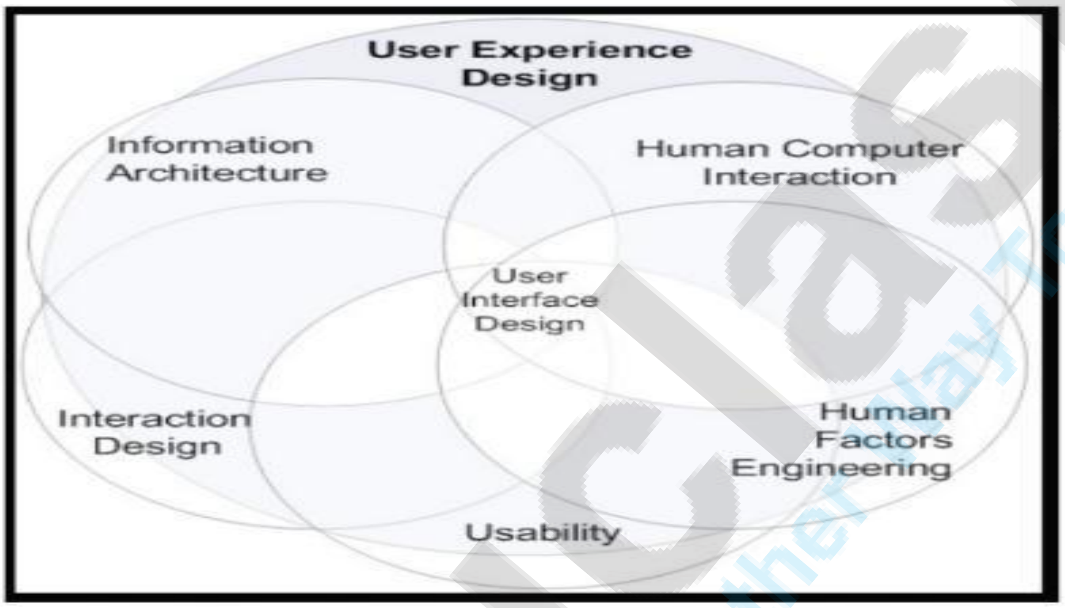
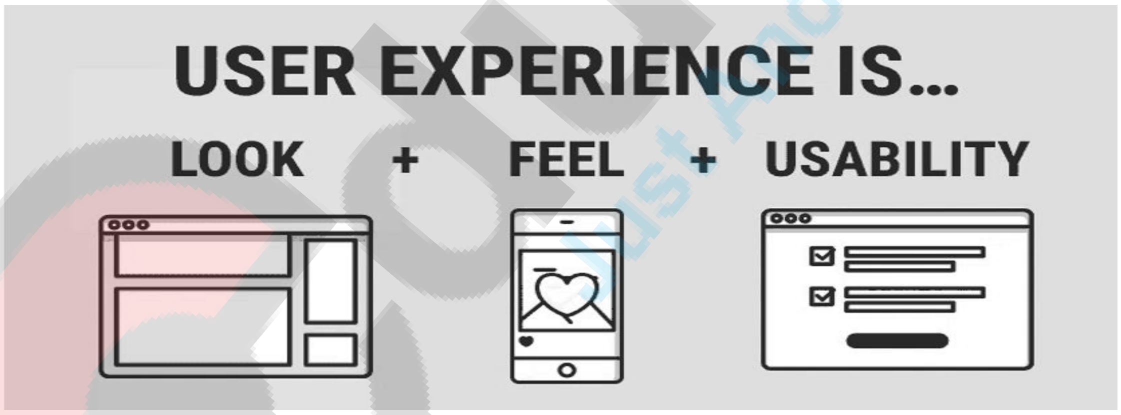
Q1.What is user experience and give example of ubiquitous experience?(unit1)

UXD is the process of creating products that provide meaningful and relevant experiences to users.
This involves the design of the entire process of acquiring and integrating the product, including aspects of branding, design, usability, and functionality.
The Aim is to design a product which we will be able to use not only with a success but with joy.
To Design a product that a user will wish to use it again.





1) Look :The look of a product is all about creating a product that has visual appeal ,In other words, it has to not only look nice, but look right too. In doing so, it establishes a bond of trust and credibility between the product and the user.

2) Feel:It is really about developing products that are a joy to use. That is, whether you're interacting with them or reacting to them, products should provide a pleasurable experience and not just a functional one.

3) Usability: It is the cornerstone of user experience. If a product isn't usable, the experience of using it can never be good. And Also it cost the reputation of the brand

Ubiquitous Interaction

The old-fashioneddesktop, laptop, and network based computing systems are alive and well and seem to be everywhere with an expanding presence in our lives.

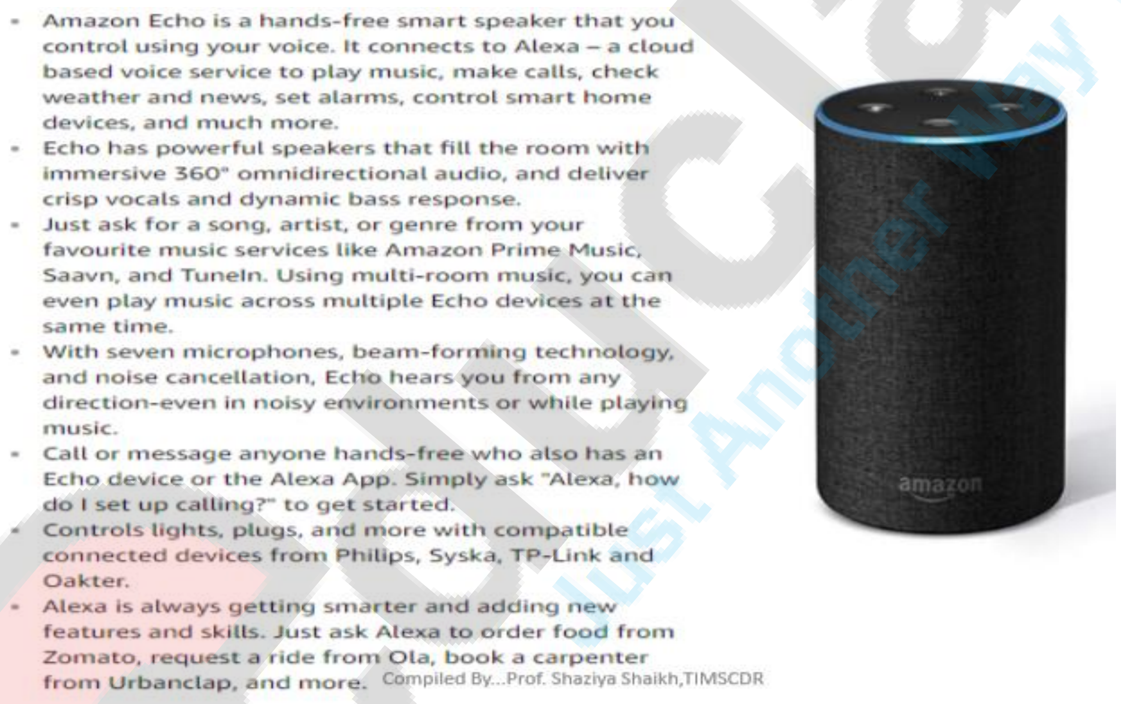
• Also complex domain systems are still the bread and butter of many business, industry, and government operations. • Web addresses are commonplace in advertisements on television and in magazines.

• The foreseeable future is still full of tasks associated with doing computing.

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.

Example

Amazon Echo



- Amazon Echo is a hands-free smart speaker that you control using your voice. It connects to Alexa – a cloud based voice service to play music, make calls, check weather and news, set alarms, control smart home devices, and much more.
- Echo has powerful speakers that fill the room with immersive 360° omnidirectional audio, and deliver crisp vocals and dynamic bass response.
- Just ask for a song, artist, or genre from your favourite music services like Amazon Prime Music, Saavn, and TuneIn. Using multi-room music, you can even play music across multiple Echo devices at the same time.
- With seven microphones, beam-forming technology, and noise cancellation, Echo hears you from any direction-even in noisy environments or while playing music.
- Call or message anyone hands-free who also has an Echo device or the Alexa App. Simply ask "Alexa, how do I set up calling?" to get started.
- Controls lights, plugs, and more with compatible connected devices from Philips, Syska, TP-Link and Oakter.
- Alexa is always getting smarter and adding new features and skills. Just ask Alexa to order food from Zomato, request a ride from Ola, book a carpenter from Urbanclap, and more.

Compiled By...Prof. Shaziya Shaikh,TIMSCDR

Q2.Draw and explain UX process life cycle template. (unit 2)

In Figure 2-1 we depict a basic abstract picture of activities for almost any kind of design, a cycle of the four elemental UX activities—Analyze, Design, Implement, and Evaluate—that we refer to generically as analysis, design, implementation, and evaluation.

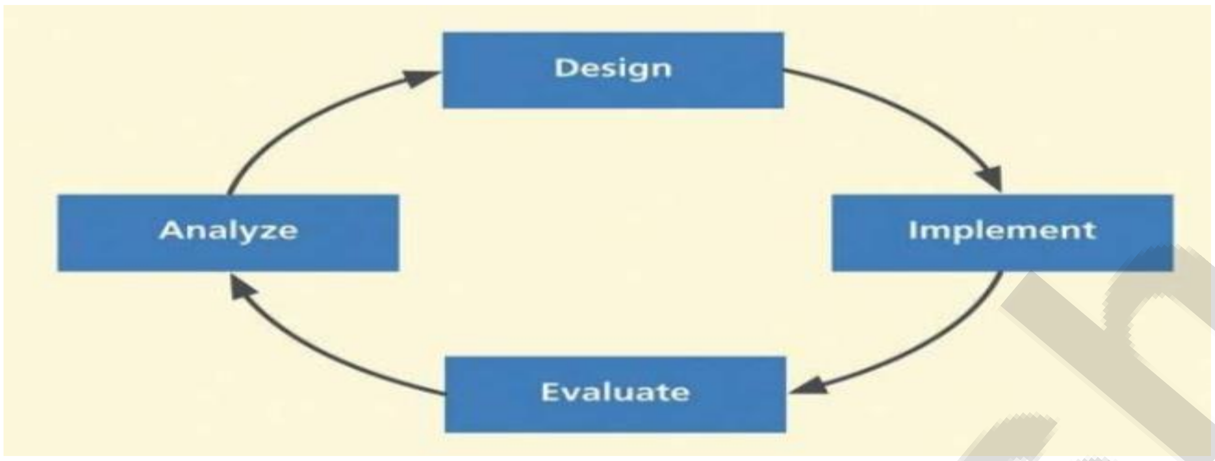


Fig 2.1

These four activities apply whether you are working with an architectural design, a hardware design, or a new car concept.

Figure 2-1 Universal abstract activity cycle of Analyze, Design, Implement, and Evaluate.

• In our lifecycle concept, specific to a UX process,

1) Analysis translates to understanding user work and needs.

2) Design translates to creating conceptual design and determining interaction behaviour and look and feel.

3) Implementation translates to prototyping,

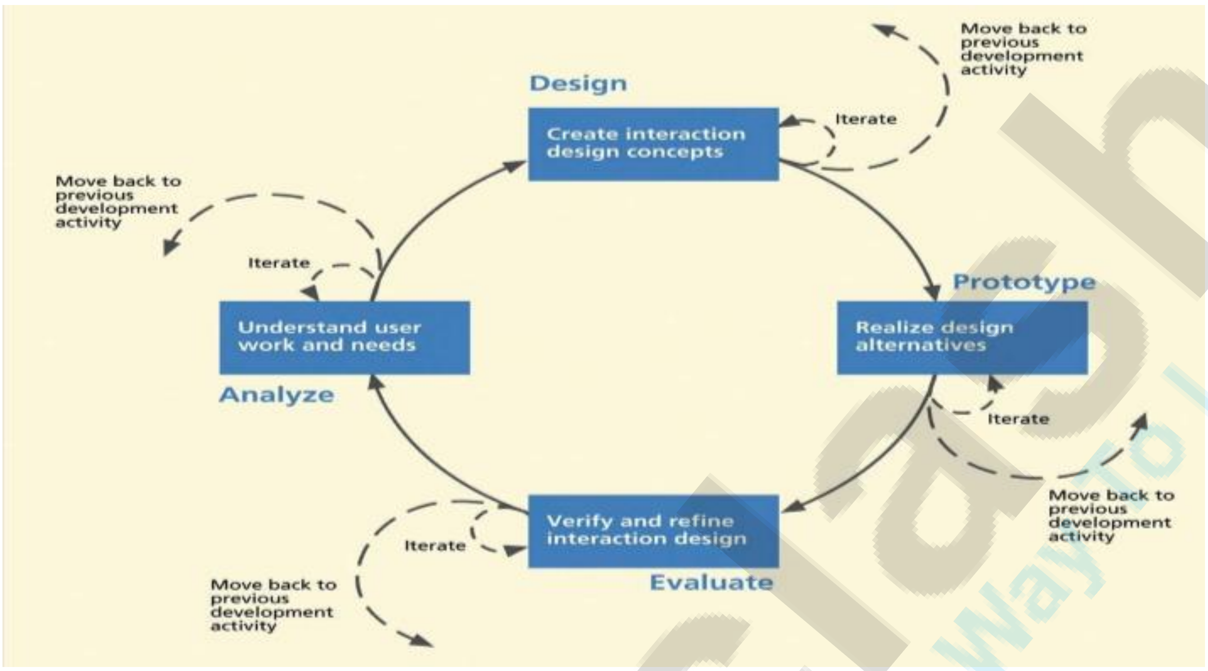
4) Evaluation translates to ways to see if our design is on track to meet user needs and requirements.

• In a larger system view, Implementation includes a final production of hardware and software, including the user interface.

• However, Implementation is limited to the interaction design component and prototyping is the design manifestation we use for evaluation before it is finalized for production.

In the context of interaction design and UX, this abstract cycle translates to our UX lifecycle template of Figure 2-2, which we call the Wheel. In our lifecycle concept, specific to a UX process, analysis translates to understanding user work and needs. Design translates to creating conceptual design and determining interaction behavior and look and feel. Implementation translates to prototyping, and evaluation translates to ways to see if our design is on track to meet user needs and requirements. In a larger system view, implementation includes a final production of hardware and software, including the user interface. However, in our UX lifecycle template, implementation is limited to the interaction design component and prototyping is the design manifestation we use for evaluation before it is finalized for production. The evaluation activity shown in Figure 2-2 includes both rigorous and rapid evaluation methods for refining interaction designs. Beyond that evaluation activity, the entire lifecycle is evaluation centered in the sense that the results of potentially every activity in the lifecycle are evaluated in some way, by testing, inspecting, analyzing, and taking it back to the customers and users. The entire lifecycle, especially the prototyping and evaluation activities, is supplemented and guided by UX goals, metrics, and targets.

Diagram:



Design
Create interaction design concepts
Iterate
Move back to previous development activity
Prototype
Realize design alternatives
Iterate
Move back to previous development activity
Evaluate
Verify and refine interaction design
Iterate
Move back to previous development activity
Analyze
Understand user work and needs
Iterate
Move back to previous development activity

Q3.What is horizontal and vertical UX process prototype? Give one example of each.(unit 4)

1) **Horizontal Prototype**: A horizontal prototype is very broad in the features it incorporates, but offers less depth in its coverage of functionality.
Horizontal prototypes are appropriate for understanding relationships across a broad system and for showing the range of abilities of a system.
A Horizontal, or User Interface, Prototype is a model of the outer shell of an entire system, i.e., windows, dialogue boxes, menus, screens, reports, and batch processes, with little or no processing behind them. It typically contains all of the system functions on menus, but includes only dummy screens, reports, and database queries (if applicable) for core functions. Initially, the Horizontal Prototype is unlikely to contain any processing logic behind the external features. In later stages, the Horizontal Prototype may be expanded to eventually evolve into the final system.
A Horizontal Prototype is usually developed during the early stages of analysis.
Purpose of Horizontal Prototype
The Horizontal Prototype is developed to:
· define project scope,
· demonstrate the external features of the system (e.g., windows, menu bars, and reports), as a means of communicating an understanding of the requirement.

2) **Vertical Prototype** :A vertical prototype contains as much depth of functionality as possible in the current stage of the project, but only for a narrow breadth of features. Vertical prototypes are most appropriate when a certain complex feature of a system is poorly-understood and needs to be explored, e.g. as a proof-of-concept.

