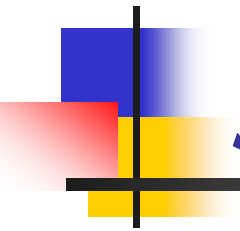


COMP 354: Introduction to Software Engineering

User Experience Design

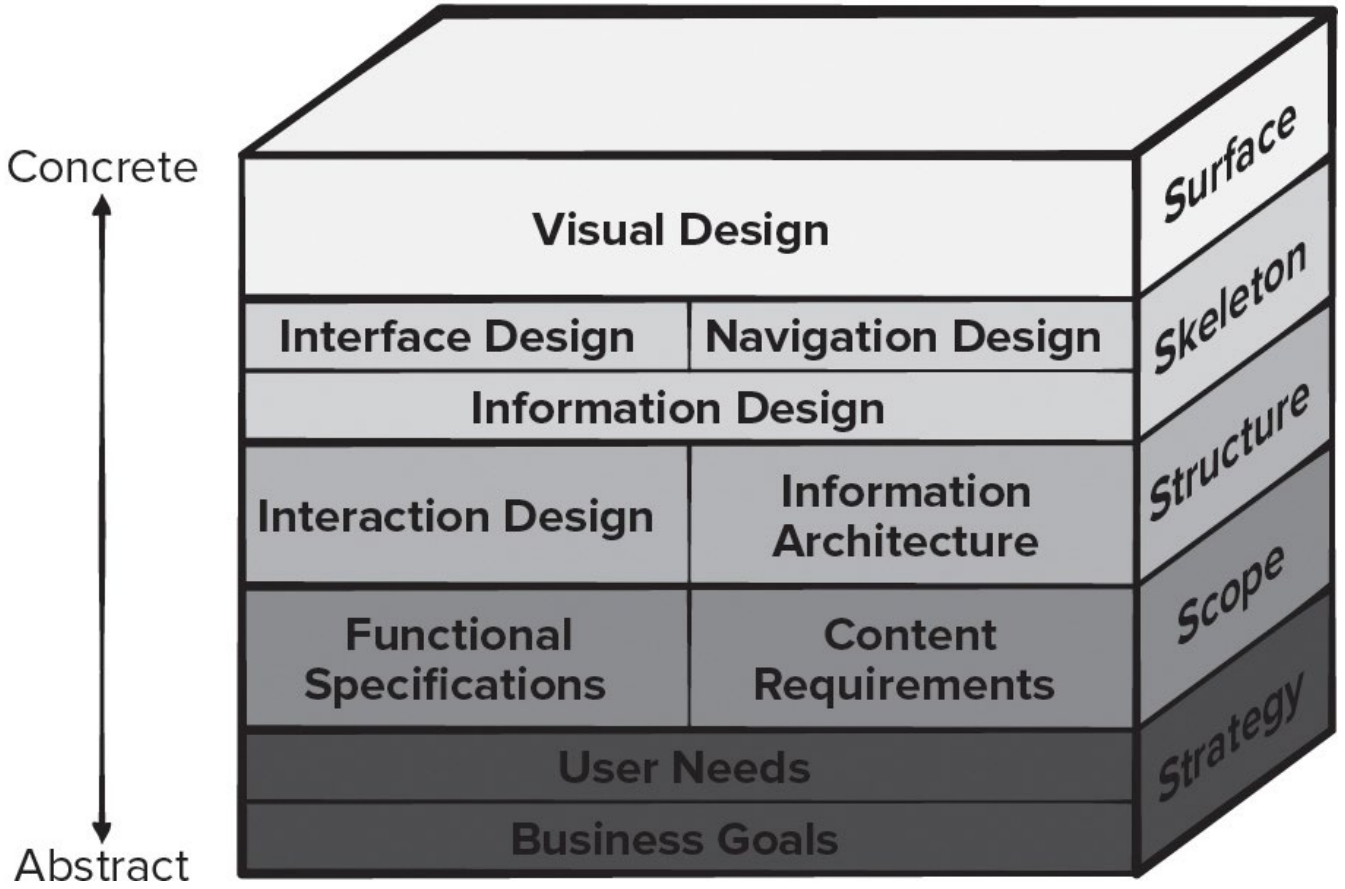
Based on Chapter 12 of the textbook

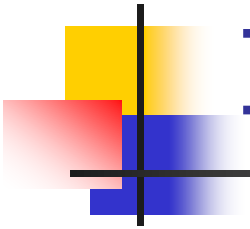
User Experience Design Elements

User experience design tries to ensure that no aspect of your software appears in the final product without the explicit decision of stakeholders to include it.

- **Strategy**. Identifies user needs and customer business goals that form the basis for all U X design work.
- **Scope**. Includes both the functional and content requirements needed to realize a feature set consistent with the project strategy.
- **Structure**. Consists of the interaction design [For example, how the system reacts in response to user action] and information architecture.
- **Skeleton**. Comprised of three components: information design, interface design, navigation design.
- **Surface**. Presents visual design or the appearance of the finished project to its users.

User Experience Design Elements





Information Architecture

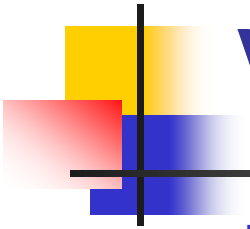
- Information architecture structures lead organization, labeling, navigation, and searching of content objects.
- Content architecture focuses on the manner content objects are structured for presentation and navigation.
- Software architecture addresses the manner the application is structured to manage user interaction, effect navigation, and present content.
- Architecture design is conducted in parallel with interface design, aesthetic design, and content design.
- Decisions made during architecture design action will influence work conducted during navigation design.

User Interaction Design

- Interaction design focuses on interface between product and user.
- Modes of user input and output include voice input, computer speech generation, touch input, 3D printed output, immersive augmented reality experiences, and sensor tracking of users.
