding. However, natural 3D interaction has limitations, as it can be challenging to replicate real-world actions precisely, potentially leading to misunderstandings.

To address these challenges, "magic" 3D interaction techniques can be employed to allow users to perform tasks more efficiently and creatively within virtual worlds. These techniques extend beyond real-world limitations, enhancing physical, perceptual, and cognitive abilities. Despite this freedom, it's crucial to provide effective constraints in 3DUIs to guide users and ensure ease of interaction. For example, in an interior design application, allowing users to position furniture freely in 3D space should be constrained to practical parameters like floor placement and vertical rotation.

Furthermore, well-designed 3DUIs prioritize user comfort, considering the physical demands and potential discomfort associated with large-scale movements and immersive displays. Designers should implement techniques that minimize user fatigue and discomfort, such as allowing interactions with arms supported against the body or physical surfaces. By applying these principles and understanding the fundamentals of human-computer interaction (HCI) and UX design, designers can create engaging, enjoyable, and productive 3D user experiences.

Key Guidelines for Designing 3D User Interfaces

1. Natural 3D Interactions
    - Replicating real-world actions (walking, swinging golf club) / Easy to understand, high spatial awareness
2. "Magic" 3D Interactions
    - Enhance physical/perceptual/cognitive abilities
    - Allow superhuman capabilities in virtual world
    - More efficient than natural interactions
3. Provide Helpful Constraints
    - Guide users for easier, more precise interactions E.g. furniture only on floor, not free 3D space
4. Design for User Comfort
    - Avoid fatigue from extended 3D movements/ Allow arm/body support during manipulation
    - Minimize rapid motions causing dizziness / Proper viewing for stereoscopic displays
5. Follow UX Principles
    - Apply fundamental HCI and UX design principles / But utilize 3D-specific research & guidelines / Enable engaging, productive 3D experiences

1.5 Emerging Desire for Usability

In the past, computer use was niche, embraced by technically adept users who saw poor usability as exclusive and protective. This mentality persisted even as mainstream users sought better experiences. "Dancing bear" software exemplifies this, where functionality triumphs over flawed design, fostering resistance to change based on success alone.

["Dancing bear" is a term used to describe software that is poor quality but still used out of necessity. The term implies that users will tolerate flaws and frustrations in the software. For example, Nick Hayward describes "dancing bearware" as software that is "rubbish" but that people have to use anyway.]

The narrative critiques public unawareness of user experience's role in design, seen in blame on poll workers for voting machine failures instead of questioning machine complexity.

Douglas Adams humorously criticizes airport design for prioritizing efficiency over aesthetics, reflecting a shift towards tech for meaningful, enjoyable interactions.

Industry practices like disclaimers on software warranties reveal a lack of accountability. A case study on a flawed police communication system underscores the consequences of ignoring user-centered design, leading to confusion and safety risks. The need for quality UX, though initially complex, benefits all stakeholders in the long run.

1.6 FROM USABILITY TO USER EXPERIENCE

1.6.1 The Traditional Concept of Usability

Usability refers to the aspects of human-computer interaction that ensure the interaction is **effective, efficient, and satisfying for the user**. It includes characteristics like ease of use, productivity, efficiency, effectiveness, learnability, memorability, and user satisfaction.

1.6.2 Misconceptions about Usability

Some common misconceptions about usability include:

- It is not about "**dummy proofing**" or trivializing the design process.
- It is not solely about making things "**user-friendly**" - users need efficient, effective, safe tools to reach their goals.
- "**Doing usability**" is more than just usability testing - it involves the entire interaction design process.
- Usability is not just about making things visually appealing after development.

1.6.3 The Expanding Concept of Quality in Design

The field of interaction design has grown from an engineering focus on usability/user performance to a **broader concept of user experience**. Defining **user experience** is still evolving, but it aims to account for aspects like **ambience**, **attention**, **aesthetics, social/cultural context etc.**

1.6.4 Emotional Impact vs User Satisfaction

While user satisfaction has been part of traditional usability definitions, in practice it focused more on intellectual responses than emotional impact. As technology becomes more personal, we need a broader definition of quality to include emotional aspects.

1.6.5 Functionality vs User Experience

A better user experience can outsell products with more functionality (e.g. iPhone vs Blackberry). While users expect correct functionality, the interface is their only experience of it. Poor usability means functionality effectively doesn't exist for users.

User experience also considers long-term phenomenological aspects - the cumulative emotional impact as technology becomes part of our lifestyles and meaning-making. Apple's "Time Machine" backup made a mundane task engaging through good design.

The authors argue for separating "do goals" evaluated by pragmatic usability, from "be goals" evaluated by the emotional, hedonic quality of the experience. Both are important for a quality user experience.

1.6.6 A Good User Experience is Not Necessarily High-Tech or "Cool”:

New "cool" high-tech products are often equated with amazing user experience by enthusiasts and the public. However, failed interaction design can quickly turn amazement into annoyance when the product becomes a barrier to use. Cool technology alone does not inherently provide a quality user experience.

The example of Microsoft's packaging design for Vista and Office 2007 products highlights this. While marketed as user-friendly and providing a great experience, actual users found it difficult to open with poor affordances violating design conventions.