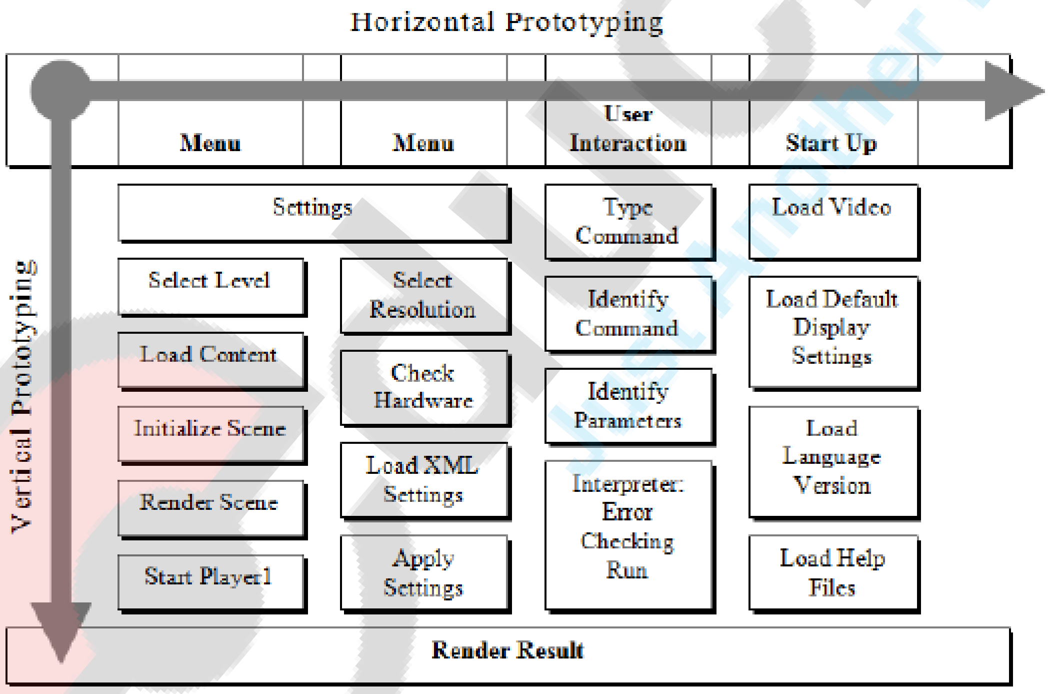
A Vertical Prototype contains a stripped down version of the core functions of the system. These functions provide the ability to enter data and store it in a database, as well as the ability to display data on query screens and reports.

A Vertical Prototype includes some user functionality but, more importantly, access to data. This can demonstrate to the user a "working" system, although not one that is fully functional or tuned.

Vertical Prototypes do not normally include:

- edits and controls,
- security features,
- audit trails,
- exception handling,
- record locking.

The Vertical Prototype is normally developed during the later stages of analysis.



Q4.How one should be prepared for contextual enquiry before the visit.

Getting prepared

Let’s review what you need for a contextual interview. To begin with the obvious, you should have some participants to visit. But how many is "some"?
Ask an experienced user researcher, "How many participants?" and you'll probably get more than one answer. There are a few reasons for this. First, any interviews are better than none, so even one participant will teach you something. Second, if the people you want to interview divide into clear types, start with 4-6 of each type. Third, if you don't know what different types there are, start with 8 people (the types will emerge from the patterns in their different experiences). Typically, I tend to start with around 20 users to design a typical system.
As well as your participants, you should have an outline discussion guide to act as a framework for eliciting stories. This discussion guide will contain the key assumptions you need to validate. Make sure your discussion guide is brief: don’t see it as a Q&A but as a kind of scaffolding to elicit and structure stories.
But even though you want the team to observe, there's a limit to how many people should come along on any one visit. A two-person research team is ideal: with three or more, the dynamics change. It's hard to stop observers interrupting the flow or changing the direction of the session. It can be a bit like people trampling over a crime scene and getting in the way.
To manage this, and to ensure that everyone gets their exposure hours, swap the note-taker for someone else on the design team after the first couple of participants. The note-taker could be a developer, designer, project owner, scrum master, or domain expert (the latter is especially helpful when there is domain-specific jargon with which you’re not familiar).
On the field visit, your role will be to develop rapport with the user; conduct the interview; and apprentice with the user. Your colleague will be the note-taker. His or her role is to take photographs of the user, the environment and any artefacts; audio record the interview; make written observations; and ask clarifying and follow-up questions.
A good field visit tends to have five stages:
Build rapport with the user.

- Transition from a traditional interview to a master-apprentice model.
- Observe.
- Interpret.
- Summarise.

Q5.What is the difference between model driven and data driven approach with reference to contextual enquiry. (unit 3)

Beyer and Holtzblatt (1998) takeanapproach tocontextual inquiry and analysis for HCI based on pure ethnographic field research. That is, their process is led entirely by work activity data. Simply stated, letting data do the driving means that if you encounter any information that seems relevant to the work practice and its milieu, collect it. This approach means forestalling any influence from yourownknowledge,experience,orexpectationsandjustgatheringdataasthey present themselves.

Data-driven contextual inquiry results in voluminous raw data describing a wide variety of topics. To digest this mass of disparate data points, make sense of them, and put these data to work in informing design, practitioners must apply contextual analysis to extract the concise and meaningful points and issues and then sort and organize them into piles or affinity diagrams. Then the sorted categories must be converted into designing forming models such as flow models, user models, and task models. In the purely data-driven approach, these categories and models are dictated by the data content. In effect, Beyer and Holtzblatt (1998) recommend not thinking of data categories in advance, but letting data suggest the categories and subsequent models. This will help avoid biasing the process by replacing data from users with analysts' hunches and opinions. Their "contextual design" approach to contextual inquiry and contextual analysis has proven itself effective. However, Constantine and Lockwood (1999) show that there is more than one effective approach to gathering contextual data to inform design. They promoteamethodtheycallmodeldriven,whichisinimportantwaysthereverse of the Beyer and Holtzblatt data-driven approach. In their "use what you know" approach, Constantine and Lockwood advocate using knowledge and expectations from experience, intelligent conjecture, knowledge of similar systems and situations, marketing analysis, mission statements, and preliminary requirements to focus your contextual inquiry data gathering to anticipate preconceived data categories and target the most useful data and to get a head start on its organization and analysis. From this experience, most practitioners know what kinds of models they will be making and what kinds of data feed each of these models. This knowledge helps in two ways: it guides data collection to help ensure that you get the kinds of contextual data you need, but at the risk of analyst bias in those data. It also helps with analysis by giving you a head start on data categories and models.

Certainly not all of this anticipatory information will be correct for a given work practice or system, but it can provide an advantageous starting point. Experienced professional practitioners, having gone through the contextual inquiry process and having done similar analyses in other work contexts, will learn to get through the chaff efficiently and directly to the wheat. Althoughtheirprocessmightseemthatitisabout modelling andthenfinding just the data to support the predefined models, it really is about starting with someinitial"exploratory"modelstoguidedatacollectionandthenfocuseddata collection to find answers to questions and outstanding issues, to refine, redirect, and complete the models. This "model-driven inquiry" approach also has a solid real-world track record of effectiveness. The Beyer and Holtzblatt contextual design approach works because, in the end, data will determine the truth about work practice in any specific real-world customer or user organization. However, the Constantine and Lockwood approach works because it encourages you to use your experience and what you know to anticipate data needs in contextual inquiry and contextual analysis. While data-driven inquiry assumes a "blank slate" and a completely open mind, model-driven inquiry acknowledges the reality that there is no such thing as a blank slate (Constantine & Lockwood, 1999). The Beyer and Holtzblatt approach is rooted in real data untainted by guessworkoranalystbiases. But the ConstantineandLockwoodapproachclaims advantages in lower cost and higher efficiency; they claim the search for data is easierifyouknowsomethingaboutwhatyouarelookingfor.Tothem,itisabout pragmatically reducing the ratio of data volume to insight yielded.

Q6.Distinguish between user interaction and user experience. (unit 3)

User Experience

User experience involves the whole spectrum of feelings and emotions of a user during his or her use of a product, which needs take into consideration a much broader scope than the traditional usability to obtain. Normally,user experience can be realized by several elements:

1. Usability which is necessary but not the only thing that matters obviously

2. Good point of interest and attractions of the service to appeal to a user

3. Concise instructions and specifications that help a user to understand and use the product easily and happily

4. Real value and experience that a user obtained throughout his or her use of the product

5. Satisfaction and good interaction the user experienced in his or her daily life and when communicates with others

In real life, however, the use of a product may not always be satisfied by users. Generally, it can be unassisted, unsatisfied, inaccessible or even failed sometimes. Imagine this, you are in a park where the roads in and out is really complicated which makes you lost your directions or leave you no choice but to go back the same old path to get to another site or just get out of there. If we take the park as a "product ", I believe your experience of using it must be a nightmare and you will never go to that park again.

So there comes the difference between user experience and interaction design. The user experience designer here must deal with the problem that though the roads are accessible to everywhere of the park and are also managed to let people go in and out the park, which were finished byinteraction designer, then why people still feel exhausted and drained and even reluctant to go to that park again.

Interaction Design

Interaction design associated more with the way a user utilizes or interacts with a product and also the method of how to design the process of interaction that between a user and a product. The interaction designer is the person that should concern how this should happen.

One of the most important factor of the design process is mastering an efficient prototyping tool like Mockplus or Axure. Beside, interaction designer must meet plenty of requirements to create and fulfill his or her interaction design, including customer research, massive practice, field study, feedback receiving, etc. On specialized conditions, the interaction designer is also the experience designer in spite of the roles are actually separated and emphasize distinguishing duties.

Here I would like to keep with the park metaphor, the interaction designer take the responsibility of designing the blueprint of this park. It is the interaction designer who builds the road that connected the whole park. He or she must be fully aware that which path could lead people to the gate, and which path could allow people to go back to the pleasure ground after having a good time in fishing pool. If the roads is a spider web, then the interaction designer is the spider who made it and made sure every junction is connected perfectly. Moreover, the interaction designer must qualified enough and keep all those interactions in check.

So, once a park like this is finished, people would appreciate the interaction designer for the work of making the park accessible to them, but the interaction designer may seek advice from user experience designer why people failed to enjoying a wonderful time in the park. And as such, the user experience and interaction design differ from the fact that the user design is often before interaction design happens for the user experience designer must deliver user insights, user demand, user journey maps and the like, or they just happen simultaneously.

Q7.Explain in brief emotional impact of user experience. (unit 1)

On Designing for the "User Experience"

• Utility The economic utility of a good or service is important to understand because it will directly influence the demand, and therefore price, of that good or service.

• Functional integrity focuses on the interrelationship of structure and function

• Usability the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.

• Graphic design combining text and pictures in advertisements, magazines, or books.

5.1 The Potential Breadth of Emotional Impact

Sometimes a user's reaction to a system or product is extremely emotional, a user experience with a deep and personal emotional impact. Atothertimes a user might be mildly satisfied (or dissatisfied) or just a bit pleased. Not all user experiences evoke throes of ecstasy, nor should they. Often just being well satisfiedwithoutit rising to apersonallyemotional level is all auser can afford in terms of emotional involvement with a software system. But, of course, we all live for the moments when the user experience hits the high end of emotional impact range when we experience amazingly cool products (software systems almost never reach these heights). We are talking about a product for which the user experience sets the product apart from the rest in the hearts and minds of discriminating users. Have you ever had something that you reallyloved to use? Something that had abeauty earned by its amazingly beautiful design? While other similar products may have an equally usable and useful design, theyjustdonothavethatsomethingextrathatsparksadeepemotionalchordof affinity. The others do not have that indefinable something that transcends form, function, usability, and usefulness, something that elevates the usage experience to pure joy and pleasure, something akin to the appreciation of well-crafted music or art.

• Attractive things make people feel good

5.2 A Convincing Anecdote

David Pogue makes a convincing case for the role of emotional impact in user experienceusingtheexampleoftheiPad.InhisNewYorkTimesstoryheexplains why the

iPad turned the personal devices industry upside down and started a whole new class of devices.

iPad is the most successful personal electronic device ever, selling 15 million in the first months.

When the iPad came out, the critics dubbed it “underwhelming,” “a disappointment,” and “a failure.” Why would anyone want or need it?

Pogue admits that the critics were right from a utilitarian or rational standpoint: “The iPad was superfluous. It filled no obvious need. If you alreadyhadatouch-screenphoneandalaptop,whyonearthwouldyouneedan iPad? It did seem like just a big iPod Touch” (Pogue). And yet, as he claims, the iPad is the most successful personal electronic device ever, selling 15 million in the first months. Why? It has little to do with rational, functional, and utility appeal and has everything to do with emotional allure. It is about the personal experience of holding it in your hand and manipulating finely crafted objects on the screen.

5.3 Aesthetics and Affect

• The movement from functionality and usability to aesthetics takes us from a utility to an experiential orientation, from a cognitive paradigm to an affective-centric paradigm.

• Interaction design can —touch humans in sensible and holistic way.

The term aesthetics is used to describe a sense of pleasure or beauty, including sensual perceptions (Wasserman, Rafaeli, & Kluger, 2000). Zhang presents a theoretical linkage between aesthetics and affect. Aesthetics,abranchofphilosophyandoftenassociatedwithart,isconsideredan elusive and confusing concept (Lindgaard et al., 2006). A key issue in studies regarding aesthetics is objectivity vs. subjectivity. The objective view is that aesthetic quality is innate in the object or the design and is known by certain features or characteristics regardless of how they are perceived. This means that objective aesthetic qualities can be evaluated analytically. The subjective view of aesthetics is that it depends on how they are perceived. Aesthetics has different effects on different people and must be evaluated with respect to users/people. It is all about perceived aesthetic quality. However, operationally, things are still a bit fuzzy. It is difficult to state goals for aesthetic design and there is no standard for measuring aesthetics: “...there is a lack of agreement and a lack of confidence on how to measure aesthetics relatedconcepts”(Zhang,2009).Itistypicaltothinkofone-dimensionalmetrics for aesthetics, such as subjective ratings of visual appeal. Lavie and Tractinsky (2004) draw a distinction between classical aesthetics— defined by orderliness in clean, pleasant, and symmetrical designs—and expressive aesthetics—defined by creativity, innovation, originality, sophistication, and fascinating use of special effects.

Norman (2004) proposes a three-level processing model for emotional design, making connection between aesthetics and emotion explicitly:

n Visceral processing requires visceral design—about appearance and attractiveness, appeals to “gut feeling” n Behavioral processing requires behavioral design—about pleasure and effectiveness (usability and performance) n Reflective processing requires reflective design—about self-image, identity, personal satisfaction, memories

Kim and Moon (1998) describe emotions, the immediate affective feelings about a system, in seven dimensions:n attractiveness n symmetry n sophistication n trustworthiness n awkwardness n elegance n simplicity

As Zhang notes, these dimensions are “non-basic” as compared to basic emotions such as joy and anger and can be domain specific. They also seem a bit arbitraryandcouldallowforquiteafewotheralternatives.Intheend,itisnotclear if, or how, these criteria can relate aesthetics in the design to affect in the users.

Q8.What is T prototype and local prototype in UX design process? (unit 4)

T Prototype : In a “T” prototype much of the design is realized at a shallow level (the horizontal top of the T), but a few parts are done in depth (the vertical part of the T).

A “T” prototype combines the advantages of both horizontal and vertical, offering a good compromise for system evaluation. Much of the interface is realized at a shallow level (the horizontal top of the T), but a few parts are done in depth (the vertical part of the T). This makes a T prototype essentially a horizontal prototype, but with the functionality details filled out vertically for some parts of the design.
In the early going, the T prototype provides a nice balance between the two extremes, giving you some advantages of each. Once you have established a system overview in your horizontal prototype, as a practical matter the T prototype is the next step toward achieving some depth. In time, the horizontal foundation supports evolving vertical growth across the whole prototype.

Local Prototype : A local prototype represents the small area where horizontal and vertical slices intersect, and is used to evaluate design alternatives for a particular isolated interaction detail.

