Managing Design Processes

- Human-computer interaction at its core is a design discipline
- Design can be loosely defined as the outcome or the process of creating specifications for synthetic artifacts, such as products, services, and processes
- *Interaction design* is the design of the digital interface itself.
- Interaction design is now an established design discipline in its own right.
- Interaction design is defined as making plans and specifications for digital objects, which include devices, interfaces, services, and information

- Early computer applications were designed by programmers to be highly functional for the programmers themselves and their peers,
- This approach quickly failed when the audience for computers grew to non-technical fields
- The current generation of users for smartphones, social media, and e- commerce have vastly different backgrounds from programmers and engineers.
- They have no interest in obscure interfaces but are more oriented toward their professional or recreational needs and are less dedicated to the technology itself.
- Therefore, effective interaction design takes the intended user as its starting point and focuses on facilitating the function of the artifact.

- Organisational support for Design

- The key to good design starts in the organization itself.
- Design is unpredictable, which requires an agile organizational structure as well as a comprehensive business strategy oriented around diverse design processes.
- The organizational setup and policies should give appropriate emphasis to support usability

Benefits of adopting Usability engineering methods

- *Return on investment* (ROI) for usability engineering and interaction design.

- Report published by IBM shows up to $100 payoffs for each dollar spent on usability,
- Benefits in reduced program-development costs, reduced maintenance costs, increased revenue due to higher customer satisfaction, and improved user efficiency and productivity.
- Usability Engineering methods lead to shortened learning times, faster performance, lower error rates on well-designed interfaces.
- Higher rates of conversion, enlarged market share, and increased customer retention in E commerce, fewer returns/complaints, increased brand loyalty, and more referrals in consumer goods

.

Character of Design

- Design is a *process;* it is not a state and it cannot be adequately represented statically.
- The design process is *nonhierarchical;* it is neither strictly bottom-up nor strictly top-down.
- The process is *radically transformational;* it involves the development of partial and interim solutions that may ultimately play no role in the final design.
- Design intrinsically involves the *discovery of new goals.*

The Three Pillars of Design

- The three pillars help user-interface architects to turn good ideas into successful systems
- They are not guaranteed to work, but experience has shown that each pillar can produce an order-of-magnitude speedup in the process and can facilitate the creation of excellent systems.

- 1. Guidelines, documents and processes

- A readable guidelines document provides a clear set of principles for the many application developers to follow and thus ensures harmony in design across products.
- Each project has different needs, but guidelines should be considered for:
- Words, icons, and graphics

  Terminology (objects and actions), abbreviations, and capitalization

  Character set, fonts, font sizes, and styles (bold, italic, underline)

  Icons, buttons, graphics, and line thickness

  Use of color, backgrounds, highlighting, and blinking
- **Screen-layout issues**

  Menu selection, form fillin, and dialog-box formats

  Wording of prompts, feedback, and error messages

  Justification, whitespace, and margins

  Data entry and display formats for items and lists

  Use and contents of headers and footers

  Strategies for adapting to small and large displays

•• Input and output devices

Keyboard, display, cursor control, and pointing devices Audible sounds, voice feedback, touch input, and other special input modes or devices

Response times for a variety of tasks

Alternatives for users with disabilities

- Action sequences

  Direct-manipulation clicking, dragging, dropping, and gestures

  Command syntax, semantics, and sequences

  Shortcuts and programmed function keys

  Error handling and recovery procedures

- Training

  Online help and tutorials

  Training and reference materials

Recommendations for guidelines documents.

- • Provides a social process for developers
- • Records decisions for all parties to see
- • Promotes consistency and completeness
- • Facilitates automation of design
- • Allows multiple levels:
    - Rigid standards
    - Accepted practices
    - Flexible guidelines
- • Announces policies for:
        - Education: how to get it?
        - Enforcement: who reviews?
        - Exemption: who decides?
        - Enhancement how often?