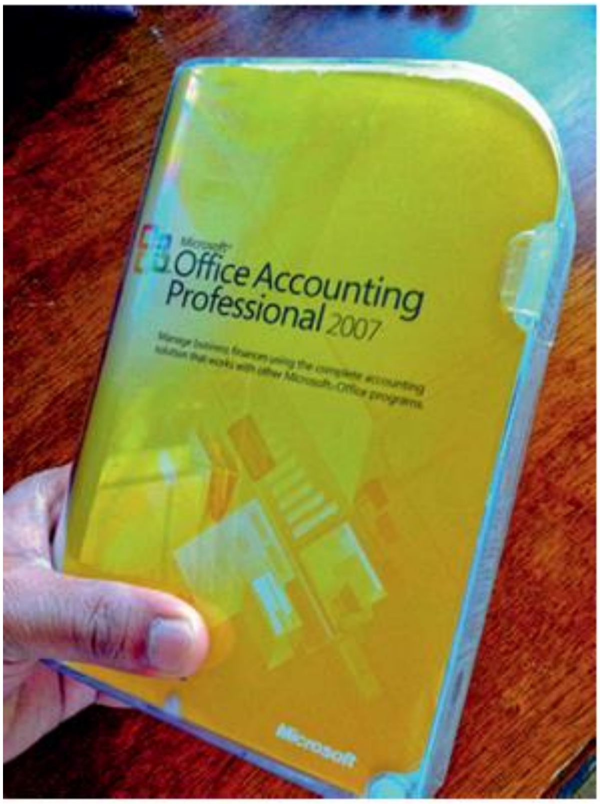


Figure 1-1

oclaims: *A new Microsoft software packaging design.*

1.6.7 Design Beyond Just Technology

This book considers technology as one context for design. The focus is on design principles and guidelines that apply universally - whether for **software interfaces, ATMs, signage or other artifacts satisfying usage needs through a creator-user dialogue**.

1.6.8 Components of User Experience

Usability factors like **ease of use** are still vital, especially for complex work domains where **efficiency** and **effectiveness** matter. User experience embodies these usability implications too. An attractive but cumbersome design undermines the joy of use.

1.6.9 User Experience is (Mostly) Felt Internally

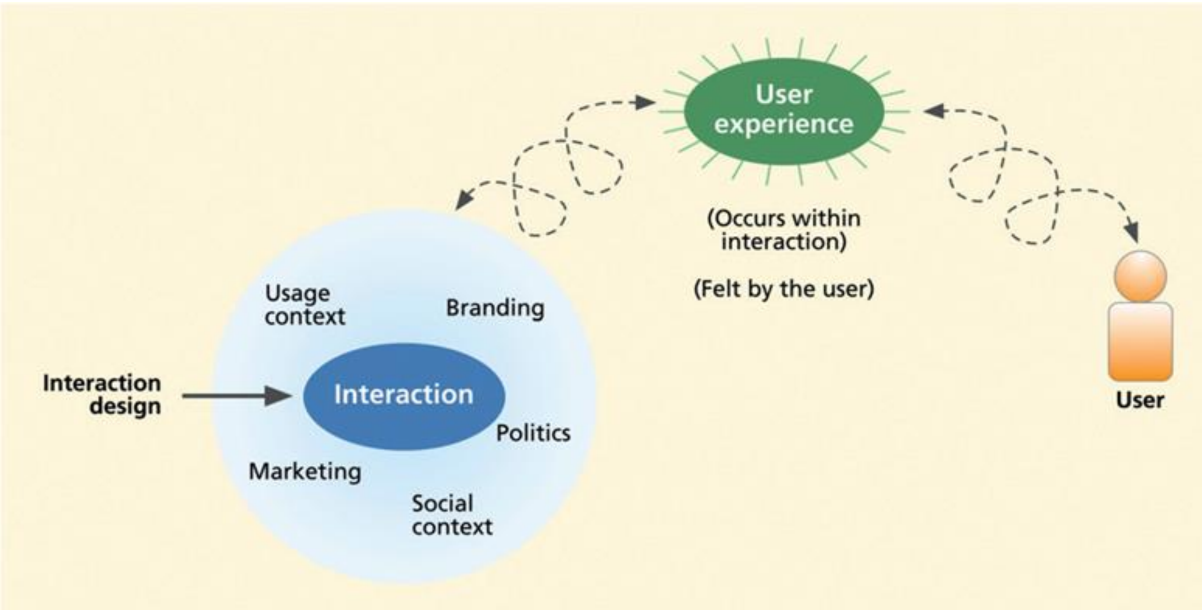
User experience is the totality of effects felt internally by the user from interacting with and using a system/product, including anticipation, actual usage, and memories afterwards.

While performance aspects of usability can be externally observed, user experience generally refers to what the user feels internally, including impacts of pragmatic usability, usefulness and emotional factors.

1.6.10 User Experience Cannot Be Designed

User experience cannot be **directly designed or engineered - only facilitated** through good design. It is revealed within a particular user's context and usage. The same design may lead to different experiences for different users/contexts.

The example of marketing Belgian chocolates highlights this - the experience resides in the **consumer**, not just the product itself, despite marketing claims. Technically, it should be phrased as "**designed to produce**" the desired experience.



1.6.11 Role of Branding, Marketing, and Corporate Culture

The user experience can extend beyond just responses to **usability, usefulness and joy of use**. **Social, cultural, marketing, political aspects, hardware choices** etc. can also influence the experience. Users may get wrapped up in what the manufacturer's brand stands for - their values, practices, image etc.

For some companies, facilitating a quality user experience is ingrained in their corporate DNA and daily practices across all roles. Apple exemplifies this obsession with design elegance permeating everything they do - from product packaging to retail stores aimed at providing an enchanting personal experience.

BMW is another example valuing emotional impact, with their motto "**Joy is BMW**" highlighting the importance of driving pleasure and experience over just engineering prowess.

While difficult to definitively design, this book focuses on what interaction designers can control through processes and guidelines.

1.6.12 Why Have Such a Broad Definition?

The **broad definition recognizes** that user experience can begin before actual usage - from initial product awareness to advertising to purchasing to unboxing. It can persist after usage through memories and sharing with others.