We call the small area where horizontal and vertical slices intersect a “local prototype” because the depth and breadth are both limited to a very localized interaction design issue. A local prototype is used to evaluate design alternatives for particular isolated interaction details, such as the appearance of an icon, wording of a message, or behavior of an individual function. It is so narrow and shallow that it is about just one isolated design issue and it does not support any depth of task flow.
A local prototype is the solution for those times when your design team encounters an impasse in design discussions where, after a while, there is no agreement and people are starting to repeat themselves. Contextual data are not clear on the question and further arguing is a waste of time. It is time to put the specific design issue on a list for testing, letting the user or customer speak to it in a kind of “feature face-off” to help decide among the alternatives.
For example, your design team might not be able to agree on the details of a “Save” dialogue box and you want to compare two different approaches. So you can mockup the two dialogue box designs and ask for user opinions about how they behave.
Local prototypes are used independently from other prototypes and have very short life spans, useful only briefly when specific details of one or two particular design issues are being worked out. If a bit more depth or breadth becomes needed in the process, a local prototype can easily grow into a horizontal, vertical, or T prototype.

Q9.Describe UX process activities. (unit 3)

Analyze: Understanding the business domain, user work, and user needs Design: Creating conceptual design, interaction behavior, and look and feel Prototype: Realizing design alternatives

1) Horizontal Prototype A horizontal prototype is very broad in the features it incorporates, but offers less depth in its coverage of functionality.

2) Vertical Prototype A vertical prototype contains as much depth of functionality as possible in the current stage of the project, but only for a narrow breadth of features.

3) T Prototype : In a “T” prototype much of the design is realized at a shallow level (the horizontal top of the T), but a few parts are done in depth (the vertical part of the T).

4) Local Prototype : A local prototype represents the small area where horizontal and vertical slices intersect, and is used to evaluate design alternatives for a particular isolated interaction detail.

Evaluate: Verifying and refining the interaction design

• Flow not always orderly

• Managing the process with activity transition criteria

when to leave an activity

where to go after any given activity

when to revisit a previous process activity

when to stop making transitions and proceed to production

• Why do we even need iteration?

The UX process must be, and always will need to be, iterative. The design domain is so vast and complex that there are essentially infinite design choices along many dimensions, affected by large numbers of contextual variables.

• Iteration is not enough the answer is about balance of all four process activities of Figure. a—analyze, design, implement, and evaluate—for a given amount of resources.

• Start iteration early The earlier the interaction design iteration begins, the better. Typically, early cycles of iteration are devoted to establishing the basic underlying essentials of the design, including look and feel, and behaviour, before getting into design details and their refinement.

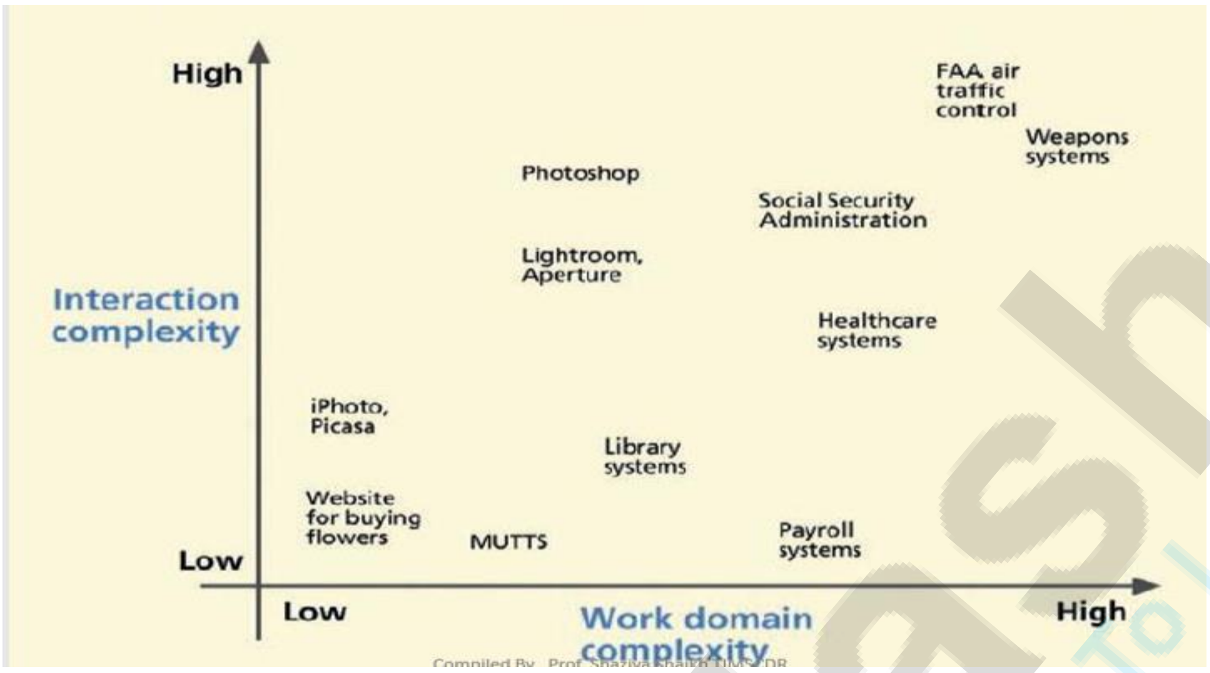
Thetransitioncriterioncomingoutofeach UXprocess activityboxisamultipathexitpointwiththreeoptions:moveforwardtothenext process activity, iterate some more within the current activity, or move back to a previous process activity. The decision of where to go next after a given process activity depends on the assessed quality of the product and/or work products of the current activity and a determination of what next activity is most appropriate. For example, after an initialprototypingactivity,ausabilityinspectionmightindicatethatthedesignisready for prototyping at a higher fidelity or that it is necessary to go back to design to fix discovered problems. Knowingwhenyouneedinter-activityiterationdependsonwhetheryouneedto pickupmoreinformationtodriveorinformthedesign.Whensomeofyourinputs are missing or not quite right, you must revisit the corresponding process activity. However, this kind of inter-activity iteration does not mean you have to redo the whole activity; you just need to do a little additional work to get what you need. Knowing when to stop iteration and proceed to production lies in a key process management mechanism.

Q10.What is system complexity space? Explain interaction complexity and work domain complexity with suitable example. (unit 2)

One of the things that makes it difficult to define a process for system design is that there is a spectrum of types of systems or products to be developed, distinguished mainly by complexity, each needing a somewhat different process and approach. In the next few sections we look at what is entailed in understanding this spectrum of system types.

Some systems are a combination of types and some are borderline cases. System or product types overlap and have fuzzy boundaries within the system complexity space. While there undoubtedly are other different ways to partition the space, this approach serves our purpose. In Figure 2-5 we show such a “system complexity space” defined by the dimensions of interaction complexity and domain complexity. Interaction complexity, represented on the vertical axis, is about the intricacy or elaborateness of user actions, including cognitive density, necessary to accomplish tasks with the system. Low interaction complexity usually corresponds to smaller tasks that are generallyeasytodoonthesystem,suchasorderingflowersfromaWebsite.High interaction complexity is usually associated with larger and more difficult tasks, oftenrequiringspecialskillsortraining,suchasmanipulatingacolourimagewith Adobe Photoshop. On the horizontal axis in Figure 2-5 we show work domain complexity, which is about the degree of intricacy and the technical nature of the corresponding field of work. Convoluted and elaborate mechanisms for how parts of the system work and communicate within the ecology of the system contribute to domain complexity.

Diagram



The work in domain-complex systems is often mediated and collaborative, with numerous “hand-offs” in a complicated workflow containing multiple dependencies and communication channels, along with compliance rules, regulations, and exceptions in the way work cases are handled. Examples of complex work domains include management of arcane financial instruments such as credit default swaps, geological fault analysis for earthquake prediction, and healthcare systems. Low work domain complexity means that the way the system works within its ecology is relatively simple. Examples of work domains with low complexity include that same Website for buying flowers and a simple calendar management application.

Q11.List the different members in UX team and explain their roles. (unit 2)

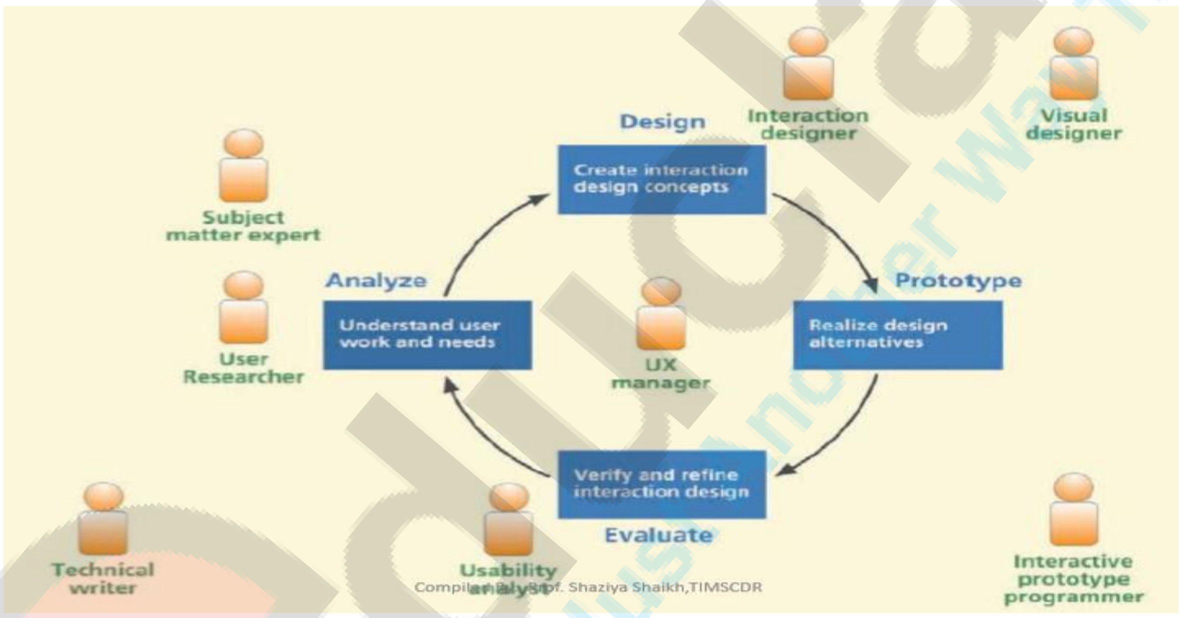
One early stage activity in all interactive software projects is building the UX team. Someone, usually the project manager, must identify the necessary roles and match them up with available individuals. Especially in small projects, the different roles are not necessarily filled with different people; you just need to maintain the distinction and remember which role is involved in which context and discussion. In addition to the software engineering roles, here we are mainly concerned with roles on the UX team. Roles we can envision include the following:

1) User researcher: involved with contextual inquiry and other work domain analysis activities. You may also need other roles even more specialized, such as a social anthropologist to perform in-depth ethnographic field studies.
2) Users, user representatives, customers, and subject matter experts: used as information sources in contextual inquiry and throughout the lifecycle.
3) User interaction designer: involved with ideation and sketching, conceptual and detailed design, and low-fidelity prototyping activities.
4) UX analyst or evaluator: involved in planning and performing UX evaluations, analyzing UX problems, and suggesting redesign solutions.
5) Visual/graphic designer: involved in designing look and feel and branding and helping interaction designers with visual aspects of designs.

6) Technical writer: involved in documentation, help system design, and language aspects of interaction designs.
7) Interactive prototype programmer: involved in programming interactive highfidelity UX design prototypes.
8) UX manager: someone with overall responsibility for the UX process.

Often terms for team roles are used loosely and with overlap. For example, “UX engineer” or “UX practitioner” are catch-all terms for someone who does contextual analysis, design, and evaluation on the UX side. As a further consideration, in many projects, team composition is not static overthe whole project.For example, people may come and gowhentheir special talentsarerequired,anditisnot unusualforthe teamtogetsmallerneartheend ofthelifecycle.Oftenneartheendoftheversionorreleasecycle,muchof project teamgetsreassignedanddisappearsandyougetapossiblynewandmuchsmaller one, with a much shorter remaining lifecycle.

Diagram


Design
Interaction designer
Visual designer
Create interaction design concepts
Subject matter expert
Analyze
Prototype
Understand user work and needs
Realize design alternatives
User Researcher
UX manager
Verify and refine interaction design
Evaluate
Technical writer
Usability analyst
Interactive prototype programmer
Compiled By Prof. Shaziya Shaikh,TIMSCDR

Q12.State guidelines to be used for synthesizing work activity notes. (unit 3)

Because some application domains can be unfamiliar to some team members, the work activity note synthesis should be done by people who have already been immersed in the contextual data, probably the same people who did the interviews and observations.

• Guidelines for synthesizing work activity notes, starting here: As you create each new work activity note, tag it with a source ID, a unique identifier of the person being observed and/or interviewed when the note was written.

Paraphrase and synthesize instead of quoting raw data text verbatim. For example: Raw data: “I think of sports events as social events, so I like to go with my friends. The problem is that we often have to sit in different places, so it is not as much fun. It would be better if we could sit together.”

In the user’s perspective: “When I am looking to buy student tickets to MU basketball, I look for an option allowing several friends to sit together.”
 Make each work activity note a simple declarative point instead of quoting an interviewer’s question plus the user’s answer. Filter out all noise and fluff; make each note compact and concise, easily read and understood at a glance. Be brief: Keep a note to one to three succinct sentences. Each note should contain just one concept, idea, or fact, with possibly one rationale statement for it. Break a long work activity note into shorter work activity notes. Make each note complete and self-standing.

Never use an indefinite pronoun, such as “this,” “it,” “they,” or “them” unless its referent has already been identified in the same note.

- State the work role that a person represents rather than using “he” or “she.”
- Add words to disambiguate and explain references to pronouns or other context dependencies.
- Avoid repetition of the same information in multiple places.

Example: Work Activity Note Synthesis for MUTTS

- User comment (in response to thinking ahead about including athletic events): When I am looking to buy student tickets to MU basketball, I like to look at different seating options vs. prices; I sometimes look for an option allowing several friends to sit together.
- Synthesized work activity notes: When I am looking to buy student tickets to MU basketball, I like to look at different seating options vs. prices. When I am looking to buy student tickets to MU basketball, I sometimes look for an option allowing several friends to sit together.

Q13.Have you ever had something that you really love to use?

DIY.

Q14.Explain the nature of gap between analysis and design using diagram.(unit 3)

Contextual inquiry and analysis are about understanding existing work practiceand context. Thenwe move on to producing designs for a new systemto supportpossibly new ways that work gets done. But what happens in between? Theoutput of contextual inquiry and analysis does not speak directly to what isneeded as inputs to design. There is a gap.

- Information coming from contextual studies describes the work domain but does not directly meet the information needs in design.
- There is a cognitive shift between analysis-oriented thinking on one side of the gap and design-oriented thinking on the other.

Figure 5-1

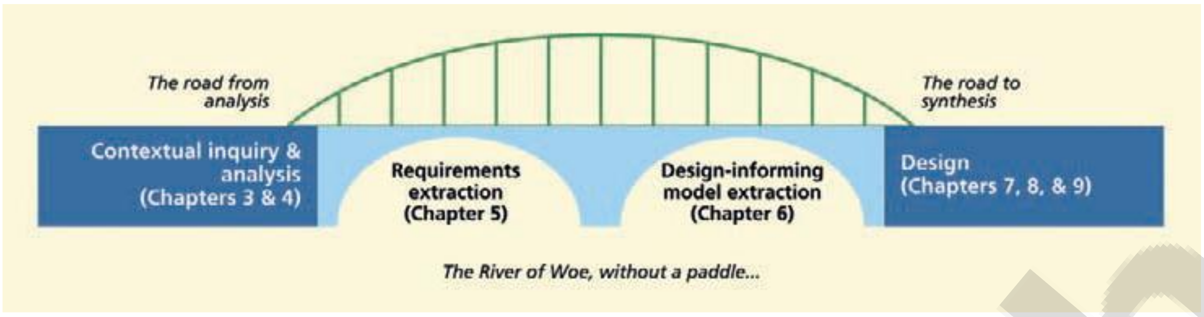

The road from analysis
Contextual inquiry & analysis (Chapters 3 & 4)
Requirements extraction (Chapter 5)
Design-informing model extraction (Chapter 6)
The road to synthesis
Design (Chapters 7, 8, & 9)
The River of Woe, without a paddle...

Figure 5-2 Overview of the bridge todesign.

- The gap is the demarcation between the old and the new—between studying existing work practice and existing systems and envisioning a new work space and new system design space.