2. User Interface tools

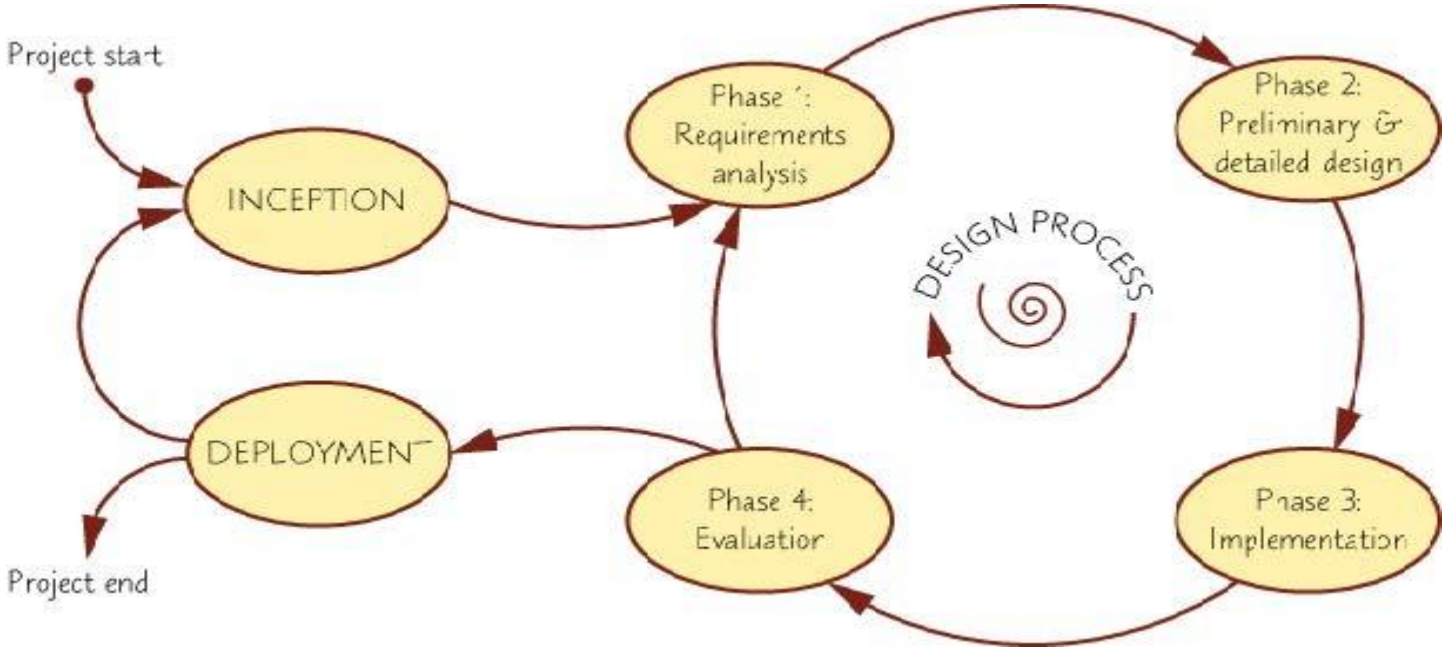
- Graphical design environments such as Macromedia's Director and Flash are widely used for developing prototypes.
- Development environments such as Microsoft's Visual Basic/C++ are also widely used as they are easy to get started and have a rich set of powerful features.

- One difficulty in designing interactive systems is that customers and users may not have a clear idea of what the system will look like when it is done.
- Since interactive systems are novel in many situations, users may not realize the implications of design decisions.
- Unfortunately, it is difficult, costly, and time consuming to make major changes to systems once those systems have been implemented.
- Even though this problem has no complete solution, some of the more serious difficulties can be avoided if, at an early stage, the customers and users can be given a realistic impression of what the final system will look like through a prototype.

3. Expert reviews and Usability Testing

- web-site designers need to carry out many small and some large pilot tests of components before release to customers .
- This can be done by a variety of expert review methods, tests with the intended users, surveys, and automated analysis tools

An iterative design process for interaction design



- An iterative design process based on this operational definition consists of four distinct phases
- Requirements analysis (Phase 1),
- Preliminary and detailed design (Phase 2),
- Build and implementation (Phase 3), and
- Evaluation (Phase4).

- The primary feature of this process is that it is *iterative* and *cyclical*
- Second, there are several crosscutting factors that contribute to each phase of the cycle, including academic and user research, guidelines and standards, and tools and patterns.

- Our focus here is purely on the human and social aspects of an interactive system or product, but the overall design process encompasses also technical aspects

- Many technical design processes, such as in software engineering, follow a similar four phase cycle, allowing interaction design and engineering to be integrated with them easily.

Phase 1: Requirements analysis

- In this phase all the necessary requirements for an interactive system or device and yields a requirements specification or document as its outcome.
- Soliciting, capturing, and specifying user requirements are major keys to success in any development activity
- Different methods are employed but the end result is the same: a clear specification of the user community and the tasks the users perform

- Collecting interaction design requirements is part of the overall requirements analysis and management phase and often has a direct impact on the engineering aspects of the design.
- for example, a finger painting app requires a multi-touch display with low touch latency.