This perspective encompasses the entire cultural and personal experience around a product. The breadth is intentional - it implicitly necessitates collaboration across multiple roles like hardware engineers, visual designers, branding experts and interaction designers. Everyone must coordinate to execute a shared vision for a quality user experience.

While criticized for making it hard to operationalize responsibilities, the broad framing is precisely why it's needed - to drive interdisciplinary efforts toward designing holistic user experiences.

1.7 Enhancing the Visitor Experience through Web Design

In this discussion, I will adopt the definition of "user experience" proposed in this book, emphasizing that it is entirely subjective and resides in the user's mind. As product designers, our goal is to create designs that lead to a positive visitor experience for our target users. Transitioning from designing desktop software to websites has highlighted that user experience is influenced by various design qualities beyond usability alone. As a web user interface designer, I use the term "visitor experience" and recognize the importance of addressing at least five different qualities of websites that impact the experience of site visitors:

1. **Utility**
2. **Functional integrity**
3. **Usability**
4. **Persuasiveness**
5. **Graphic design**

Utility:
Utility refers to the usefulness, importance, or interest of a website's content to its visitors. It is relative and varies among different visitors based on their needs and interests.

Functional Integrity:
Functional integrity measures the extent to which a website works as intended, free from issues like dead links, crashes, or browser incompatibility.

Usability:
Usability relates to how easy it is for visitors to learn and use a website efficiently, irrespective of its utility or functional integrity.

Persuasiveness:
Persuasiveness focuses on how effectively a website encourages desired behaviors, like making a purchase or signing up for a newsletter.

Graphic Design:
The graphic design of a website, including colors, images, and layout, can evoke emotional responses and influence the overall visitor experience.

Designing for an optimal visitor experience requires an interdisciplinary team comprising experts in market research, web development, usability engineering, persuasion psychology, and graphic design. Collaborating effectively among these disciplines is key to creating websites that deliver a positive and engaging experience tailored to the target audience.

1.8 Emotional Impact as Part of the User Experience

1.8.1 The Potential Breadth of Emotional Impact

Emotional impact refers to the affective parts of interaction - **pleasure, fun, aesthetics, novelty, sensations, and experiential features**. The user's reaction can range from mildly satisfied to deeply personal and emotional. Some products spark a deep emotional chord and affinity beyond just form, function and usability - elevating the experience to pure joy akin to appreciating art or music.

1.8.2 A Convincing Anecdote

David Pogue uses the example of the iPad to illustrate the role of emotional allure. Critics initially dismissed it as superfluous since existing devices covered its functionality. Yet, it became wildly successful due to the finely crafted personal experience of using it, not rational utility.

1.8.3 Aesthetics and Affect

Aesthetics moves interaction from a utilitarian to an experiential orientation focused on pleasure and beauty. The relationship between objective aesthetic qualities and subjective aesthetic perceptions affecting emotions is complex. The same aesthetic design can evoke different emotions based on an individual's subjective experience.

1.8.4 The Centrality of Context

Context is central in interpreting emotional impact. The same product is marketed differently (e.g. Garmin GPS) based on contrasting anticipated contexts and associated emotional resonance for different user groups. Designing for desired experiences requires deep user and context understanding.

1.8.5 What about Fun at Work?

While emotional factors like fun are desirable for personal use, their role at work is debated. Some evidence suggests fun can enhance appeal and performance for repetitive tasks. However, fun and usability can conflict - unpredictability

may hinder traditional usability. Some high-focus roles don't allow distractions. Finding the right balance is key.
In summary, emotional impact through aesthetics, fun, pleasure etc. is an important part of the holistic user experience beyond just pragmatic usability and usefulness, though its appropriate extent depends on the specific usage context.

1.9 USER EXPERIENCE NEEDS A BUSINESS CASE

Innovative Design with Seamless Connectivity: The Toshiba Satellite Receiver Box

1.9.1 Is the Fuss over Usability or User Experience Real?

As professionals in this field, we often face challenges in gaining buy-in for user experience processes from upper management and business stakeholders. So, what exactly is the business case for UX?

The need for better design in computer software, especially in user interaction, is undeniable. Mitch Kapor, founder of Lotus, has publicly highlighted that "The lack of usability of software and the poor design of programs are the secret shame of the industry" (Kapor, 1991, 1996). This sentiment is widely shared within the industry. Poor user experience serves as an uncontrolled source of overhead for companies using software, leading to lost productivity, error correction, data loss, learning and training costs, and expenses related to help desks and field support.

Charlie Kreitzburg, founder of Cognetics Corporation, attributes the chaos, waste, and failure in software development primarily to practices that prioritize technology over users. He advocates for the industry to "rethink current software design practice to incorporate user-centered design principles."

These critical assessments are not based on personal opinion alone but on comprehensive surveys conducted by reputable groups in the software industry.

For instance, The Standish Group (Cobb, 1995; TheStandishGroup, 1994, 2001) surveyed 365 IT executive managers across companies of varying sizes and found that neglecting user input is one of the primary reasons many software projects fail, costing corporations $80 billion annually.

Some estimates suggest that over 60% of software projects exceed their budgets (Lederer & Prasad, 1992). According to May (1998), the average software development project is 187% over budget, 222% behind schedule, and implements only 61% of specified features.

A report by Computer World (Thibodeau, 2005) highlights that poorly designed software costs businesses millions annually due to usability challenges requiring extensive training and support, leading many users to underutilize applications. Keith Butler of Boeing notes that usability issues can inflate software ownership costs by up to 50%.

Such reports illustrate the dismal performance of the software development industry. Kwong, Healton, and Lancaster (1998) cite the Gartner Group's characterization that 25% of software development efforts fail outright, and 60% produce sub-standard products. This level of inefficiency would be intolerable in other industries. As Kreitzburg has aptly put it, imagine if 25% of all bridges collapsed or 25% of all airplanes crashed.