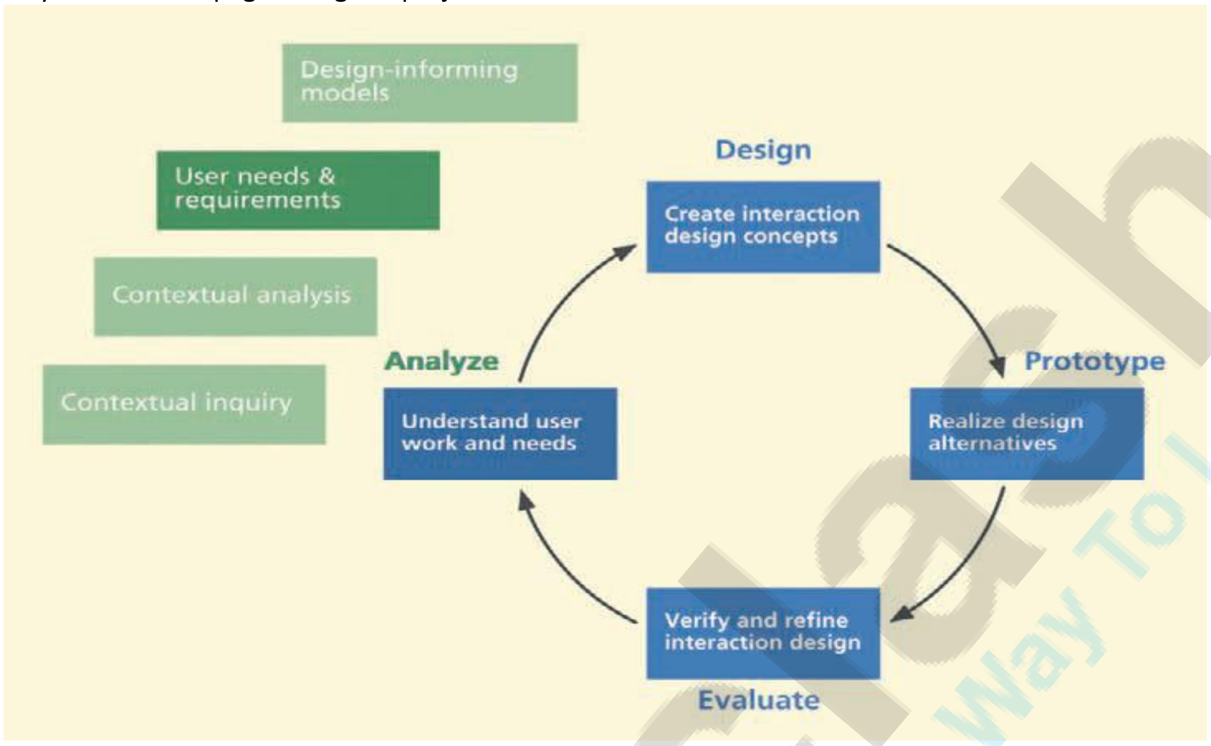
Q15.What are the requirements for interaction design and explain its importance.(unit 4)

What Are "Requirements"?
The term refers to a statement of what is needed to design a system that will fulfill user and customer goals. But when you start getting specific, it is a term that can mean something different to just about everyone associated with developing interactive software systems. To one, it is about ascertaining all the functionality needed to do the job. To another it is a compilation of all the user tasks needed to do the job. In the UX domain, interaction design requirements describe what is required to support user or customer work activity needs. To that end we are also concerned with functional requirements to ensure the usefulness component of the user experience. Finally, we will have requirements to fulfill the need for emotional impact and long-term phenomenological aspects of the user experience.
Requirements "Specifications"
Before we get into extracting requirements from contextual data, let us look briefly at the forms interaction design requirements can take. One term we often think of when coupled with "requirements" is "specifications."
In past software engineering traditions, a formal written requirements document was de rigueur and could even designate details about how the corresponding software is to be implemented, including such software stuff as object models, pseudo-code, use cases, and software structure. However, currently in software engineering and software requirements engineering there is an increasing recognition that:

- Detailed formal requirements cannot ever be complete.
- Detailed formal requirements cannot ever be 100% correct.
- Detailed formal requirements cannot be prevented from changing throughout the lifecycle.

As a result, there appears to be a trend toward abandoning the detailed requirements specifications in favour of ascertaining the important features and capabilities.
Software and Functional Implications of Interaction: Design Requirements
User needs are really not just interaction needs. Usability and UX includeusefulness that we get from functionality. Often an initial requirementextracted from contextual data first appears as a requirement for a broad overall system capability—that is, it expresses a need for both functionalityand user interface support.As an example, a Ticket Kiosk System requirement might state that a usershould be able to buy tickets for up to 10 different events in one session ortransaction. We recommend that you devise a way to record the functional needsthat correspond

to user needs and requirements revealed in this process andpass them on to your software engineering counterparts. It will help them beaware of needed functionality and will help you both stay on the same pageduring the project.


Design-informing models
User needs & requirements
Contextual analysis
Contextual inquiry
Design
Create interaction design concepts
Analyze
Understand user work and needs
Prototype
Realize design alternatives
Verify and refine interaction design
Evaluate

Q16.Explain a process of extracting requirements systematically from conceptual data.

or

Q17.Explain the standard guidelines for extracting the requirements.(unit 3)

This process of extracting needs and requirements is similar to datainterpretation and consolidation sessions of contextual analysis in that itinvolves a group sitting down together and going over a large amount of data,including the WAAD and evolving design-informing models. But here it isactually easier because much of the hard work is already done.

Walking the WAAD for Needs and Requirements

It is now time for your team to get re-immersed inwork activity data; this time with the focus of the walkthrough on extractingneeds and requirements rather than iteratively improving the data. The generalidea is to traverse the hierarchical WAAD structure and focus on extractingrequirement statements from work activity notes.

Switching from Inductive to Deductive Reasoning

Extracting requirements from the WAAD calls for a deductive thinkingprocess. It is deductive because each work activity note in the WAAD istreated as the major premise in a logical syllogism. The second "premise"is everything you know about UX and interaction design. The conclusionof this syllogism is a statement of user needs and requirements youdeduce from the work activity note, something we capture in a"requirement statement."

To clarify with a small example from MUTTS and the Ticket Kiosk System, aWAAD note, say in node C19, that says "I am concerned about security and privacy of my transactions" can imply a design requirement (at a high level): "Shall protect security and privacy of ticket-buyer transactions." In the design, this requirement might be at least partially met by a timeout feature to clear the screen between customers. Note that at this level, requirements can be a mix of interaction and functional requirements

Preparation

Select a requirements team, including people you think will be best at deductive reasoning and creativity. You will need both UX and software people represented, plus possibly system architects and maybe managers.

This team approach enhances SE-UX communication because the SE and UX roles are working together at a crucial point in their mutual lifecycles, describing and funneling the different kinds of requirements to the places they will be used. Choose a requirements team leader and a recorder, a person experienced in writing requirements. You may need a requirements "record" template in a word processing document, a spreadsheet, or a database schema to capture the requirement statements in a consistent and structured format in an interaction design requirements document (or requirements document, for short in this context). The requirements team will work in the room where the WAAD is posted on the wall.

Systematic Deduction of Needs as "Hinges" to Get at Requirements

Start by letting everyone walk through the WAAD, individually and silently, to accommodate those who need to think quietly and to allow everyone to write notes about ideas for requirements. Then begin the main part of the process.

As the leader walks the team through the WAAD, one node and one note at a time, the team works together to ask what user needs, if any, are reflected in this work activity note and the hierarchical labels above it.

Such user needs are still expressed in the perspective of the user and in the work domain. Although the user need is not documented in the requirements document, it is an important "hinge" in the mental process of getting from work activity notes to requirements. This interim step will become almost automatic with only a little practice.

Terminology Consistency

This pass through the contextual data is a chance to standardize terminology and build consistency. Your contextual data will be full of user comments about an infinitude of usage and design concepts and issues. It is natural that they will not all use exactly the same terms for the same concepts.

For example, users of a calendar system might use the terms "alarm," "reminder," "alert," and "notification" for essentially the same idea. Sometimes differences in terminology may reflect subtle differences in usage, too. So it is your responsibility to sort out these differences and act to help standardize the terminology for consistency issues in the requirements document.

Requirement Statements

Next, the team translates each user need into one or more interaction design requirement statements. Each requirement statement describes a way that you decide to support the user need by providing for it in the interaction design. Ask what new or more specific user interface feature you should see in the design to support the user needs implied by this WAAD note. There is not necessarily a one-to-one correspondence between work activity notes in the WAAD and needs or requirements.

A given work activity note might not generate a need or requirement. The ideas in some notes may no longer be relevant in the envisioned system design. Sometimes one work activity note can produce more than one need. A single need can also lead to more than one requirement.

Interaction requirements often imply functional requirements for the system, which you may also capture here for communicating to your software people.

For example:

- Interaction requirement: "Ticket buyers shall be able to see a real-time preview of available seating for a venue."

- Corresponding system requirement: “System shall have networked infrastructure to poll all kiosk transactions as they are happening and coordinate with the venue seating data to ‘lock and release’ selected seats.”

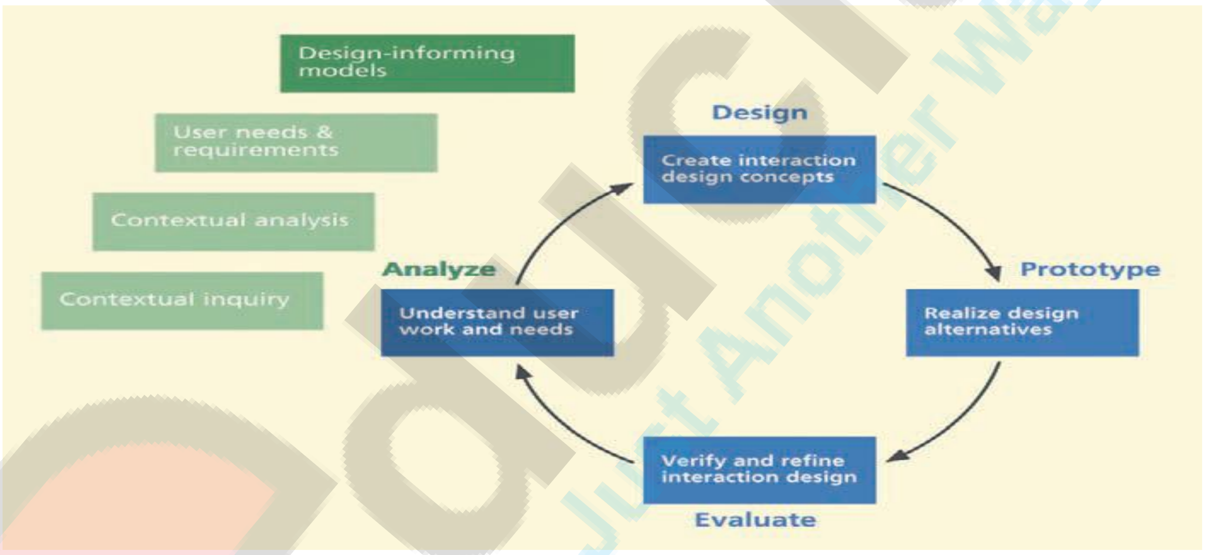
Requirement Statement Structure

A requirements document is essentially a set of requirement statements organized on headings at two or more levels. For systems where risk is high and traceability is important, each requirement is tagged with the WAAD source node ID, which serves as a link back to the source of this requirement statement within the WAAD. The WAAD in turn has a link back to its source in raw work activity data. Later, if a question arises about a particular need or requirement, the connection to original work activity data and the person who was its source can be traced to find the answers (sort of the UX lifecycle analog of a software requirements traceability matrix).

Requirements Document Structure

We show two levels of headings, but you should use as many levels as necessary for your requirements.
As an example of an extracted requirement for the Ticket Kiosk System, suppose in our contextual inquiry a user mentioned the occasional convenience of shopping recommendations from Amazon.com.

Q18.Explain the design informing models in detail.(unit 3)



Design-informing models
User needs & requirements
Contextual analysis
Contextual inquiry
Analyze
Understand user work and needs
Design
Create interaction design concepts
Prototype
Realize design alternatives
Verify and refine interaction design
Evaluate

Design-informing models are not building blocks that appear directly in a design but are artifacts that embody, drive, inform, and inspire the design. They are design-oriented constructs, such as task descriptions or user personas, that turn raw data into actionable items as design ideas, as elements to consider or take into account in the design.
Like WAADs and requirements, design-informing models:

- help integrate and summarize the contextual data
- point back to the data, to maintain the “chain of custody” to ensure that the designis based on real contextual data
- provide a shared focus for analysis now and, later, design
- provide intermediate deliverables, which can be important to your workingrelationship with the customer.

Envisioned Design-Informing Models

Use these models as springboards to your design scenarios, sketches, andstoryboarding. Using the flow model and physical model as guides, look forways to make flows more efficient and to avoid

redundant data entry andunnecessary physical motions. From the task interaction models, try toreduce and automate steps.
Using the social model as a guide, find ways to increase communication,reinforce positive values, address concerns of people in work roles, andaccommodate influences. One important way to use each kind of modelto inform design is to look at all the barriers identified in the models and solvethe problems they represent.
When the new work practice and supporting system are quite different fromthe existing ones, the transition from modelling to design begins with atransition from the models of existing work practice by envisioning howeach model will make the transition to the new work practice and supportingdesign.
When the new work practice and supporting system are quite different from the existing ones, the transition from modelling to design begins with a transition from the models of existing work practice by envisioning how each model will make the transition to the new work practice and supportingdesign.
Each model directly informs its envisioned counterpart. Envisioned design-informing models are a step closer toward design from analysis. Most of the envisioned design-informing models can bevery brief, addressing only the differences from the existing models.

Q19.Differentiate between design informing models and user models.

Q20.Write a note on user persona.(unit 4)

A well-defined user persona contains four key pieces of information:
• Header
• Demographic Profile
• End Goal(s)
• Scenario
Before you create a persona, conduct research to make sure your personas accurately representyour users. After you gatheran adequate amount of data, organize the information into personagroups that represent your ideal customers. Remember tofocus on the major needs of the most important user groups—you can't be everything to everyone, nor should you try to be!
Add a Header
The header includes a fictional name, image, and quote that summarize what matters most to the persona as it relates to your
product. These features aid in improving memorability, keeping your design team focused on the users they are building the
product for.

Add a Demographic Profile
While the name and image can be fictional, demographic details are factual and based on user research. The demographic
profile includes four main sections: personal background, professional background, user environment, and psychographics.
• Personal Background
The personal background includes details such as age, gender, ethnicity, education, persona group (e.g. working
moms), and family status (e.g. single, married with children, widowed, etc.).
• Professional Background
The professional background includes details such as job occupation, income level, and work experience.
• User Environment

The user environment represents the physical, social, and technological context of the user. This section is used to
answer questions like: What technological devices do users have access to? Do they spend most of their time in a
corporate office or a home office? And how often do they collaborate with others?
• Psychographics
Psychographics include details such as attitudes, interests, motivations, and pain points. Overall, the demographic
profile adds an additional layer of realism to a user persona, boosting empathy when exploring user needs and goals.

Add End Goal(s)
The end goal is the motivating factor that inspires action, and answers the question: what do users want or need to accomplish
by using your product? End goals are the main driving forces of your users and determine what the persona wants or needs to
fulfill.

Add a Scenario
A scenario is a "day-in-the-life" narrative that describes how a persona would interact with your product in a particularcontext to achieve his or her end goal(s). The scenario usually defines when, where, and how the narrative takes place. Theyare typically written from the perspective of the persona and describe use cases that may happen in the future.
Each user persona is typically one page, which requires you to focus on the essential elements. As a rule of thumb, avoidadding extra details that cannot be used to influence the design. If it does not affect the final design or help make any decisionseasier: omit it.Personas are also key to findingideal customers in real life so you can test and validate your productwith real people.Therefore, make sure each persona is specific and realistic: avoid exaggerated caricatures, andinclude enough detail to helpyou find real-life representation.

Q21.Write a note on Work Role. (unit 3)

Managing Complexity with Work Roles and Flow Models
We stand at the beginning of a process in which we will invest a lot of effortto understand the user's work domain for which the system is being designedand how the users of that system are best served in the design. When we arestarting cold and just beginning this undertaking, the task seems enormous anddaunting. We need two things to help control the complexity and wrap ourheads around the problem:

- a big picture of the work domain, its components, and how information flowsamong them
- a way to divide the big picture into manageable pieces

Because these two things are somewhat in opposition and cannot be done byone single means, we need two complementary concepts to solve the two parts ofthe problem, respectively:

- a flow model to provide the big picture
- the concept of work roles as a basis to divide and conquer

We cannot overemphasize the importance of work roles and the flow modelin almost everything else you do in contextual inquiry and analysis andmodelling. These two notions influence almost all the UX activities that follow inthis book, including contextual inquiry, contextual analysis, requirements, design, user experience goals, and UX evaluation.