- User interaction should be defined by the stakeholders in the user stories created to describe how users can accomplish their goals using the software product.
- User interaction design should also include a plan for how information should be presented within such a system and how to enable the user to understand that information.
- It is important to recall that the purpose of the user interface is to present just enough information to help the users decide what their next action should be to accomplish their goal and how to perform it.

Usability Engineering

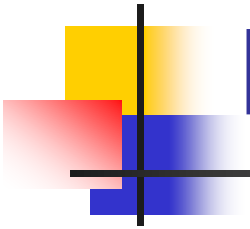
- Usability engineering is part of UX design work that defines the specification, design, and testing of the human-computer interaction portion of a software product.
- This software engineering action focuses on devising human-computer interfaces that have high usability.
- If developers focus on making a product easy to learn, easy to use, and easy to remember over time, usability can be measured quantitatively and tested for improvements in usability.
- Accessibility is the degree to which people with special needs are provided with a means to perceive, understand, navigate, and interact with computer products.
- Accessibility is another aspect of usability engineering that needs to be considered during design.



Visual Design

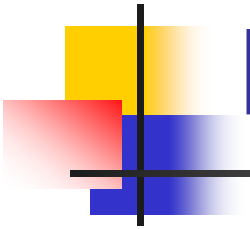
- Visual design (aesthetic design or graphic design) is an artistic endeavor that complements the technical aspects of the user experience design.
- Without it, a software product may be functional, but unappealing.
- With it, a product draws its users into a world that embraces them on an emotional as well as an intellectual level.
- Graphic design considers every aspect of the look and feel of a web or mobile app.
- Not every software engineer has artistic talent. If you fall into this category, hire an experienced graphic designer to help.