1.9.2 No One Is Complaining and It Is Selling Like Hotcakes

It's easy to mistake positive signs, such as strong sales, as indicators that a product has no user experience problems. Managers often assert, "This system has to be good; it’s selling big time." "I’m not hearing any complaints about the user interface." However, these may not be reliable indicators of a product's user experience quality. Sometimes, project managers are the only ones not hearing user experience complaints. Additionally, despite demands for an improved user experience, some users simply will not complain.

If you're uncertain about the user experiences with your product, but your users are not complaining, watch for certain indicators that suggest potential usability and user experience issues:

- Your users access only a small portion of the overall functionality your system offers.
- There's a significant number of technical support calls about how to use a particular feature in the product.
- Requests are made for features that already exist in the product.
- Your competitor's products sell better even though your product has more features.

This book can help you address these issues. It is designed for those who recognize the importance of a good user interface and want to learn more about what a quality user experience means, how to ensure it, and how to recognize when you have achieved it. This book is especially aimed at practitioners—people who put theory into practice in a real-world development environment. The methods and techniques described here can be used by anyone involved in any part of the development of a user interaction design for a user interface.

1.9.3 A Business Strategy: Training as a Substitute for Usability in Design

"It might not be easy to use right off, but with training and practice, it will be a very intuitive design." While this may sound absurd, it reflects what many people imply when they suggest training as a way to fix usability problems.

Unfortunately, a real-world example often involves large governmental organizations attempting to solve user experience problems by issuing "instructional bulletins" to all field users. These bulletins present real user experience challenges, increasing task time, introducing significant opportunities for errors, and requiring users to memorize special-case instructions for various situations. Moreover, these bulletins are issued only once, leaving users responsible for understanding complex contents, even if they were hired after the bulletins were issued.

In one such case, a critical situation arises when an applicant calls an 800 phone number, and an agent, acting as an information intermediary, denies certain information. Screens referred to in the "instructional bulletin" about this interaction must be completed properly to comply with policy and ensure that applicants receive required notices. Will a user remember these complex instructions months after they were issued?

Training as a substitute for usability is an ongoing per-user cost that often fails to meet the goals of increased productivity and reduced risk, errors, and costs. The real question is, how could someone send out such a memo with a straight face? How could the memo author not see the folly of the situation? Perhaps they had been part of the bureaucracy for so long that they truly believed "this is how we have always done it."

1.10 ROOTS OF USABILITY:

The origins of computer usability are debated, but it evolved from earlier work on usability for non-computer machines in design and human factors. Interest in computer usability grew in the late 1970s and early 1980s, leading to dedicated conferences on the topic. Human-computer interaction (HCI) and usability drew inspiration from various fields such as **psychology, psychometrics, systems engineering, and computer science.**

1.10.1 A Discipline Coming of Age: Compared to established fields like architecture or civil engineering, computer science and human-computer interaction (HCI) are relatively young. The earliest computer science departments are only a few decades old, and personal computers have been around for about 30 years. HCI has rapidly evolved since its inception.

The early roots of HCI can be traced to work at Virginia Tech and other universities in the late 1970s and 1980s, as well as at institutions like IBM and the National Bureau of Standards. Significant progress began with unofficial conferences in the early 1980s and later culminated in the CHI conferences in Boston starting in 1983. These gatherings marked a key point in HCI's development.

1.10.2 Human Factors and Industrial and Systems Engineering: Human factors, a field focused on human interaction with systems, originated in the early 20th century with efforts to improve industrial efficiency, notably by Frederick

Winslow Taylor. It expanded to include safety in complex systems like airplane cockpits, which eventually influenced HCI by emphasizing user-centered design to prevent errors.

1.10.3 Psychology and Cognitive Science: Psychology played a significant role in shaping HCI, with cognitive psychology providing insights into human behavior and cognition. Early HCI work integrated concepts from psychology, especially in modeling user behavior and understanding cognitive processes during interaction with computers.

1.10.4 Task Analysis: Task analysis, a method to understand user tasks and interactions, predates HCI and was refined to support interaction design. It helps designers build predictive models of user performance and informs interface design decisions.

1.10.5 Theory: HCI's foundation lies in psychological and cognitive theories, adapted to model user behavior and interaction with computers. Theories like the Model Human Processor and GOMS have been instrumental in understanding user cognition and performance.

1.10.6 Formal Methods: Formal methods, though not strictly theory, are used in HCI to create precise design specifications that can be analyzed for correctness and consistency. They support both theoretical understanding and practical implementation of interaction designs.

1.10.7 Human Work Activity and Ethnography: Work activity theory and ethnography have influenced HCI by providing tools to study work practices and gather design requirements based on real-world contexts.

1.10.8 Computer Science: Interactive Graphics, Devices, and Interaction Techniques: Advancements in computer graphics, interaction styles, and user interface technologies have shaped HCI by enabling practical implementations of interaction designs, including direct manipulation techniques that changed how users interact with computers.

1.10.9 Software Engineering: Software engineering shares parallels with HCI in development lifecycles, although the disciplines historically operated independently. Integration of usability concerns into software engineering processes is essential for creating effective interactive systems

Key Terms

User Experience:

User experience is the totality of the effect or effects felt by a user as a result of interaction with, and the usage context of, a system, device, or product, including the influence of usability, usefulness, and emotional impact during interaction, and savoring the memory after interaction. "Interaction with" is broad and embraces seeing, touching, and thinking about the system or product, including admiring it and its presentation before any physical interaction.

Usability:

Usability is the pragmatic component of user experience, including effectiveness, efficiency, productivity, ease-of-use, learnability, retainability, and the pragmatic aspects of user satisfaction.