Identify Work Roles as Early as Possible

The very first thing to start doing as you talk with customers and users is to identify work roles. A work role is defined and distinguished by a corresponding job title or work assignment representing an area of workresponsibility.

As Beyer and Holtzblatt (1998, p. 163) put it, a work role is a "collection of responsibilities that accomplish a coherent part of the work." The work activities of the enterprise are carried out by individual people who act in the work roles, simplicity we will refer to a work role as a person instead of spelling out that it is a person in that work role. A given work role may or may not involve system usage and some roles can be external to the organization, for example, a parts vendor, as long as they participate in the work practice of the organization.

Example: Initial Work Role Identification in MUTTS

The two obvious work roles in MUTTS are the ticket seller and ticket buyer. Among the other roles we discovered early in contextual inquiry are the event manager, the advertising manager, and the financial administrator. The event manager interacts with external event sponsors and venue managers to book events for which they sell tickets. The financial manager is responsible for accounting and credit card issues. The advertising manager interacts with outside sponsors to arrange for advertising, for example, ads printed on the back of tickets, posted on bulletin boards, and on the Website. In addition, we discovered a few more work roles that we will introduce in later sections.

Q22.Explain flow model with the help of an example.(unit 3)

A flow model is your picture of the work domain, its components andinterconnections among them, and how things get done in that domain.A flow model captures workflow relationship among key work roles. A flowmodel tells who does what and how different entities communicate to getwork done.Even though your early contextual inquiry data will be incomplete and notentirely accurate, we recommend you start as early as possible acquiring anunderstanding of the work roles and a sketch of the flow model, refining as thepicture of the work domain, the system, and its users slowly becomes clearer. Youwill be constantly updating this overview as you learn more via the interviewsand observations. Because the flow model is a unifying representation of how thesystem fits into the workflow of the enterprise, it is important to understand itand get it established as early as possible.Even the sketchiest flow model will help guide the remaining contextualinquiry. You will want to use the flow model as a reference to keep everything elsein perspective as you do the research of your contextual analysis. Within the workdomain and within the system, work is done by the work roles described inthe previous section, which play a central part in the flow model. Therefore, you should begin your flow model sketch by drawing icons, labelled with the work roles, which will be nodes in a connected graph isthe flow model. Include any roles external to the organization but involved inany way in the work practice. Add additional labelled nodes for any other entity,such as a database, into which and from which anything related to the work practice can flow.We will soon refine flow models in more detail.

Example: Sketching the Flow Model for MUTTS

As we conducted contextual inquiry sessions for the MUTTS ticket-buyingactivity, we sketched out an initial flow model on flip charts, which we recreatedhere in Figure 4-3.

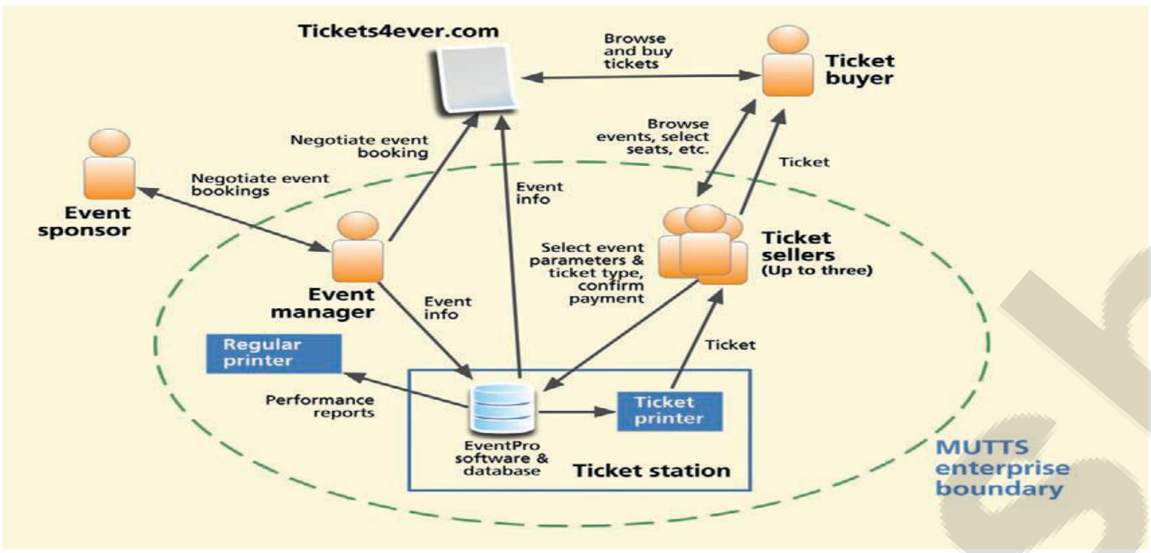


Figure 4-3: An initial flow model sketch of the MUTTS system.

Q23.Explain in detail Ecological, Interaction and Emotional Design perspective.

Ecological Perspective

The ecological design perspective is about how the system or product workswithin its external environment. It is about how the system or product is used inits context and how the system or product interacts or communicates with itsenvironment in the process. This is a work role and workflow view, whichincludes social interaction and long-term phenomenological aspects of usage aspart of one’s lifestyle.

System infrastructure (Norman, 2009a) plays an important role in theecological perspective because the infrastructure of a system, the other systemsand devices with which it interacts in the world, is a major part of its ecology.Infrastructure leads you to think of user activities, not just isolated usage. Norman (2009b) states it in a way that designers should take to heart, “A productis actually a service.”

Interaction Perspective

The interaction design perspective is about how users operate the system or product. It is a task and intention view, where user and system come together.

It is where users look at displays and manipulate controls, doing sensory, cognitive, and physical actions.

Emotional Perspective

The emotional design perspective is about emotional impact and value-sensitive aspects of design. It is about social and cultural implications, as well as the aesthetics and joy of use. System infrastructure (Norman, 2009b) can also play a role in the emotional perspective because the infrastructure of a system provides scaffolding for the phenomenological aspects of usage, which are about broader usage contexts over longer periods of time.

A product is not just a product; it is an experience (Buxton, 2007a). People do not usually use a product in isolation from other activities. People use products as part of an activity, which can include many different kinds of usage of many different things. And that starts with the out-of-the-box experience, which is not enhanced by difficult hard plastic encasing, large user manuals, complex installation procedures, and having to consent to a legal agreement that you cannot possibly read.

Q24.Differentiate between sketching and prototyping.

Q25.Explain in detail wireframing?(unit 4)

Wireframes, a major bread-and-butter tool of interaction designers, are a form of prototype, popular in industry practice. Wireframes comprise lines and outlines (hence the name "wire frame") of boxes and other shapes to represent emerging interaction designs. They are schematic diagrams and "sketches" that define a Web page or screen content and navigational flow. They are used to illustrate high-level concepts, approximate visual layout, behaviour, and sometimes even look and feel for an interaction design. Wireframes are embodiments of maps of screen or other state transitions during usage, depicting envisioned task flows in terms of user actions on user interface objects.
The drawing aspects of wireframes are often simple, offering mainly the use of rectangular objects that can be labelled, moved, and resized. Text and graphics representing content and data in the design is placed in those objects. Drawing templates, or stencils, are used to provide quick means to represent the more common kinds of user interface objects (more on this in the following sections).
Wireframes are often deliberately unfinished looking; during early stages of design they may not even be to scale. They usually do not contain much visual content, such as finished graphics, colours, or font choices. The idea is to create design representations quickly and inexpensively by just drawing boxes, lines, and other shapes.

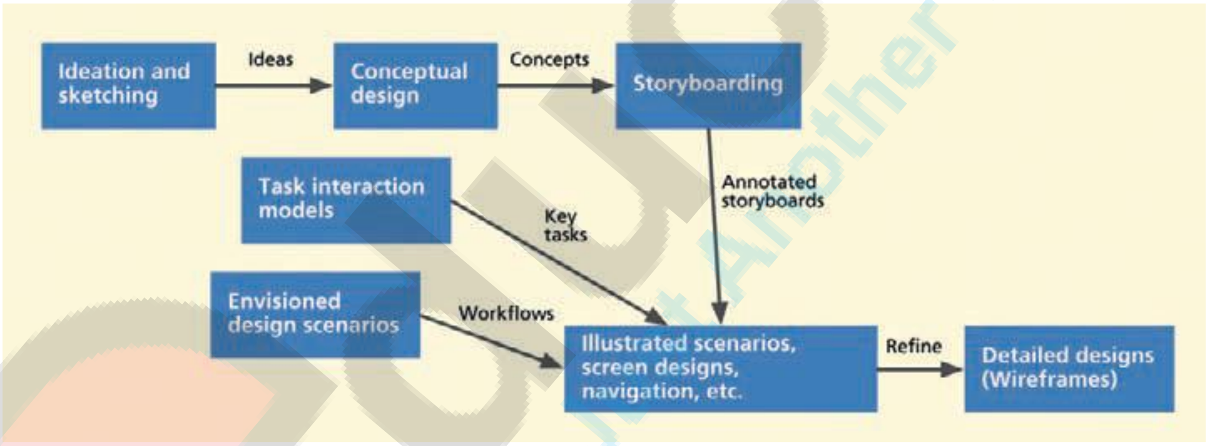


Figure 9-3 The path from ideation and sketching, task interaction models, and envisioned design scenarios to wireframes.

As an example of using wireframes to illustrate high-level conceptual designs, see Figure 9-4. The design concept depicted in this figure is comprised of a three-column pattern for a photo manipulation application. A primary navigation pane (the "nav bar") on the left-hand side is intended to show a list of all the user's photo collections. The center column is the main content display area for details, thumbnail images and individual photos, from the collection selected in the left pane. The column on the right in Figure 9-4 is envisioned to show related contextual information for the selected collection. Note how a simple wireframe using just boxes, lines, and a little text can be effective in describing a broadinteraction conceptual design pattern. Often these kinds ofpatterns areexplored during ideation and sketching, and selected sketches are translatedinto wireframes.

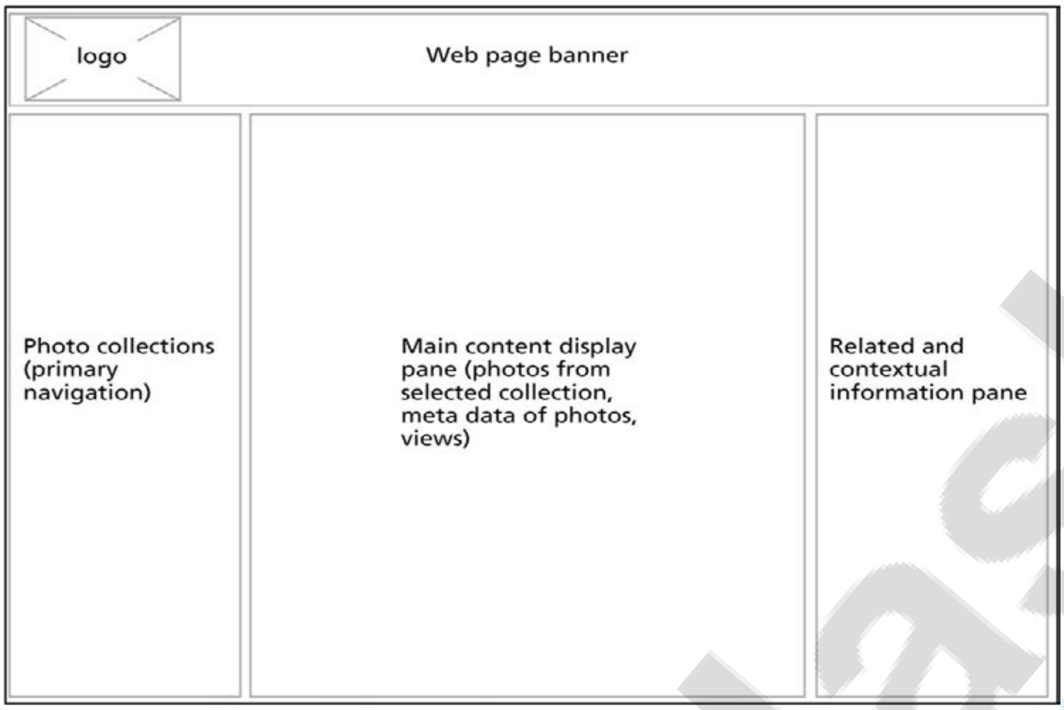

logo
Web page banner
Photo collections (primary navigation)
Main content display pane (photos from selected collection, meta data of photos, views)
Related and contextual information pane

Figure 9-4 An example wireframe illustrating a high-level conceptual design.

Wireframes are used as conversational props to discuss designs and design alternatives. They are effective tools to elicit feedback from potential users and other stakeholders. A designer can move through a deck of wireframes one slide at a time, simulating a potential scenario by pretending to click on interaction widgets on the screen. These page sequences can represent the flow of user activity within a scenario, but cannot show all possible navigational paths.

Q26.Discuss various design paradigms.(unit 4)

DESIGN PARADIGMS

In a seminal paper that we think should have receivedmore exposure, Harrison,Tatar, and Sengers (2007) paint the history of the focus of design in human–computer interaction (HCI) as a series of paradigms: engineering, humaninformation processing (HIP), and phenomenological. They get credit foridentifying the phenomenological perspective as a major design paradigmwithin the three major intellectual waves that have formed the field of HCI:

- Engineering and human factors: deconstruct work with the objective of designing themachine for optimum human performance.
- Cognitive science: the theory of what is happening in the human mind during and withrespect to interaction by treating human minds as information processors.
- The phenomenological paradigm (they call it the phenomenological matrix): emphasisin interaction is about making meaning (more on this later).

The increasing importance of social and situated actions in HCI was at oddswith both the usability-oriented engineering paradigm and the cognitivelogic of the human information processor approach. The initial reluctance ofHCI as a field to recognize and embrace the phenomenological paradigmspawned a parallel exploration in computer-supported cooperative work.Activity theory helped explain the situated actions in work practice but didnot do much to help design and evaluation. The paper by Harrison, Tatar,and Sengers (2007) is an evangelical wake-up call to include thephenomenological paradigm in mainstream HCI.

Engineering Paradigm
With some of its roots in software engineering, the HCI engineering paradigmprescribed starting with an inventory of the functionality envisioned for anew system and proceeding to build an interaction design of the best qualitypossible given available resources.With recognition that user interaction deserved attention on its own, usabilityengineering emerged as a practical approach to usability with a focus onimproving user performance, mainly through evaluation and iteration. Theengineering approach casts design as just another lifecycle phase, a systematicapproach that often works well for building systems with complex work domains.
The engineering paradigm also had strong roots in human factors, wherework was studied, deconstructed, and modeled. Here, the goal was userproductivity and eliminating user errors. An example is the study of an assemblyline where each action required to do work efficiently was described carefully.These descriptions were then more or less translated into requirements.Designs focused on how to support these requirements and to automatewhere desirable. It was a purely utilitarian and requirements-driven approach.Success was measured by how much the user could accomplish, and alternativemethods and designs were compared with statistical summative studies.

Human Information Processing (HIP) Paradigm
The human information processing approach to HCI is based on the metaphorof "mind and computer as symmetrically coupled information processors"(Tatar, Harrison, &Sengers, 2007). This paradigm, which at its base is aboutmodels of how information is sensed, accessed, and transformed in the humanmind and, in turn, how those models reflect requirements for the computer sideof the information processing, was defined by Card, Moran, and Newell (1983)and well explained by Williges (1982).The HIP paradigm has its roots in psychology and human factors, from whichit gets an element of cognitive theory. Especially as psychology is used in thediscipline of human factors, it is about human mental states and processes;it is about modelling human sensing, cognition, memory, informationunderstanding, decision making, and physical performance in task execution.The idea was that once these human parameters were codified, it is possible todesign a product that "matches" them. Guidelines, such as not having more thanseven plus or minus two items on transient lists on a user interface because oflimits on human short-term memory, were a result of this type of thinking.
Design-Thinking Paradigm
Harrison, Tatar, and Sengers (2007) propose a third HCI design paradigmthat they call the "phenomenological matrix." We call it the design-thinkingparadigm because our use of that concept goes a bit beyond their description of a "pure" phenomenological approach. This third design paradigm brings a vision of the desired user experience and product appeal and how the design of a product can induce that experience and appeal. For B_dker and Buur (2002), the third paradigm for HCI design is motivated by a desire to "reframe usability practice." The heavy priority for usability testing in traditional usability methods meant that usability concerns were being brought into the process too late and that emphasis was on refining a design but not on getting the right design in the first place [as Buxton (2007b) would say it]. They used participatory design techniques to experiment with and explore design through early prototypes as design sketches.
Another of their reasons for reframing usability practice is the fact that the usual usability techniques focused on snapshots of usage, user performance evaluated during single tasks. But they wanted to include emergence of use. They also wanted to overlap the four basic process activities—analysis, design, implementation, and evaluation—instead of "pipelining" them in an iterative process.