- Requirements documents written specifically for user experience and interaction design aspects are often specified in terms of three components

- **Functional requirements** define specific behavior that the system should support
- **Non-functional requirements** specify overall criteria governing the operation of the interactive system without being tied to a specific action or behavior (hardware, software, system performance, reliability, etc.);
- **User experience requirements** explicitly specify non-functional requirements for the user interaction and user interface of the interactive system (navigation, input, colors, etc.).

- Requirements documents provide a shared understanding between the members of the product team.
- The success or failure of software projects often depends on the precision and completeness of this

understanding between all the designers, developers, and users.

- What happens without adequate requirements definition?
- You are not sure what problem you are solving, and you do not know when you are done.

Phase 2 Preliminary and Detailed design

- The core of the design process is realizing the requirements from the previous phase.
- The design phase in turn consists of two stages: a preliminary stage, where the high-level design or architecture of the interactive system is derived, and a detailed stage, where the specifics of each interaction are planned out.
- The outcome from the design phase is a detailed design document.

- The preliminary design is also known as *architectural design*, and in engineering settings this stage includes deriving the architecture of the system.
- For user experience and interaction design, preliminary design consists of mapping out the high-level concepts such as the user, controls, interface displays, navigation mechanisms, and overall workflow.
- . Overall, this activity is about developing the mental model that users should have about the interactive system when using it.
- Is your system focused on a central view, such as a map or a table, or is it a sequence of forms or a set of linked displays?
- Is it an app that integrates with other apps to pop up on demand, or is it intended for focused, sustained use?
- These are questions to answer and refine during this stage.

- The high-level concepts and their relations provide a starting point for the detailed design.
- This stage entails planning out all of the operations that take place between user and interactive system to a level where only implementation and technical details remain

- Phase 3: Build and implementation

- The implementation phase is where all of the planning gets turned into actual, running code.
- The outcome from this phase is a working system, albeit not necessarily the final one.

- We will not discuss the actual l software and hardware engineering needed to achieve this.
- :

Phase 4: Evaluation

- In the final phase of the design cycle, developers test and validate the system implementation to ensure that it conforms to the requirements and design set out earlier in the process.
- The outcome of the validation process is a validation report specifying test performance.
- Depending on this outcome, the design team can decide to proceed with production and deployment of the system or to continue another cycle through the design process.

- Design Frameworks
- Approach to conducting the design Process

- Many software development projects fail to achieve their goals.
- Some estimates of the failure rate put it as high as 60%.
- Much of this problem can be traced to poor communication between developers and their business clients or between developers and their users.
- Software-engineering methodologies are effective in facilitating the software development process, however, they do not provide a clear processes for studying the users, understanding their needs, and creating a usable interface.

User Centered Design

- Careful attention to user-centered design issues at the early stages of software development dramatically reduces both development time and cost.
- User-centered design leads to systems that generate fewer problems during development and have lower maintenance costs over their lifetime.
- They are easier to learn, produce faster performance, reduce user errors substantially, and encourage users to explore features that go beyond the minimum required to get by.
- User-centered design practices help organizations align system functionality with their business needs and priorities

- The development of human-centric software solutions is becoming increasingly important.

- By human-centered software we mean software that is made by people for people and is user-appropriate and has functionality that is needed by users

- UCD prescribes a design process that primarily takes the needs, wants, and limitations of the actual end users into account during each phase of the design process .
- UCD reduces the risk of designers building the “wrong system”: a system that the end users neither need nor asked
- ,

Participatory Design

Participatory design (PD) (also known as *cooperative design*) is the direct involvement of people in the collaborative design of the things and technologies they use.

- More user involvement brings more accurate information about tasks and an opportunity for users to the other hand, extensive user involvement may be costly and may lengthen the impleinfluence design decisions and increased user acceptance of the final system
- On mentation period.
- It may also generate antagonism from people who are not involved or whose suggestions are rejected. and potentially force designers to compromise their designs to satisfy incompetent participants

- The *Logical User-Centered Interactive Design Methodology* (LUCID) identifies six stages

- **Stage 1: Envision**
- Align the agendas of all stakeholders, balancing the needs to meet business objectives, manage technical constraints and support users' needs for a highly usable product.
- Develop a clear, shared product vision among the stakeholders.
- Identify and deal with potential problems that could impair the development team's ability to collaborate effectively.
- Begin the design process at a concept sketch level.
- **Stage 2: Discovery**
- *Develop* a clear understanding of the characteristics of each distinct segment of the product's users.
- Understand the tasks users perform, the information they need, the terminology they use, their priorities and their mental models.
- Analyze the data gathered and create the product's user requirements.

- **Stage 3: Design Foundation**

- Develop and validate the basic conceptual design of the product.
- Develop a visual look for the product.
- Present the completed design as a key screen prototype.