Usefulness:

Usefulness is the component of user experience to which system functionality gives the ability to use the system or product to accomplish the goals of work (or play).

Functionality:

Functionality is power to do work (or play) seated in the non-user-interface computational features and capabilities.

Emotional Impact

Emotional impact is the affective component of user experience that influences user feelings. Emotional impact includes such effects as pleasure, fun, joy of use, aesthetics, desirability, pleasure, novelty, originality, sensations, coolness, engagement, novelty, and appeal and can involve deeper emotional factors such as self-expression, self-identity, a feeling of contribution to the world, and pride of ownership.

Contextual Inquiry

Contextual inquiry is an early system or product UX lifecycle activity to gather detailed descriptions of customer or user work practice for the purpose of understanding work activities and underlying rationale. The goal of contextual inquiry is to improve work practice and construct and/or improve system designs to support it. Contextual inquiry includes both interviews of customers and users and observations of work practice occurring in its real-world context.

Phenomenological Aspects of Interaction

Phenomenological aspects (deriving from phenomenology, the philosophical examination of the foundations of experience and action) of interaction are the cumulative effects of emotional impact considered over the long term, where usage of technology takes on a presence in our lifestyles and is used to make meaning in our lives.

"UX"

"UX" is an almost ubiquitous term that we use to refer to most things that have to do with designing for a high quality user experience. So this means we will use terms like the UX field, UX work, a UX practitioner, the UX team, the UX role, UX design or UX design process.

Chapter 2: Extracting Interaction Design Requirements

1. **Introduction** : Contextual Analysis , Requirements and Gap between Analysis and Design
2. Needs and Requirements
3. Formal Requirements Extraction
4. Abridged Methods for Requirements

2.1 Introduction : Contextual Analysis , Requirements and Gap between Analysis and Design

2.1.1 What is WAAD?

After doing contextual analysis user needs and requirements should be understood. The requirements gathering WAAD can be utilized. WAAD stands for "Work Activity Affinity Diagram". It refers to a diagram or representation that was created by organizing and grouping together related work activity notes obtained from contextual inquiries and observations of users in their work environment.

The WAAD provides an overview and understanding of the users' current work practices, tasks, concerns, and usage contexts within their work domain. It does not yet contain actual design requirements or specifications for a new system, but rather serves as a foundation for extracting and identifying those requirements based on the insights gained from studying the users' existing work activities.

The work activity notes in the work activity affinity diagram (WAAD) are not actual designs or requirements.

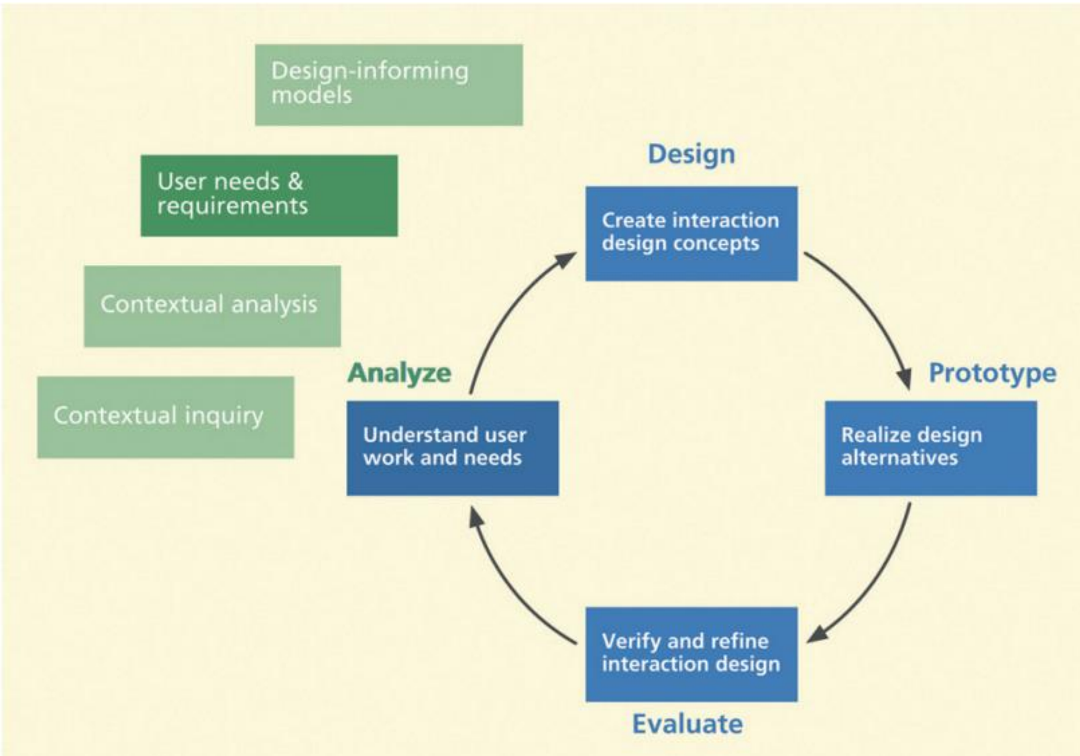
Except for a few cases where users directly commented on a specific need or requirement, the work activity notes in the WAAD represent the users' work domain, their concerns, and their current usage.

The contextual inquiry and analysis performed so far provide an accurate and complete picture of the users' work domain.

Now, the task is to identify the needs and design requirements for a proposed new system to optimize, support, and facilitate work in that domain.

This involves carefully analyzing the WAAD and any preliminary design-informing models, such as the flow model, to extract the user needs and requirements.

By extracting the user needs and requirements from the WAAD and models, the first step in constructing the bridge (between the current state and the proposed system) is taken.