Q27.Differentiate between designers model and user mental model.

Q28. Write a short note on story boarding.(unit 4)

A storyboard is a sequence of visual “frames” illustrating the interplay between auser and an envisioned system. Storyboards bring the design to life in graphical“clips,” freeze-frame sketches of stories of how people will work with the system.This narrative description can come in many forms and at different levels.Storyboards for representing interaction sequence designs are like visualscenario sketches, envisioned interaction design solutions. A storyboard mightbe thought of as a “comic-book” style illustration of a scenario, with actors,screens, interaction, and dialogue showing sequences of flow from frameto frame.

Making Storyboards to Cover All Design Perspectives

From your ideation and sketches, select the most promising ideas for each of the three perspectives. Create illustrated sequences that show each of these ideas in a narrative style.

Include things like these in your storyboards:

- Hand-sketched pictures annotated with a few words
- All the work practice that is part of the task, not just interaction with the system, forexample, include telephone conversations with agents or roles outside the system
- Sketches of devices and screens
- Any connections with system internals, for example, flow to and from a database
- Physical user actions
- Cognitive user actions in “thought balloons”
- Extra-system activities, such as talking with a friend about what ticket to buy

For the ecological perspective, illustrate high-level interplay among human users, the system as a whole, and the surrounding context. Look at the envisioned flow model for how usage activities fit into the overall flow. Look in the envisioned social model for concerns and issues associated with the usage incontext and show them as user “thought bubbles.”

Q29. Explain mapping between designers and user mental model with help of diagram?(unit 4)

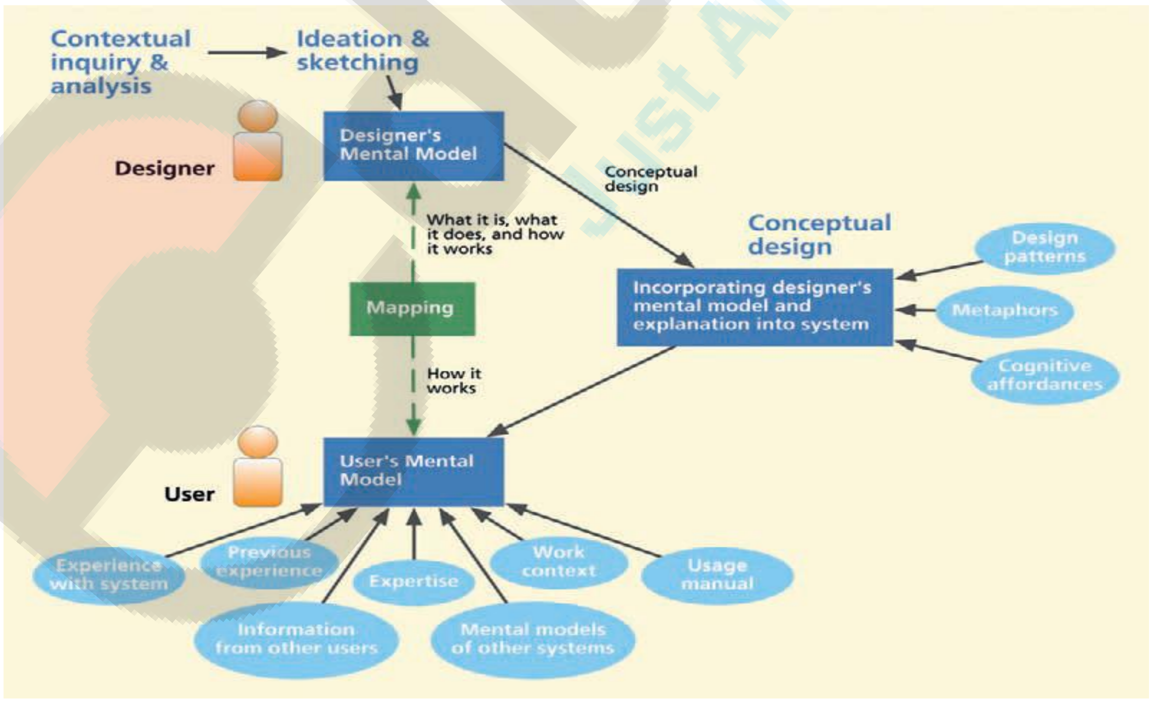


Figure 8-2 Mapping the designer’s mental model to the user’s mental model

Sometimes called a conceptual model (Johnson & Henderson, 2002, p. 26), the designer's mental model is the designer's conceptualization of the envisioned system—what the system is, how it is organized, what it does, and how it works. If anyone should know these things, it is the designer who is creating the system. But it is not uncommon for designers to "design" a system without first forming and articulating a mental model. The results can be a poorly focused design, not thought through from the start. Often such designs proceed in fits and starts and must be retraced and restarted when missing concepts are discovered along the way. The result of such a fuzzy start can be a fuzzy design that causes users to experience vagueness and misconceptions. It is difficult for users to establish a mental model of how thesystem works if the designer has never done the same.

As shown in Figure 8-2, the designer's mental model is created from what is learned in contextual inquiry and analysis and is transformed into design by ideation and sketching. Johnson and Henderson (2002, p. 26) include metaphors, analogies, ontological structure, and mappings between those concepts and the task domain or work practice the design is intended to support. The closer the designer's mental model orientation is to the user's work domain and work practice, the more likely users will internalize the model as their own. Toparaphrase Johnson and Henderson's rule for relating the designer's mentalmodel to the final design: if it is not in the designer's mental model, the systemshould not require users to be aware of it.

Q30. How does ecological, interaction and emotional design perspective is applicable for conceptual design?(unit 4)

Ecological Perspective
The ecological design perspective is about how the system or product works within its external environment. It is about how the system or product is used in its context and how the system or product interacts or communicates with its environment in the process. This is a work role and workflow view, which includes social interaction and long-term phenomenological aspects of usage as part of one's lifestyle.
System infrastructure (Norman, 2009a) plays an important role in the ecological perspective because the infrastructure of a system, the other systems and devices with which it interacts in the world, is a major part of its ecology. Infrastructure leads you to think of user activities, not just isolated usage. Norman (2009b) states it in a way that designers should take to heart, "A product is actually a service."
Interaction Perspective
The interaction design perspective is about how users operate the system or product. It is a task and intention view, where user and system come together.
It is where users look at displays and manipulate controls, doing sensory, cognitive, and physical actions.
Emotional Perspective
The emotional design perspective is about emotional impact and value-sensitive aspects of design. It is about social and cultural implications, as well as the aesthetics and joy of use. System infrastructure (Norman, 2009b) can also play a role in the emotional perspective because the infrastructure of a system provides scaffolding for the phenomenological aspects of usage, which are about broader usage contexts over longer periods of time.
A product is not just a product; it is an experience (Buxton, 2007a). People do not usually use a product in isolation from other activities. People use products as part of an activity, which can include many different kinds of usage of many different things. And that starts with the out-of-the-box experience, which is not enhanced by difficult hard plastic encasing, large user manuals, complex installation procedures, and having to consent to a legal agreement that you cannot possibly read.

Q31. Discuss the process of creating conceptual design?

A conceptual design is the part of an interaction design containing a theme, notion, or idea with the purpose of communicating a design vision about a system or product. A conceptual design is the manifestation of the designer's mental model within the system. It is the part of the system design that brings the designer's mental model to life within the system. A conceptual design corresponds to what Norman calls the "system image" of the designer's mental model (Norman, 1990, pp. 16, 189–190), about which he makes the important point: this is the only way the designer and user can communicate.
Conceptual design is where you innovate and brainstorm to plant and first nurture the user experience seed. You can never iterate the design later to yield a good user experience if you do not get the conceptual part right up front. Conceptual design is where you establish the metaphor or the theme of the product—in a word, the concept.

Start with a Conceptual Design
Johnson and Henderson (2002) introduced with conceptual design before sketching any screen or user interface objects. As they put it, screen sketches are designs of "how the systempresents itself to users. It is better to start by designing what the system is to them." Screen designs and widgets will come, but time and effort spent on interaction details can be wasted without a well-defined underlying conceptual structure. Norman (2008) puts it this way: "What people want is usable devices, which translates into understandable ones" (final emphasis ours).
To get started on conceptual design, gather the same team that did the ideation and sketching and synthesize all your ideation and sketching results into a high-level conceptualization of what the system or product is, how it fits within its ecology, and how it operates with users.
For most systems or products, especially domain-complex systems, the best way to start conceptual design is in the ecological perspective because that captures the system in its context. For product concepts where the emotional impact is paramount, starting with that perspective is obvious. At other times the "invention" of an interaction technique like that of the iPod Classic scroll wheel might be the starting point for a solution looking for a problem and is best visualized in the interaction perspective.

Q32. Explain the process of wire framework?

Or

Q32. Explain the ideation and sketching, task interaction models &envisiones design scenario is associated for wire framework. Explain with diagram?(unit 4)

REFER QUESTION NO 25

Q33. Explain wire framework with help of example?(unit 4)

As an example of using wireframes to illustrate high-level conceptual designs,see Figure 9-4. The design concept depicted in this figure is comprised of athree-column pattern for a photo

manipulation application. A primarynavigation pane (the "nav bar") on the left-hand side is intended to show a list ofall the user's photo collections. The center column is the main content displayarea for details, thumbnail images and individual photos, from the collectionselected in the left pane.
The column on the right in Figure 9-4 is envisioned to show relatedcontextual information for the selected collection. Note how a simple wireframeusing just boxes, lines, and a little text can be effective in describing a broad interaction conceptual design pattern. Often these kinds of patterns areexplored during ideation and sketching, and selected sketches are translatedinto wireframes.

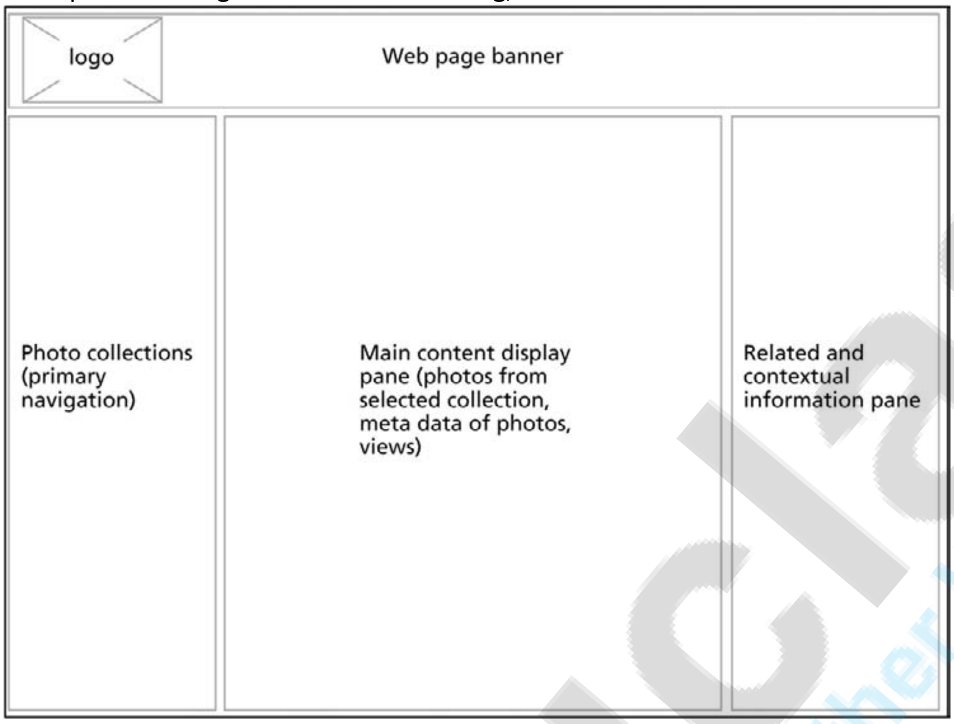

logo
Web page banner
Photo collections (primary navigation)
Main content display pane (photos from selected collection, meta data of photos, views)
Related and contextual information pane

Figure 9-4An example wireframeillustrating a high-levelconceptual design.

While wireframes can be used to illustrate high-level ideas, they are used morecommonly to illustrate medium-fidelity interaction designs. For example, theidea of Figure 9-4 is elaborated further in Figure 9.5. The navigation bar in theleft column now shows several picture collections and a default "work bench"where all uploaded images are collected. The selected item in this column, "Italytrip," is shown as the active collection using another box with the same label anda fill color of gray, for example, overlaid on the navigation bar. The centercontent area is also elaborated more using boxes and a few icons to show ascrollable grid of thumbnail images with some controls on the top right. Notehow certain details pertaining to the different manipulation options are leftincomplete while showing where they are located on the screen.
Wireframes can also be used to show behavior. For example, in Figure 9-6 weshow what happens when a user clicks on the vertical "Related information" barin Figure 9-5: a pane with contextual information for this collection (orindividual photo) slides out. In Figure 9-7 we show a different view of the content pane, this time as a result of a user clicking on the "One-up" view switcher buttonin Figure 9-5 to see a single photo in the context pane. Double-clicking athumbnail image will also expand that image into a one-up view to fill thecontent pane.

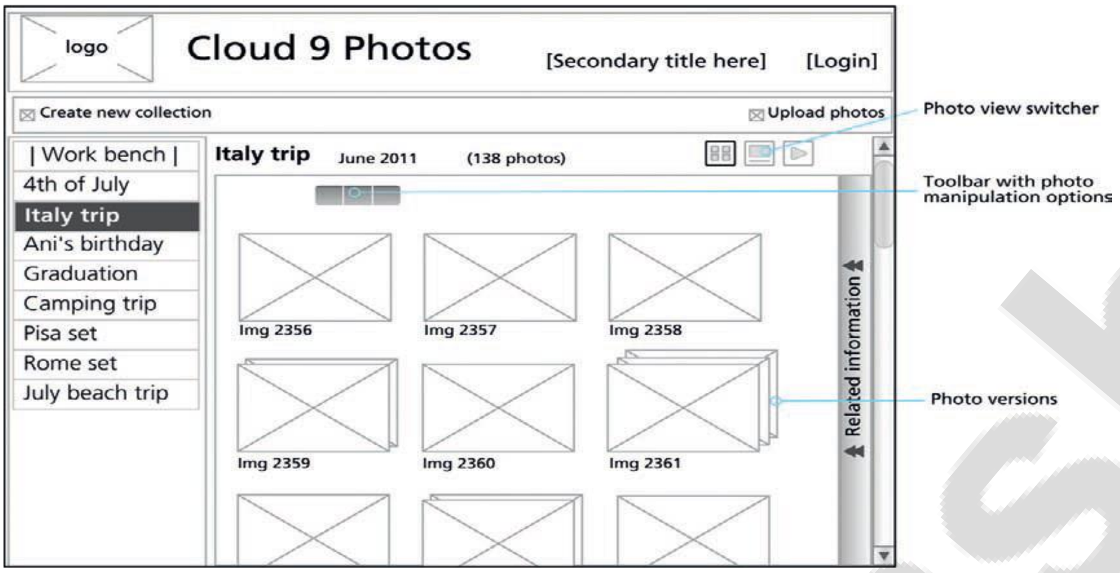


Figure 9-5 Further elaboration of the conceptual design and layout of Figure 9-4.