Golden Rule 1: Place User in Control



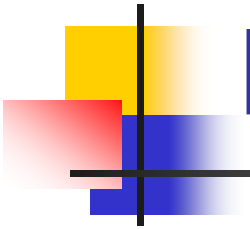
- Define interaction modes in a way that does not force a user into unnecessary or undesired actions.
- Provide for flexible interaction.
- Allow user interaction to be interruptible and undoable.
- Streamline interaction as skill levels advance and allow the interaction to be customized.
- Hide technical internals from the casual user.
- Design for direct interaction with screen objects.

Golden Rule 2: Reduce User's Memory Load



- Reduce demand on short-term memory.
- Establish meaningful defaults.
- Define shortcuts that are intuitive.
- The visual layout of the interface should be based on a real-world metaphor.
- Disclose information in a progressive fashion.

Golden Rule 3: Make Interface Consistent

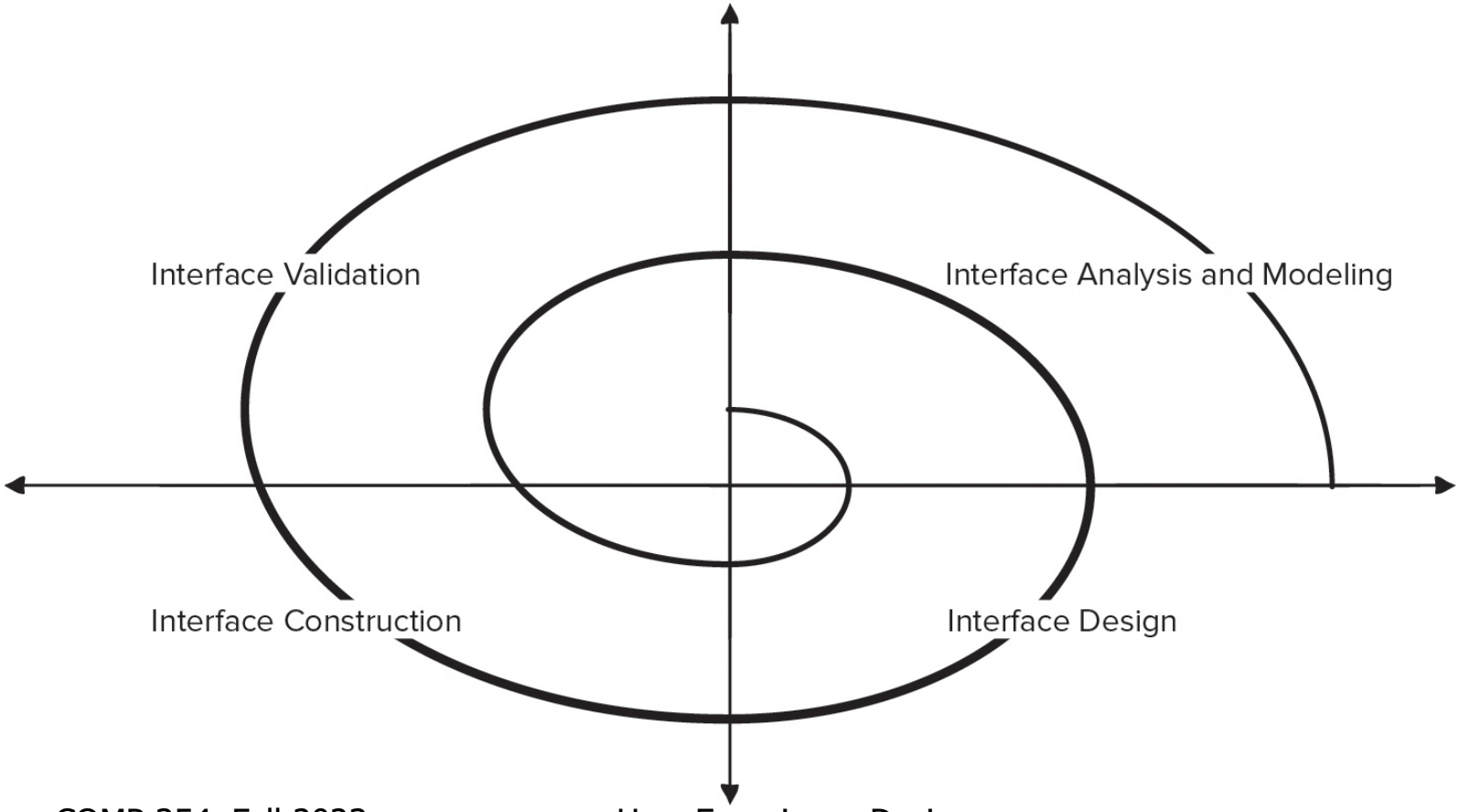


- Allow the user to put the current task into a meaningful context.
- Maintain consistency across a family of applications.
- If past interactive models have created user expectations, do not make changes unless there is a compelling reason to do so.

User Interface Design Models

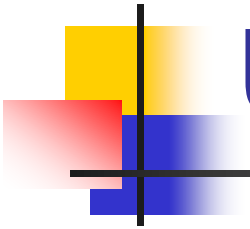
- User model — a profile of all end users of the system.