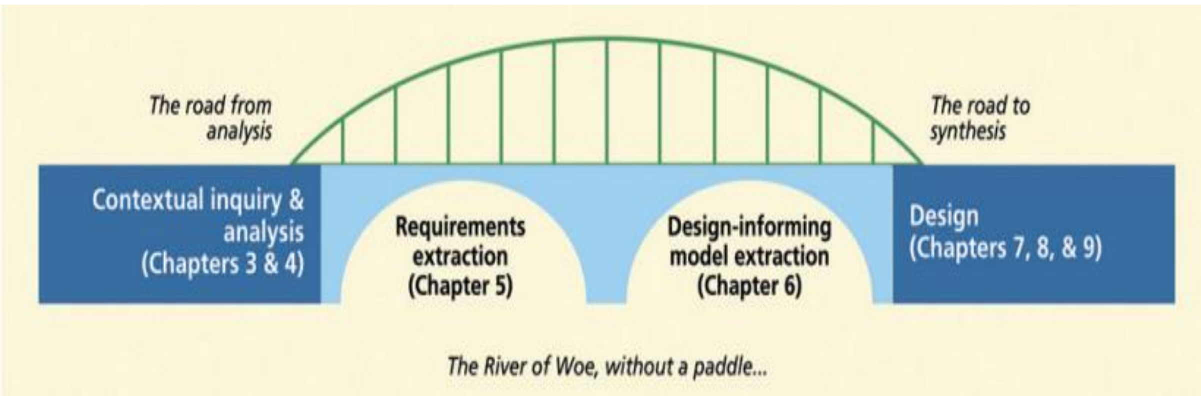


2.1.2 Gap between Analysis and Design

The key gaps between the analysis phase (contextual inquiry and understanding existing work practices) and the design phase (producing designs for a new system) are:

1. Output mismatch: The output of contextual inquiry and analysis describes the existing work domain, but does not directly provide the information needed as inputs for design.
2. Cognitive shift: There is a cognitive shift required in moving from analysis-oriented thinking (focused on understanding current practices) to design-oriented thinking (focused on envisioning new systems and work processes).
3. Transition from old to new: The gap represents the demarcation or separation between studying the old (existing work practices and systems) and envisioning the new (designing a new work space and system).
4. Information needs differ: The information gathered from contextual studies about the work domain does not inherently meet the specific information needs required for design activities.
5. Translating insights: There is a need to translate the insights and understanding gained from the analysis phase into actionable inputs and requirements that can inform the design of a new system.
6. Bridging the gap: Specific efforts and methods are required to bridge this gap between analysis and design, to ensure that the user research and contextual inquiry effectively inform the design process.

In summary, the key gaps highlighted are the differences in output, cognitive approaches, focus (old vs. new), and information needs between the analysis and design phases, necessitating a intentional transition and translation of insights to effectively inform the design activities.



2.2 NEEDS AND REQUIREMENTS:

2.2.1 What are requirements?

"Requirements" in software development refer to statements outlining what is necessary to design a system that meets user and customer objectives. However, this term can vary widely in interpretation among those developing interactive software. Some view it as encompassing all needed functionalities, while others see it as a compilation of user tasks. In UX, interaction design requirements focus on supporting user work activity needs and ensuring functional usefulness. Additionally, requirements aim to address emotional impact and long-term user experience aspects.

2.2.2 Requirement Specifications:

In **User interface design**, there's a move towards identifying important features and capabilities over rigid specifications, particularly focusing on user needs rather than software implementation details. The approach to interaction design requirements stems from analysed contextual data, resulting in a diverse set of descriptions that collectively form the interaction design requirements specification. Ultimately, this process yields various deliverables like personas, tasks, and usage scenarios that collectively inform the design process.

2.2.3 Software and Functional Implications of Interaction Design Requirements:

User needs extend beyond mere interaction needs, encompassing usability and overall user experience derived from functionality. Initially identified requirements often translate into broader system capabilities, reflecting both functionality and user interface support.

For instance, a Ticket Kiosk System requirement might specify the ability to purchase tickets for multiple events in one session. It's advisable to document corresponding functional needs derived from user requirements to facilitate collaboration with software engineering teams and ensure alignment throughout the project.

2.3 Formal Requirements Extraction

1. **Walking the WAAD for Needs and Requirements** : To conduct a "wall walk," a walkthrough of contextual data in the WAAD, with a new focus on extracting needs and requirements instead of refining the data. The objective is to navigate through the hierarchical WAAD structure and identify requirement statements directly from work activity notes. This approach aims to pinpoint essential user needs for system desig
2. **Switching from Inductive to Deductive Reasoning** : When extracting requirements from the WAAD, a deductive approach is used where each work activity note acts as a major premise. This process combines knowledge of UX and interaction design to deduce user needs and requirements, resulting in concise "requirement statements." For example, a note expressing concern about security in the Ticket Kiosk System leads to a high-level design requirement to protect transaction privacy, potentially implemented with a screen timeout feature. These requirements can blend interaction and functional aspects at a conceptual level.
3. **Preperation :** Create a requirements team comprising individuals skilled in deductive reasoning and creativity, including representatives from UX, software, system architecture, and possibly management. This collaborative approach improves communication between SE and UX roles, ensuring that various requirement types are channeled effectively. Designate a team leader and a recorder experienced in writing requirements. Use a structured template in a word processor, spreadsheet, or database to document requirement statements systematically. Conduct team meetings in the room with the WAAD displayed, utilizing a screen projector if needed for collective review of requirement statements while systematically traversing the WAAD hierarchy under the leader's guidance.
4. Systematic Deduction of Needs as “Hinges” to Get at Requirements : Initiate the process by allowing individuals to silently review the WAAD and jot down requirement ideas. The leader then guides the team through each node and note, prompting discussion on potential user needs reflected in the activity. These user needs, though not directly documented in the requirements, serve as pivotal points in transitioning from activity notes to requirements. With practice, this step becomes intuitive and facilitates the systematic deduction of user needs leading to concrete requirements.