- **Stage 4: Design Detail**
  - Complete a style guide containing both the graphic design and UI policy decisions.
  - Flesh out the high-level design into a complete specification.
  - Conduct usability evaluations of specific screens or workflows.
  - Create detailed layouts for each screen and detailed specifications for each element of each screen.

- **Stage 5: Build**
- Answer questions and support developers during coding, redesigning screens if needed.
- Conduct usability evaluation of critical screens, if necessary.
- Support the build process through *review* and late-stage change management.
- **Stage 6: Release**
- Develop a rollout plan to support the new product
- Conduct usability evaluation of the "out of the box" or installation experience.
- Measure user satisfaction.

- A distinctive aspect of LUCID is its focus on a key-screen prototype that incorporates the major navigational paths of the system.
- The key-screen prototype is used to show users the design of the proposed system and allow them to evaluate and refine it.
- The key-screen prototype is also used for usability testing and heuristic review.
- Key screens usually evoke strong reactions, generate early participation, and create momentum for the project

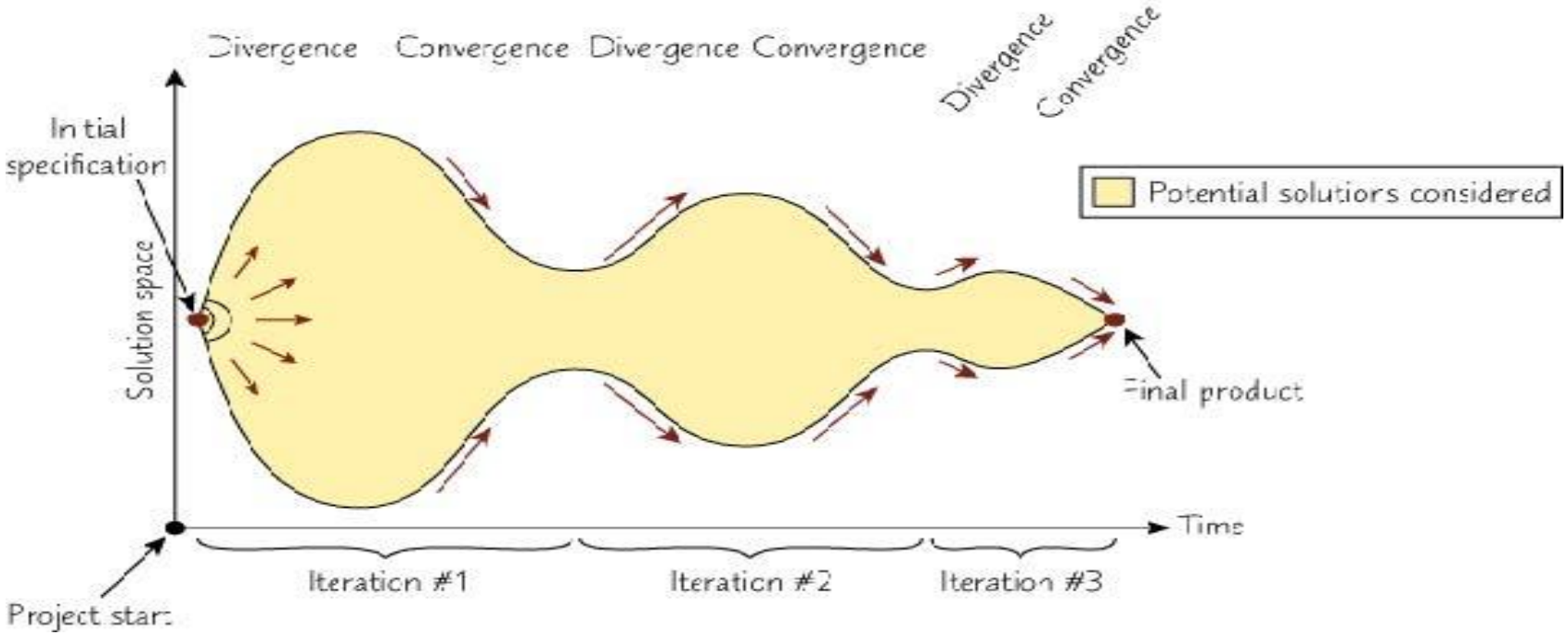
, LUCID employs rapid prototyping and iterative usability testing .

- Rapid prototyping is key to meeting schedule and budget requirements,.
- The prototypes are usually developed in conjunction with a programmer who is part of the software engineering team.
- One of this programmer's responsibilities is to identify interface issues that have implications for the technical architecture of the product.
- When completed and approved by users, the prototype serves as part of the programming specification for the software engineers.