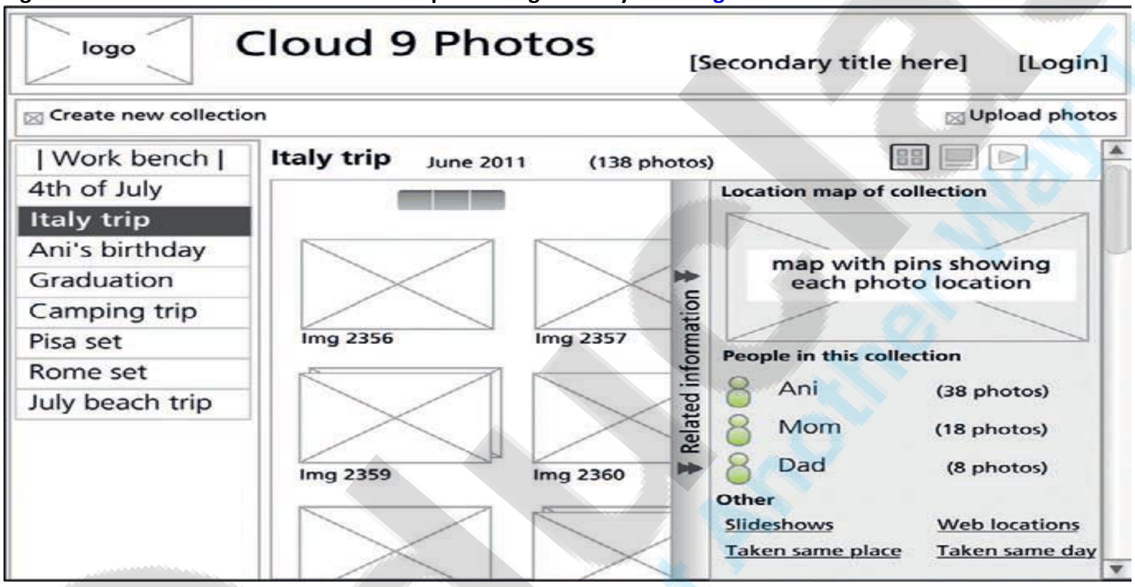


Figure 9-6 The display that results when a user clicks on the "Related information" bar.

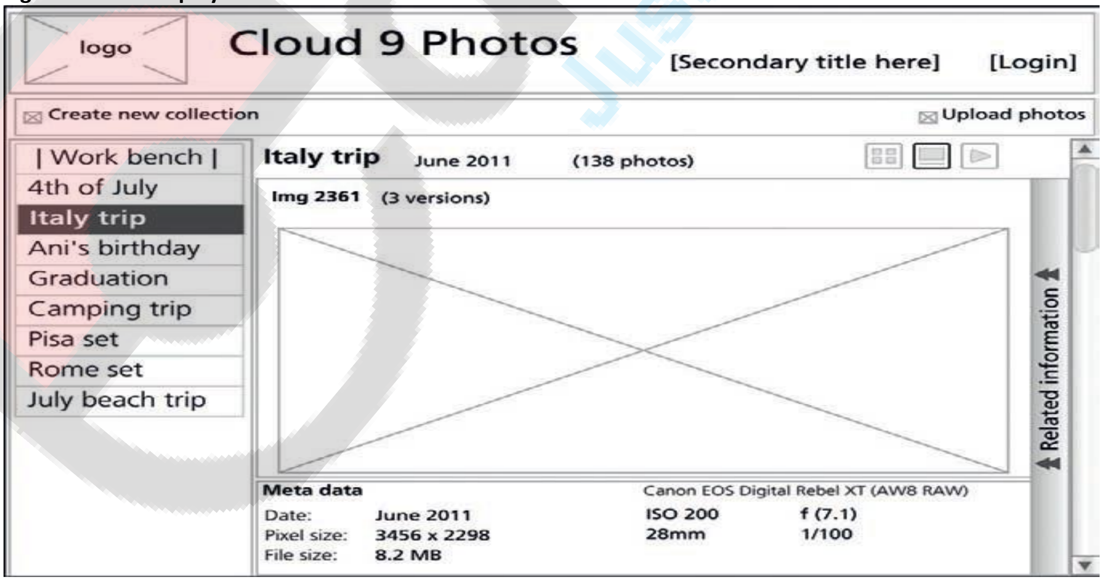


Figure 9-7 The display that results when a user clicks on the "One-up" view button.

Q34. Differentiate between formative and summative evaluation technique?(unit 5)

Formative Evaluation:

1. Formative evaluation is used during the teaching learning process to monitor the learning process.

2. Formative evaluation is developmental in nature. The aim of this evaluation is to improve student's learning and teacher's teaching.

3. Generally teacher made tests are used for this purpose.

4. The test items are prepared for limited content area.

5. It helps to know to what extent the instructional objectives has been achieved.

6. It provides feed-back to the teacher to modify the methods and to prescribe remedial works.

7. Only few skills can be tested in this evaluation.

8. It is a continuous and regular process.

9. It considers evaluation as a process.

10. It answers to the question, whether the progress of the pupils in a unit is successful?

Summative Evaluation:

1. Summative evaluation is used after the course completion to assign the grades.

2. Summative evaluation is terminal in nature. Its purpose is to evaluate student's achievement.

3. Generally standardized tests are used for the purpose.

4. The tests items are prepared from the whole content area.

5. It helps to judge the appropriateness of the instructional objectives.

6. It helps the teacher to know the effectiveness of the instructional procedure.

7. Large number of skills can be tested in this evaluation.

8. It is not regular and continuous process.

9. It considers evaluation as a product.

10. It answers to the question, the degree to which the students have mastered the course content.

Q35. Explain Ux evaluation method in detail?(unit 5)

Q36. Explain Ux evaluation technique with suitable example?(unit 5)

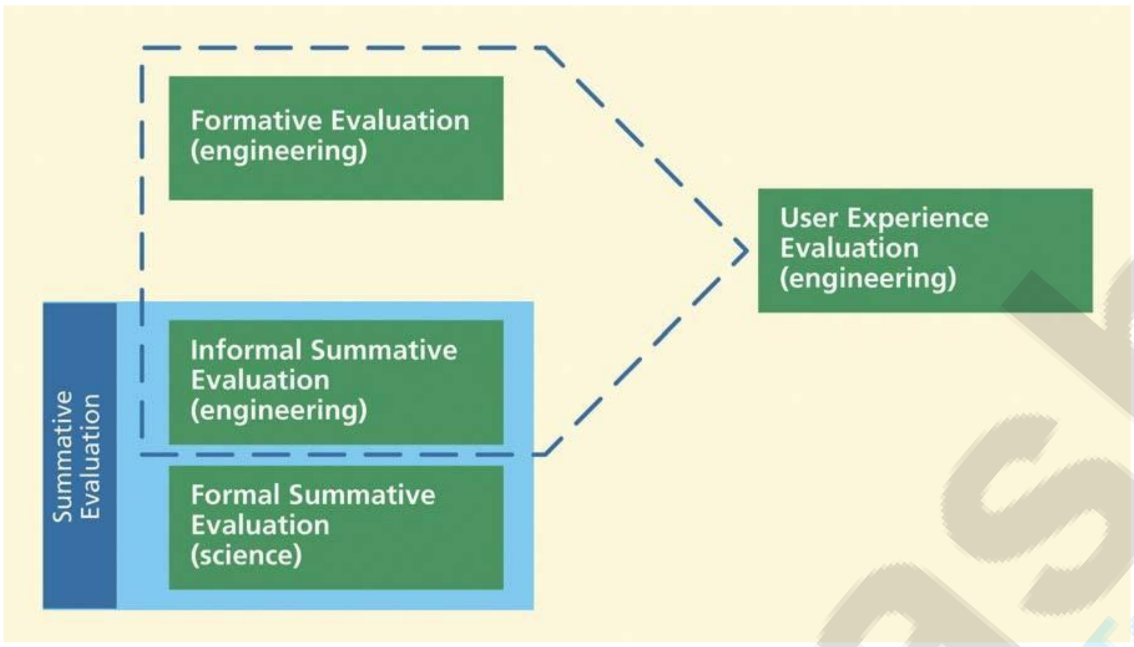
When the cook tastes the soup, that's formative; when the guests taste the soup, that's summative" (Stake, 2004, p. 17).
–**Formative evaluation**is primarily diagnostic; it is about collecting qualitative data to identify and fix UX problems and their causes in the design.

–**Summative evaluation**is about collecting quantitative data for assessing a level of quality due to a design, especially for assessing improvement in the user experience due to formative evaluation.

•**Qualitative Data**
–Qualitative data are non-numeric and descriptive data, usually describing a UX problem or issue observed or experienced during usage.
•**Quantitative Data**
–Quantitative data are numeric data, such as user performance metrics or opinion ratings.

•**Formal summative evaluation**
–Formal summative evaluation is a kind of controlled hypothesis testing with an m by n factorial design with y independent variables, the results of which are subjected to statistical tests for significance. Formal summative evaluation is an important HCI skill.
•**Informal summative evaluation**
–is used, as a partner of formative evaluation, for quantitatively summing up or assessing UX levels using metrics for user performance (such as the time on task), for example, as indicators of progress in UX improvement, usually in comparison with pre-established UX target levels.

Engineering Evaluation of UX: Formative Plus Informal Summative


Formative Evaluation (engineering)
User Experience Evaluation (engineering)
Summative Evaluation
Informal Summative Evaluation (engineering)
Formal Summative Evaluation (science)

UX evaluation is a combination of formative and informal summative evaluation.

Q37. Explain various data collection technique in detail?(unit 5)

SOME DATA COLLECTION TECHNIQUES

1. **Critical Incident Identification**

The key objective of formative evaluation is to identify defects in the interaction design so that we can fix them. But during an evaluation session, you cannot always see the interaction design flaws directly. What we can observe directly or indirectly are the effects of those design flaws on the users. We refer to such effects on the users during interaction as critical incidents. Much of the attention of evaluators in evaluation sessions observing usage is spent looking for and identifying critical incidents.

- **Critical incidents**

Despite numerous variations in procedures for gathering and analysing critical incidents, researchers and practitioners agree about the definition of a critical incident. A critical incident is an event observed within task performance that is a significant indicator of some factor defining the objective of the study.
Critical incident data about a UX problem should contain as much detail as possible, including contextual information, such as:

- the user's general activity or task
- object or artifacts involved
- the specific user intention and action that led immediately to the critical incident
- expectations of the user about what the system was supposed to do when the critical incident occurred
- what happened instead
- as much as possible about the mental and emotional state of the user
- indication of whether the user could recover from the critical incident and, if so, a description of how the user did so
- additional comments or suggested solutions to the problem

- **Relevance of critical incident data**

Critical incident identification is arguably the single most important source of qualitative data in formative evaluation. These detailed data, perishable if not captured immediately and precisely as they arise during usage, are essential for isolating specific UX problems within the user interaction design.

- **History of critical incident data**

The origins of the critical incident technique can be traced back at least to studies performed in the Aviation Psychology Program of the U.S. Army Air Forces in World War II to analyze and classify pilot error experiences in reading and interpreting aircraft instruments.

- **Mostly used as a variation**

When Flanagan designed the critical incident technique in 1954, he did not see it as a single rigid procedure. He was in favor of modifying this technique to meet different needs as long as original criteria were met. The variation occurring over the years, however, may have been more than Flanagan anticipated. Forty years after the introduction of Flanagan's critical incident technique.

- **Critical incident reporting tools**

Human factors and human–computer interaction researchers have developed software tools to assist identifying and recording critical incident information. del Galdo et al. (1986) investigated the use of critical incidents as a mechanism to collect end-user reactions for simultaneous design and evaluation of both online and hard-copy documentation.

- **Who identifies critical incidents?**

One factor in the variability of the critical incident technique is the issue of who makes the critical incident identification. In the original work by Fitts and Jones (1947), the user (an airplane pilot) reported the critical incidents after task performance was completed. Flanagan (1954) used trained observers to collect critical incident information while observing users performing tasks.

- **Timing of critical incident data capture: The evaluator's awareness zone**

While users are known to report major UX problems in alpha and beta testing (sending software out for comments on how well it worked), one reason these methods cannot be relied upon for thorough identification of UX problems to fix is the retrospective nature of that kind of data collection. Lab-based UX evaluation has the advantage of having the precious and volatile details right in front of you as they happen.

2. **The Think-Aloud Technique**

Also called "think-aloud protocol" or "verbal protocol," the think-aloud technique is a qualitative data collection technique in which user participants, as the name implies, express verbally their thoughts about their interaction experience, including their motives, rationale, and perceptions of UX problems. By this method, participants let us in on their thinking.

- **Why use the think-aloud technique?**

General observational data are important during an evaluation session with a participant attempting to perform a task. You can see what parts of a task the user is having trouble with, you can see hesitations about using certain widgets, and so on. But the bulk of real UX problem data is hidden from observation, in the mind of the participant. What is really causing a hesitation and why does this participant perceive it as a problem or barrier? To get the best qualitative data, you have to tap into this hidden data, buried in the participant's mind, which is the goal of the think-aloud technique.

- **What kind of participant works best?**

Some participants can talk while working; get them if you can. The usualparticipant for think-aloud techniques is someone who matches the work roleand user class definitions associated with the tasks you will use to drive theevaluation. This kind of participant will not be trained as a UX practitioner, but that usually will not deter them from offeringopinions and theories about UXproblems and causes in your design, which is what you want.

- **What kind of participant works best?**

Some participants can talk while working; get them if you can. The usualparticipant for think-aloud techniques is someone who matches the work roleand user class definitions associated with the tasks you will use to drive theevaluation. This kind of participant will not be trained as a UX practitioner, but
that usually will not deter them from offering opinions and theories about UXproblems and causes in your design, which is what you want.

- **How to manage the think-aloud protocol?**

The think-aloud technique is intended to draw out cognitive activity, not toconfirm observable behavior. Therefore, your instructions to participantsshould emphasize telling you what they are thinking, not describing what theyare doing.

- **Retrospective think-aloud techniques**

If, as facilitator, you perceive that the think-aloud technique, when usedconcurrently with task performance, is interfering in some way with taskperformance or task measurements, you can wait until after task completion(hence the name "retrospective") and review a video recording of the session
with the participant, asking for more complete "thinking aloud" during thisreview of his or her own task performance.

- **Co-discovery think-aloud techniques**

Using a single participant is the dominant paradigm in usability and UX testingliterature. Single users often represent typical usage and you want to be surethat single users can work their way through the tasks.

- **Does thinking aloud affect quantitative task performance metrics in lab-based evaluation?**

It depends on the participant. Some participants can naturally chat about whatthey are doing as they work. For these participants, the concurrent think-aloudtechnique usually does not affect task performance when used with measuredbenchmark tasks.

3. Questionnaires

A questionnaire is the primary instrument for collecting subjective data fromparticipants in all types of evaluations. It can be used to supplement objective(directly observable) data from lab-based or other data collection methods or asan evaluation method on its own. A questionnaire can contain probing
questions about the total user experience.

- **Semantic differential scales**

A semantic differential scale, or Likert scale (1932), is a range of valuesdescribing an attribute. Each value on the scale represents a different level ofthat attribute. The most extreme value in each direction on the scale is calledan anchor.

- **The Questionnaire for User Interface Satisfaction (QUIS)**

The QUIS, developed at the University of Maryland (Chin, Diehl, & Norman,1988) is one of the earliest available user satisfaction questionnaires for use withusability evaluation. It was the most extensive and most thoroughly validatedquestionnaire at the time of its development for determining subjective
interaction design usability.

- **The System Usability Scale (SUS)**

The SUS was developed by John Brooke while at Digital EquipmentCorporation (Brooke, 1996) in the United Kingdom. The SUS questionnairecontains 10 questions. As an interesting variation from the usual questionnaire,the SUS alternates positively worded questions with negatively wordedquestions to prevent quick answers without the responder really consideringthe questions. These 10 statements are (used with permission):

- I think that I would like to use this system frequently
- I found the system unnecessarily complex
- I thought the system was easy to use
- I think that I would need the support of a technical person to be able to use thissystem
- I found the various functions in this system were well integrated
- I thought there was too much inconsistency in this system
- I would imagine that most people would learn to use this system very quickly
- I found the system very cumbersome to use
- I felt very confident using the system
- I needed to learn a lot of things before I could get going with this system

The 10 items in the SUS were selected from a list of 50 possibilities, chosenfor their perceived discriminating power.

- **The Usefulness, Satisfaction, and Ease of Use (USE)Questionnaire**

With the goal of measuring the most important dimensions of usability forusers across many different domains, Lund (2001, 2004) developed USE, aquestionnaire for evaluating the user experience on three dimensions:usefulness, satisfaction, and ease of use. USE is based on a seven-point Likertscale.

- **Modifying questionnaires for your evaluation**

As an example of adapting a data collection technique, you can make up aquestionnaire of your own or you can modify an existing questionnaire for yourown use by:

- choosing a subset of the questions
- changing the wording in some of the questions
- adding questions of your own to address specific areas of concern
- using different scale values.