5. Terminology Consistency : During this review of contextual data, focus on standardizing terminology to ensure consistency. User comments may use different terms for similar concepts (e.g., "alarm," "reminder," "alert," "notification" for calendar functions), reflecting variations in usage. It's important to identify and reconcile these differences to establish consistent terminology in the requirements document, aiding clarity and understanding across the team.
6. Requirement Statements: The team translates user needs from the WAAD into interaction design requirement statements, supporting these needs in the design. Each requirement reflects a specific user interface feature to address identified needs. Not every work activity note generates a need or requirement, and one note can produce multiple needs or requirements. The recorder documents requirement statements in the requirements document, including functional requirements that may imply system functionalities for software communication. The process involves using phrases like "**Users shall be able to**..." and may include rationale or notes to explain the relationship and considerations during extraction.
7. **Requirement statement structure** : A requirements document consists of structured requirement statements organized under headings. Each requirement statement includes a tag linking back to its source node in the WAAD, allowing traceability to original work activity data. This traceability ensures clarity and accountability throughout the UX lifecycle, akin to a software requirements traceability matrix. Before extraction, all WAAD nodes should be labeled with unique identification numbers (e.g., A, B, C for hierarchy levels and AA, AB, AC for sub-levels) to facilitate linking back to specific work activity notes in requirement statements.

Fig: Generic structure of a requirement statement

Name of major feature or category
Name of second-level feature or category
Requirement statement [WAAD source node ID]
Rationale (if useful): Rationale statement
Note (optional): Commentary about this requirement

Fig : Example requirement statement.

Security

Privacy of ticket–buyer transactions

Shall protect security and privacy of ticket-buyer transactions [C19]

Note: In design, consider timeout feature to clear screen between customers.

8. **Requirement Document Structure:**

Example: Extracting a Requirement Statement for the Ticket Kiosk System

Figure : Sample requirement statement for the Ticket Kiosk System.

..... **Transaction flow**

Recommendations for buying

Ticket-buyer purchases shall be supported by recommendations for the purchase of related items. [DE2].

Implied system requirement: During a transaction session the Ticket Kiosk System shall keep track of the kinds of choices made by the ticket buyer along with the choices of other ticket buyers who bought this item. [DE2].

Note: Amazon.com is a model for this feature.

Figure : Example requirement statement for the Ticket Kiosk System

Finding events

Direct keyword search by event description

Ticket buyers shall be able to find (e.g., search) by content to identify relevant current and future events [CA9].

Browse events by parameters

Ticket buyers shall be able to browse by category, description, location, time, rating, and price.

9. **Continue the Process for the Whole WAAD** : The process continues throughout the entire WAAD, extracting requirements for various aspects like event search and browsing functionalities. It's crucial to capture all connections between rationale and user activities to leverage the benefits of contextual analysis fully. Each requirement statement should be reviewed collectively to ensure alignment with the team's understanding. While not all extracted requirements may be feasible due to constraints and judgment, the focus at this stage is on systematically documenting requirements.

10. **Keep an Eye out for Emotional Impact Requirements and Other Ways to Enhance the Overall User Experience:** The focus is on recognizing the emotional impact and enhancing user experience during requirements gathering. It stresses moving beyond functional and usability considerations to achieve a superior user experience. By paying attention to user emotions like concerns, frustrations, and preferences, designers can create systems that better meet user needs. References to "fun" or dissatisfaction with routine tasks like data entry provide valuable insights for improving user engagement. Additionally, it advises recording all user feedback, including challenging needs, for later feasibility assessment.
11. **Extrapolation Requirements: Generalization of Contextual Data**: In the context of requirements gathering, extrapolation requirements involve broadening specific user statements from a Work Activity and Analysis Document (WAAD) to encompass more general needs. For instance, if ticket buyers express a desire to search for events based on criteria but omit browsing, an extrapolation requirement might include adding event browsing functionality. Similarly, user feedback about exchanging tickets can lead to requirements for posting, checking status, and exchanging tickets, necessitating user accounts tied to unique identifiers. These extrapolations should be validated with users to ensure they reflect genuine needs before finalizing them in the requirements document.
12. **Other Possible Outputs from the Requirements Extraction Process**: Work activity notes in a Work Activity and Analysis Document (WAAD) can lead to various outputs beyond requirement statements.

Missing Data Questions: Work activity notes in a WAAD can raise questions about missing contextual data. For example, during a contextual inquiry for MUTTS, the process of aggregating sales revealed gaps in understanding how ticket sales were managed across different systems, highlighting the need to explore reconciliation processes between local and national ticketing systems.

System Support Needs: WAAD insights may uncover system requirements beyond user experience and software domains, such as security and privacy. For instance, user concerns about identity theft and privacy led to requirements for specific system features addressing these issues, emphasizing the importance of collaboration with systems teams to devise appropriate solutions.

Marketing Opportunities: User comments captured in WAADs can provide valuable insights for marketing strategies. Comments about user preferences or concerns can be adapted into advertising copy, fostering collaboration between development and marketing teams and enhancing the project's market appeal.

13. **Constraints as Requirements :** Constraints such as legacy systems, implementation platforms, and system architecture are crucial considerations in real-world development projects. These constraints act as requirements that impact product scope, size, weight (especially for portable or mobile equipment), integration with existing systems, and compliance mandates.

Integration with Development Considerations: Interaction design must eventually align with constraints from systems engineering, hardware engineering, software engineering, management, and marketing. This alignment involves reconciling design decisions with development cost, schedule, and profitability targets, ensuring that interaction design is feasible within project constraints.