- Design model — a design realization of the user model.
- Mental model (system perception) — the user's mental image of what the interface is.
- Implementation model — the interface "look and feel" coupled with supporting information that describe interface syntax and semantics.
- An interface designer needs to reconcile these models and derive a consistent representation of the interface.

User Interface Design Process



User Interface Analysis and Design

- Interface analysis focuses on the profile of the users who will interact with the system.
- Interface design defines a set of interface objects and actions that enable a user to perform all defined tasks in a manner that meets every usability goal defined for the system.
- Interface construction normally begins with the creation of a prototype that enables usage scenarios to be evaluated.
- Interface validation focuses on:
  1. The ability of the interface to implement every user task correctly.
  2. The degree to which interface is easy to use and easy to learn.
  3. The user's acceptance of the interface as a tool in her work.



User Experience Analysis

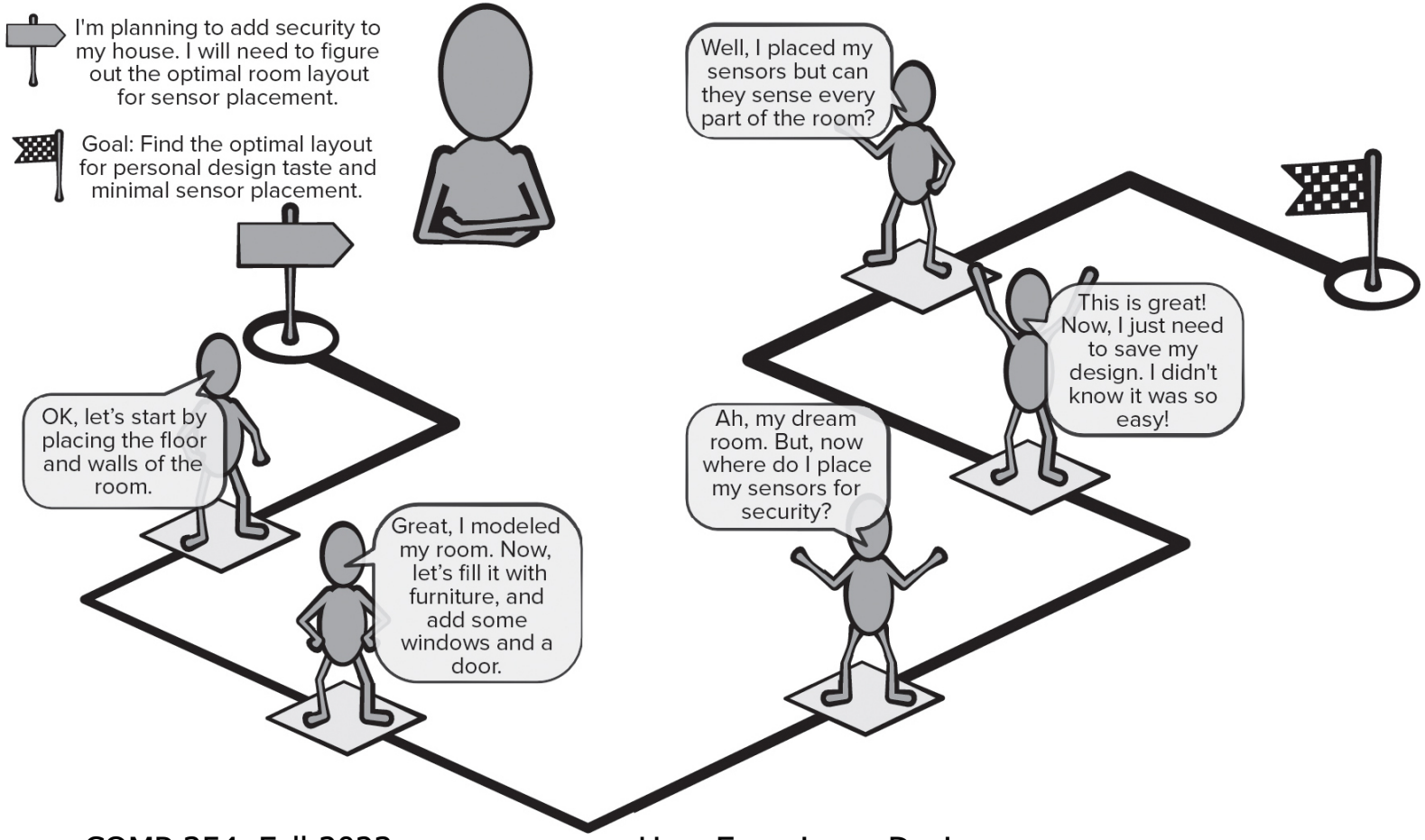
In the case of user experience design, understanding the problem means understanding:

1. the people (end users) who will interact with the system through the interface.
2. the tasks that end users must perform to do their work.
3. the content that is presented as part of the interface.
4. the environment in which these tasks will be conducted.

Using Customer Journey Map

1. Gather stakeholders.
2. Conduct research. Collect all information you can about all the things users may experience as they use the software product and define your customer phases (touchpoints).
3. Build the model. Create a visualization of the touch points.
4. Refine the design. Recruit a designer to make the deliverable visually appealing and ensure touchpoints are identified clearly.
5. Identify gaps. Note any gaps in the customer experience or points of friction or pain (poor transition between phases).
6. Implement your findings. Assign responsible parties to bridge the gaps and resolve pain points found.

Customer Journey Map

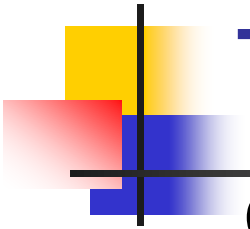


Creating and Using Personas in UX Design

- Data collection and analysis. Stakeholders collect information about proposed product users and determine the user group needs.
- Describe personas. The developers need to decide how many personas to create and decide which persona will be their focus.
- Develop scenarios. Scenarios are user stories about how personas will use the product being developed. They may focus on the touchpoints and obstacles described in the customer journey.
- Acceptance by stakeholders. Often this is done by validating the scenarios using a review technique or demonstration called cognitive walkthrough (stakeholders assume the role defined by a persona and work through a scenario using a system prototype).

Persona Example

 Elizabeth	Works as an elementary teacher in a small midwestern city. Is 38 years old and holds a masters in elementary education. Prefers open-design concepts and shabby chic interior design.	Used to working with computers, but has little experience with virtual reality and tends to get motion sickness. Wants to renovate her house with her design preferences and added security features, but needs help visualizing layouts and lines of sight.