- Design Methods

- Design methods are the practical building blocks that form the actual day-to-day activities in the design process
- Relation between design frameworks and design methods?
- Specific design frameworks have an affinity to specific design methods;
- for example, participatory and user-centered design tends to incorporate a lot of ethnographic observation, whereas rapid and agile development employs sketching to a high degree

Ideation and Creativity



- Design is as an incremental fixation of the solution space, where the range of possible solutions is gradually whittled down until only a single solution exists.

- *This is called convergence* or *convergent thinking*

- interleaving divergent and convergent thinking iteratively leads to a well-rounded and balanced design process that considers a large portion of the potential solution space.
- Ideation (or idea generation) and creativity techniques are methods for such divergent thinking in that they require designers to test their limits, abandon their assumptions, and reframe their problems.
- Many creativity techniques exist in the literature, including lateral thinking, brainstorming and brainwriting, improvisation and role playing, and aleatoricism (incorporation of chance) and bootlegging .

- Brainstorming often results in *mind maps* that show the main concepts as bubbles with links describing relations between the concepts.
- *sketching* facilitates both divergence and convergence by inviting both common ground and consensus as well as deviation and diversity.

Surveys, interviews, and focus groups

- The most straightforward way to elicit requirements and desires from users is simply to ask them

- . 1.Surveys—online or paper-based—are the simplest and cheapest approach and simply entail distributing a questionnaire to representative users

Interviews

- In-person interviews are more labor-intensive than surveys but will yield more accurate and high-quality responses.
- Interviews can take place either in one-on-one settings between designer and user or in *focus group* discussions with multiple users and designers. Again, the
- choice between individual or group interviews depends largely on cost; a group session requires less time investment but may not be able to collect in-depth feedback from all participants.
- However, group dynamics yield synergistic effects where one participant's response may trigger additional feedback from other participants.
- group interviews are often used in UCD and PD design frameworks.
- Interviews may be structured or unstructured.
- *Structured interviews* are essentially verbal surveys, but the method often lets the designer follow up with additional questions based on the answer.
- *Unstructured interviews* have no specific questions to ask the user,
- only a general discussion topic

Ethnographic observation

Guidelines for conducting ethnographic studies for interaction design

- ***Preparation***
- Understand policies in the target environment (work, home,public space, etc.).
- Familiarize yourself with the existing interface and its history.
- Set initial goals and prepare questions.
- Gain access and permission to observe or interview
- ***Field study***
- Establish a rapport with all users.
- Observe or interview users in their setting, and collect subjective and objective quantitative and qualitative data.
- Follow any leads that emerge from the visits.
- Record your visits.

- ***Analysis***
- Compile the collected data in numerical, textual, and
- multimedia databases.
- Quantify data and compile statistics.
- Reduce and interpret the data.
- Refine the goals and the process used.
- ***Reporting***
- Consider multiple audiences and goals.
- Prepare a report and present the findings

Advantages of ethnographic studies

- It increase trustworthiness and credibility, since designers learn about the complexities of the intended environment by visits to the workplace, school, home, or other environment where the eventual system will be deployed.
- Personal presence allows designers to develop working relationships with several end users to discuss ideas.

Scenario development and storyboarding

- Developing Use case scenarios for specific tasks of the user
- Storyboarding is the use of graphical sketches and illustrations to convey important steps in a
- scenario.
- Several additional methods for scenario development are useful.
- A flowchart or transition diagram helps designers to record and convey the sequences of possible
- actions; the thickness of the connecting lines indicates the frequency of the transitions.
- For example an easy way to describe a novel system is to write scenarios of usage and then, if possible, to act them out as a form of theater.
- This technique can be especially effective when multiple users must cooperate (for example, in control rooms, cockpits, or financial trading rooms) or multiple physical devices are used

Prototyping

- Types of prototypes at different levels of fidelity:
- **Low-fidelity prototypes** are generally created by sketching, using sticky notes, or cutting and gluing pieces of paper together (paper mockups);
- **Medium-fidelity prototypes** are often called *wireframes*, provide some standardized elements (such as buttons, menus, and text fields), even if potentially drawn in a sketchy fashion, and have some basic navigation functionality
- **High-fidelity prototypes** look almost like the final product and may have some rudimentary computational capabilities; however, the prototype is typically not complete and may not be fully functional.

Design tools

- prototypes can be developed with simple drawing or word-processing tools or even Microsoft
- PowerPoint presentations of screen drawings manipulated with PowerPoint slideshows and other animation.
- Other design tools that can be used are Adobe InDesign , Photoshop , or Illustrator