Impact of Constraints: Constraints arise from various factors including legacy systems, platform limitations, hardware and software demands, and budgetary and scheduling constraints. Understanding and managing these constraints is essential to delivering a successful product that meets both user needs and operational requirements.

Example :

Hardware Constraints for MUTTS:

The secure credit card server at MUTTS must remain operational continuously to avoid disrupting credit card transactions, which are vital for their business. With limited technical support available, maintaining this server alongside other office equipment poses a significant hardware constraint.

Physical Space and Workflow Efficiency:

MUTTS' leased office space is physically constrained, affecting work areas especially during busy periods. This constraint underscores the need for optimizing workflow efficiency within limited office space, emphasizing the importance of operational efficiency.

Anticipated Constraints for Ticket Kiosk System:

Transitioning to a ticket kiosk system introduces new hardware constraints, including ***the need for specialized kiosk hardware, a durable and vandal-proof outer shell, touchscreen interaction without a keyboard, and efficient network communications***. Maintenance priorities are also highlighted, such as ensuring uninterrupted ticket printing and providing backup communication features for customer support. These constraints reflect the specific requirements and challenges associated with implementing a ticket kiosk system.

14.Prioritizing Requirements :

Drawback of Affinity Diagrams:

Affinity diagrams lack prioritization, treating all notes equally regardless of significance. This results in unprioritized requirements, where major tasks hold the same weight as passing comments.

Remedying Prioritization:

To address this, during the validation process, engage customers and users to prioritize requirements. They can identify key requirements versus secondary ones, which can be organized into distinct sections or color-coded in the requirements document. Alternatively, each requirement can be tagged with an importance rating to guide design decisions.

Outcome of Prioritization:

Prioritization fosters mutual understanding between stakeholders, often leading to the realization that some requirements are unfeasible for the current product version. These insights enable strategic decision-making, setting aside certain requirements for future consideration.

15.Taking Requirements Back to Customers and Users for Validation:

Customer and User Validation of Requirements:

After internal review, it's crucial to validate the requirements document or WAAD with customers and users before proceeding to design. This step allows stakeholders to provide inputs, correct misconceptions, and strengthen the collaborative partnership.

Scheduling Validation Meetings:

Arrange meetings with representative users, including those previously interviewed and new users, to review the requirements thoroughly.

Engage participants from different work roles to ensure accurate interpretation of work activity notes.

Importance of New User Feedback:
New users can offer valuable perspectives by identifying overlooked aspects or providing fresh insights into requirements. Pay attention to their feedback, considering their expertise in the work domain but potential unfamiliarity with technical terms in interaction design or software development.

16. **Resolve Organizational, Sociological, and Personal Issues with the Customer:**

Addressing Organizational, Sociological, and Personal Issues: When validating requirements with the customer, seize the opportunity to resolve organizational, social, and personal concerns. Requirements can reveal potential issues in workflow changes, job roles, or work environment alterations, triggering concerns like territoriality, fear, or control among stakeholders.

Challenges in Workflow Changes: Changes driven by requirements may challenge established responsibilities and authorities, leading to resistance from affected individuals or teams. Legal requirements and platform constraints can also limit design options, emphasizing the need for early awareness and discussion.

Surprising Team Considerations: Team focus on technical and design aspects may overlook organizational and social implications highlighted by requirements. Addressing these issues early fosters collaboration and ensures alignment between design goals and organizational realities.

2.4 ABRIDGED METHODS FOR REQUIREMENTS EXTRACTION

1. **Using WAAD as Implicit Requirements:** The WAAD can serve as a direct representation of implicit requirements to save time and cost. Highlight work activity notes that imply requirements or design ideas directly on the WAAD. For example, a note expressing concern about security and privacy translates directly into a requirement to protect ticket-buyer transactions, sparking immediate design ideas documented directly on the WAAD.

2. **Anticipating Requirements in Contextual Analysis**: In contextual analysis, anticipate requirements by interpreting raw data on the fly to reflect needs more rapidly. For instance, restate interview notes into explicit needs and requirements to streamline the process.

3. **Using Work Activity Notes as Requirements**: Experienced practitioners can eliminate the WAAD entirely and use sorted work activity notes as requirements. This abridged approach involves deducing requirements directly from notes, bypassing formal documentation but requiring quick interpretation to identify relationships and derive requirements.

Keywords :

Usability: Usability is the pragmatic component of user experience, including effectiveness, efficiency, productivity, ease-of-use, learnability, retainability, and the pragmatic aspects of user satisfaction

Phenomenological Aspects of Interaction: Phenomenological aspects (deriving from phenomenology, the philosophical examination of the foundations of experience and action) of interaction are the cumulative effects of emotional impact considered over the long term, where usage of technology takes on a presence in our lifestyles and is used to make meaning in our lives.

Domain-Complex Systems: Domain-complex systems are systems with high degree of intricacy and technical content in the corresponding field of work. Often, characterized by convoluted and elaborate mechanisms for how parts of the system work and communicate, they usually have complicated workflow containing multiple dependencies and communication channels. Examples include an air traffic control system and a system for analyzing seismic data for oil exploration.

Legacy System : A legacy system is a system with maintenance problems that date back possibly many years

Work Role A work role is defined and distinguished by a corresponding job title or work assignment representing a set of work responsibilities. A work role usually involves system usage, but some work roles can be external to the organization being studied

Work Activity Affinity Diagram A work activity affinity diagram (WAAD) is an affinity diagram used to sort and organize work activity notes in contextual analysis, pulling together work activity notes with similarities and common themes to highlight common work patterns and shared strategies across all users.

End of Module1