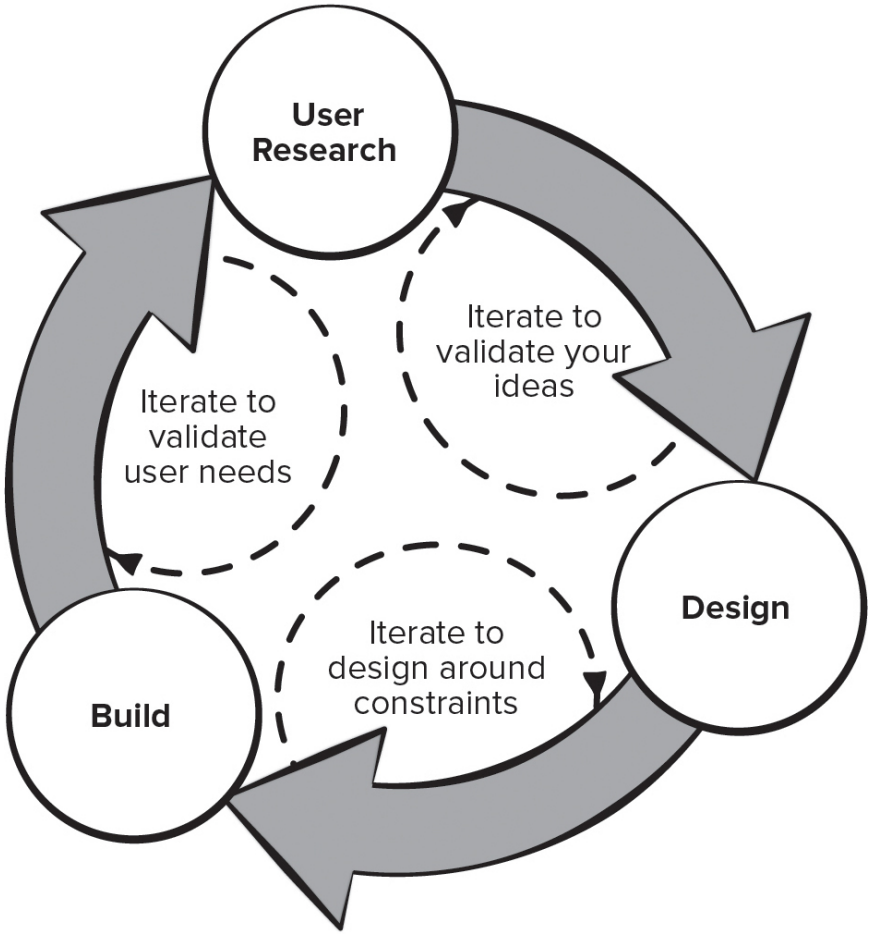


Task Analysis

Goal of task (scenario) analysis is to answer the following questions:

- What work will the user perform in specific circumstances?
- What tasks and subtasks will be performed as the user does the work?
- What specific problem domain objects will the user manipulate as work is performed?
- What is the sequence of work tasks—the workflow?
- What is the hierarchy of tasks?