4. Data Collection Techniques Especially for EvaluatingEmotional Impact

Shih and Liu (2007), citing Dormann (2003), describe emotional impact interms of its indicators: “Emotion is a multifaceted phenomenon which peopledeliver through feeling states, verbal and non-verbal languages, facialexpressions, behaviors, and so on.” Therefore, these are the things to “measure”or at least observe or ask about.

- **Self-reported indicators of emotional impact**

While extreme reactions to a bad user experience can be easy to observe andunderstand, we suspect that the majority of emotional impact involvingaesthetics, emotional values, and simple joy of use may be perceived and felt bythe user but not necessarily by the evaluator or other practitioner. To access
these emotional reactions, we must tap into the user’s subjective feelings; oneeffective way to do that is to have the user or participant do the reporting. Thus,verbal participant self-reporting techniques are a primary way that we collectemotional impact indicators.

- **Questionnaires as a verbal self-reporting technique forcollecting emotional impact data (AttrakDiff and others)**

Questionnaires about emotional impact allow you to pose to participantsprobing questions based on any of the emotional impact factors, such as joy ofuse, fun, and aesthetics, offering a way for users to express their feelings aboutthis part of the user experience.
Reasons for using the AttrakDiff questionnaire for UX data collection include
the following:

- AttrakDiff is freely available.

- AttrakDiff is short and easy to administer, and the verbal scale is easy to understand(Hassenzahl, Beu, &Burmester, 2001; Hassenzahl, et al., 2000).
- AttrakDiff is backed with research and statistical validation. Although only the Germanlanguageversion of AttrakDiff was validated, there is no reason to believe that theEnglish version will not also be effective.
- AttrakDiff has a track record of successful application.

Observing physiological responses as indicatorsof emotional impact

In contrast to self-reporting techniques, UX practitioners can obtain emotionalimpact indicator data through monitoring of participant physiologicalresponses to emotional impact encounters as usage occurs. Usage can beteeming with user behaviors, including facial expressions, such as ephemeral grimaces or smiles, and body language, such as tapping of fingers, fidgeting, orscratching one’s head, that indicate emotional impact.

Bio-metrics to detect physiological responsesto emotional impact

The use of instrumented measurement of physiological responses inparticipants is called biometrics. Biometrics are about detection andmeasurement of autonomic or involuntary bodily changes triggered by nervoussystem responses to emotional impact within interaction events. Examplesinclude changes in heart rate, respiration, perspiration, and eye pupil dilation.Changes in perspiration are measured by galvanic skin response measurementsto detect changes in electrical conductivity.

Bio-metrics to detect physiological responsesto emotional impact

The use of instrumented measurement of physiological responses inparticipants is called biometrics. Biometrics are about detection andmeasurement of autonomic or involuntary bodily changes triggered by nervoussystem responses to emotional impact within interaction events. Examplesinclude changes in heart rate, respiration, perspiration, and eye pupil dilation.Changes in perspiration are measured by galvanic skin response measurementsto detect changes in electrical conductivity.

5. Data Collection Techniques to Evaluate

Phenomenological Aspects of InteractionLong-term studies required forphenomenological evaluation

Phenomenological aspects of interaction involve emotional impact, butemotional impact over time not emotional impact in snapshots of usage as youmight be used to observing in other kinds ofUXevaluation. The new perspectivethat the phenomenological view brings to user experience requires a new
kind of evaluation.

Goals of phenomenological data collection techniques

Regardless of which technique is used for phenomenological data collection, theobjective is to look for occurrences within long-term usage that are indicators of:

- ways people tend to use the product
- high points of joy in use, revealing what it is in the design that yields joy of use and opportunities to make it even better
- problems and difficulties people have in usage that interfere with a high-quality user experience
- usage people want but is not supported by the product
- how the basic mode of usage changes, evolves, or emerges over time
- how usage is adapted; new and unusual kinds of usage people come up with on their own

The idea is to be able to tell stories of usage and emotional impactover time.

Diaries in situated longitudinal studies

In one kind of self-reporting technique, each participant maintains a “diary,”documenting problems, experiences, and phenomenological occurrenceswithin long-term usage. There are many ways to facilitate this kind of datacapture within self-reporting, including:

- paper and pencil notes
- online reporting, such as in a blog
- cellphone voice-mail messages
- pocket digital voice recorder

We believe that the use of voice-mail diaries for self-reporting on usage hasimportance that goes well beyond mobile phone studies.

- **Evaluator triggered reporting for more representative data**

Regardless of the reporting medium, there is still the question of when the selfreporting
is to be done during long-term phenomenological evaluation. If youallow the participant to decide when to report, it could bias reporting towardtimes when it is convenient or times when things are going well with theproduct, or the participant might forget and you will lose opportunities to collect data.

- **Periodic questionnaires over time**

Periodically administered questionnaires are another self-reporting techniquefor collecting phenomenological data. Questionnaires can be used efficientlywith a large number of participants and can yield both quantitative andqualitative data. This is a less costly method that can get answers to predefinedquestions, but it cannot be used easily to give you a window into usage in context
to reveal growth and emergence of use over time.

- **Direct observation and interviews in simulatedreal usage situations**

The aforementioned techniques of self reporting, triggered reporting, andperiodic questionnaires are ways of sampling long-term phenomenologicalusage activity. As another alternative, the analyst team can simulate real long-termusage within a series of direct observations and interviews. The idea is to
meet with participant(s) periodically, each time setting up conditions toencourage episodes of phenomenological activity to occur during theseobservational periods.

Q38. Difference between qualitative and quantitative analysis.(unit 5)

Quantitative Data vs. Qualitative Data

—**Quantitative**data are numeric data, such as data obtained by user performance metrics or opinion ratings. Quantitative data are the basis of an informal summative evaluation component and help the team assess UX achievements and monitor convergencetoward UX targets, usually in comparison with the specified levels set in the UX targets.

—**Qualitative**data are non-numeric and descriptive data, usually describing a UX problem or issue observed or experienced during usage. Qualitative data are usually collected viacritical incident and/or the think aloud technique and are the key to identifying UX problems and their causes.

Q39. Note on critical incident identification?(unit 5)

Critical Incident Identification
–the user’s general activity or task

–objects or artifactsinvolved

–the specific user intention and action that led immediately to the critical incident

–expectations of the user about what the system was supposed to do when the critical incident occurred

–what happened instead

–as much as possible about the mental and emotional state of the user

–indication of whether the user could recover from the critical incident and, if so, a description of how the user did so

–additional comments or suggested solutions to the problem
–Relevance of critical incident data

–History of critical incident data

–Mostly used as a variation

–Critical incident reporting tools

–Who identifies critical incidents?

–Timing of critical incident data capture: The evaluator’s awareness zone

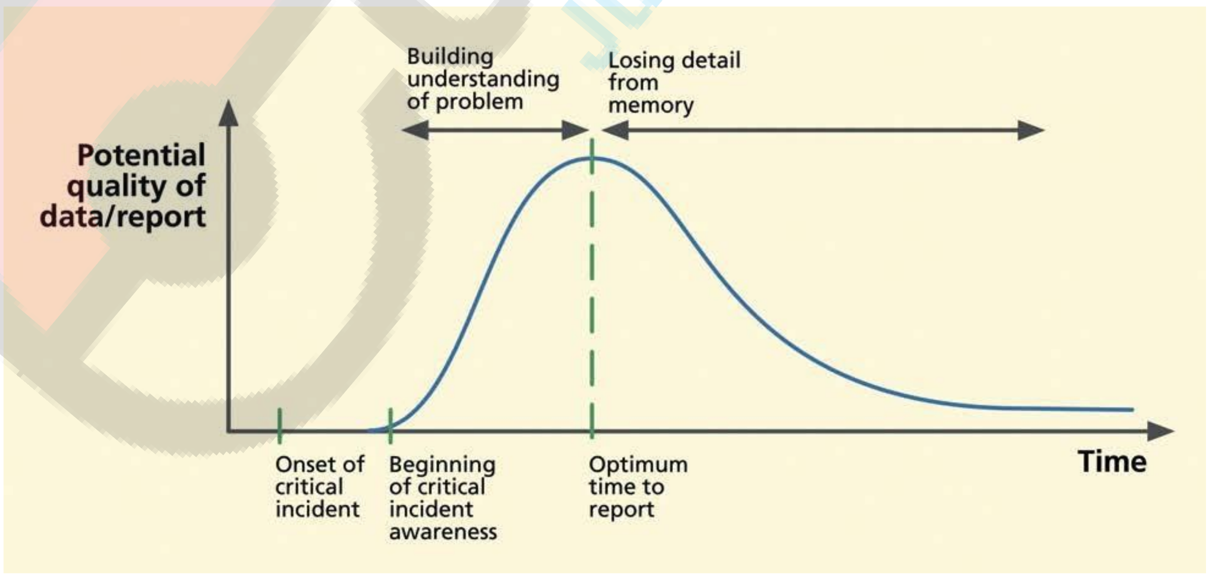


Fig: Critical incident description detail vs. time after critical incident.

Q40. Explain standard guideline for rigorous Empirical evaluation?(unit 5)

Rigorous evaluation

Formative UX evaluation methods can be either rigorous or rapid. We definerigorous UX evaluation methods to be those methods that maximizeeffectiveness and minimize the risk of errors regardless of speed or cost,meaning to refrain from shortcuts or abridgements.Rigorous empirical UX evaluation methods entail a full process ofpreparation, data collection, data analysis, and reporting. In practical terms, this kind of rigorous evaluation is usuallyconducted in the UX lab. Similarly, the same kind of evaluation can beconducted at the customer's location in the field.
Rigorous empirical methods such as lab-based evaluation, while certainly notperfect, are the yardstick by which other evaluation methods are compared.Rigorous and rapid methods exist mainly as quality vs. cost trade-offs.
n Choose a rigorous empirical method such as lab-based testing when you need effectiveness and thoroughness, but expect it to be more expensive and timeconsuming.
n Choose the lab-based method to assess quantitative UX measures and metrics, such as time-on-task and error rates, as indications of how well the user does in a performanceoriented context.
n Choose lab-based testing if you need a controlled environment to limit distractions.
n Choose empirical testing in the field if you need more realistic usage conditions for ecological validity than you can establish in a lab.

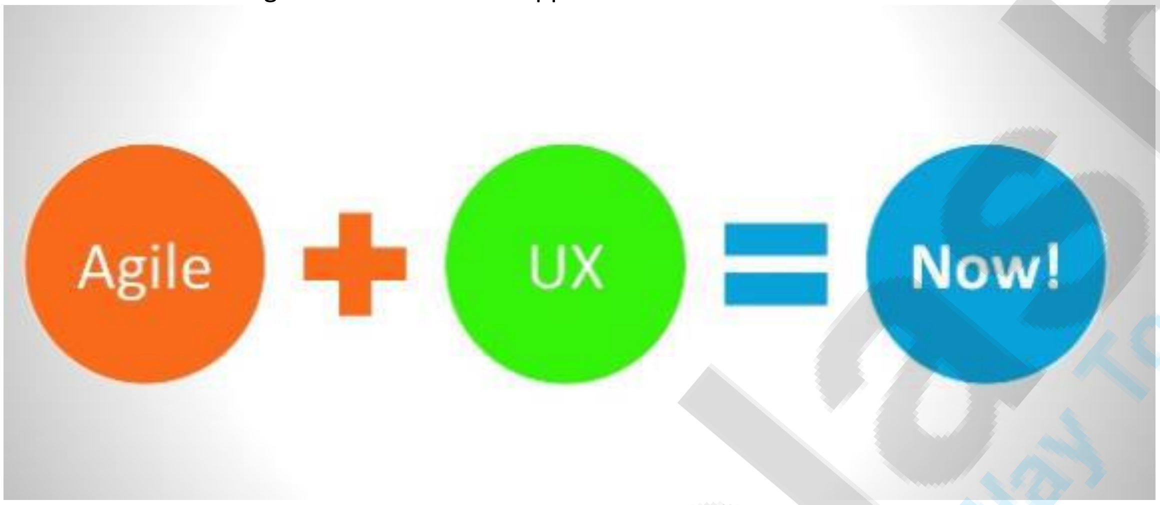
Empirical evaluation

Empirical methods are sometimes called "payoff methods" (Carroll,Singley, & Rosson, 1992; Scriven, 1967) because they are based on how a designor design change pays off in terms of real observable usage. Examples of the kindof data collected in empirical methods include quantitative user performance
data, such as time on task and error rates, and qualitative user data derived fromusage observation, such as UX problem data stemming from critical incidentidentification and think-aloud remarks by user participants. Analytical methodsare sometimes called "intrinsic methods" because they are based on analysing intrinsic characteristics of the design rather than seeing the design in use.
In describing the distinction between payoff and intrinsic approaches toevaluation, Scriven wrote an oft-quoted analogy featuring an axe "If you want to evaluate a tool, say an axe, you might study the design of the bit,the weight distribution, the steel alloy used, the grade of hickory in the handle,etc., or you might just study the kind and speed of the cuts it makes in the handsof a good axeman," speaking of intrinsic and payoff evaluation, respectively.In Hartson, Andre, and Williges (2003) we added our own embellishments,which we paraphrase here.Although this example served Scriven's purpose well, it also offers us a chanceto make a point about the need to identify UX goals carefully before establishingevaluation criteria. Giving a UX perspective to the axe example, we note thatuser performance observation in payoff evaluation does not necessarily requirean expert axeman (or axeperson). Expert usage might be one component of thevision in axe design, but it is not an exclusive requirement in payoff evaluation.UX goals depend on expected user classes of key work roles and the expected
kind of usage.

Q41. How agile SE is important for Ux designing?(unit 6)

"Agile development is a rapid and lightweight approach to software and system development characterized by ultra fine grained iteration. Coding begins very early, producing early and frequent small releases representing small but working functionality."

- Agile SE approaches, now well known and popularly used, are incremental, iterative, and test-driven means of delivering pieces of useful working software to customers frequently, for example, every two weeks.
- Therefore, the entire system development team needs an overall approach that includes UX while retaining the basics of the SE approach.



Agile SE development methods begin coding very early. Agile SE has a shorter, almost nonexistent, requirements engineering phase and far less documentation than that of traditional software engineering.

To clarify the concept of agile software methods, a group met at a workshop in Snowbird, Utah in February 2001 and worked out an "agile manifesto" (Beck, 2000). From their stated principles behind this manifesto, goals for agile software development emerged:

–Satisfy the customer by giving them early and continuous deliverables that produce valuable and working software.

–Recognize that changing requirements are the norm in any software development effort.

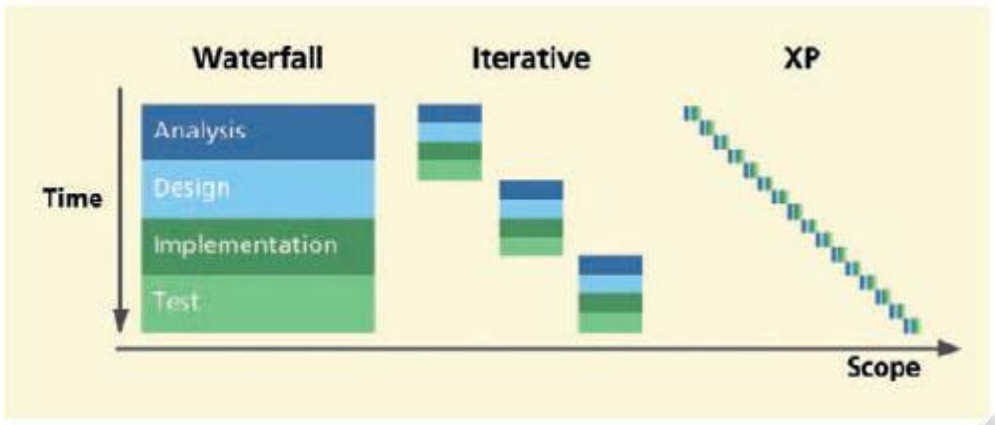
–Understand that time and budget constraints must be managed.
Practitioners of agile SE methods value (Beck, 2000):
–Individuals and interactions over processes and tools

–Working software over comprehensive documentation

–Customer collaboration over contract negotiation

–Responding to change over following a plan

In agile approaches developers do just enough—a micro-level of each activity—to support one small feature request.



Comparison of scope of development activities across methodologies, taken with permission from Beck