Iterative UX Design Process



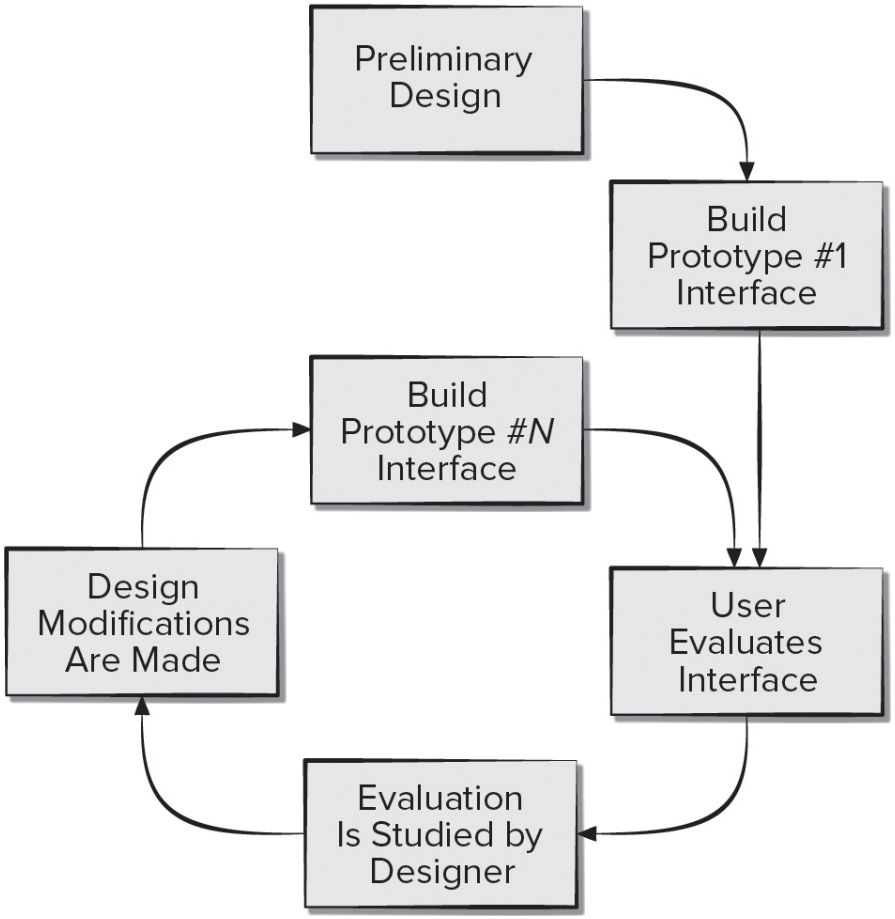
Google 5-Day UX Design Sprint

- Understand. User research activities (user needs and business goals) for the software product. This information is posted on whiteboards (For example, customer journey maps, personas, user task workflow) for easy reference throughout the sprint.
- Sketch. Individual stakeholders are given the time and space needed to brainstorm solutions to the problems discovered in the understand phase. Paper drawings and notes are easy to generate, easy to modify, and quite inexpensive.
- Decide. Each stakeholder presents his solution sketch and the team votes to determine the solutions that should be tackled in the prototyping activities that will follow. If there is not a clear consensus following the voting, the development team may decide to consider assumptions that involve project constraints and resources.

Google
5-Day UX Design Sprint

- Prototype. May be a minimally viable product based on the solution selected from the sketch phase, or it may be based on the portions of the customer journey map or storyboard you want to evaluate with potential users in the validate phase. This means the team should be developing test cases based on the user stories as the prototype is being built.
- Validate. Every developer watching users try out the prototype this is the best way to discover major issues with its UX design, which in turn lets you start iterating immediately. This is critical to capturing potential learning opportunities by exposing product decision makers to user feedback in real time.

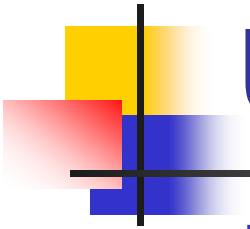
Interface Design Evaluation Cycle



User Interface Design Evaluation Criteria

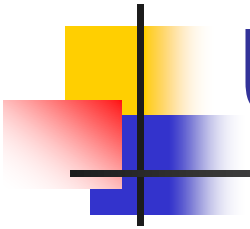
The design model (user stories, storyboard, personas, etc.) of the interface can be evaluated during early design reviews:

1. Length and complexity of the requirements model or written specification of the system and its interface provide an indication of the amount of learning required by users of the system.
2. Number of user tasks specified and the average number of actions per task provide an indication of interaction time and the overall efficiency of the system.
3. Number of actions, tasks, and system states indicated by the design model imply the memory load on users of the system.
4. Interface style, help facilities, and error-handling protocol provide a general indication of the complexity of the interface and the degree to which it will be accepted by the user.



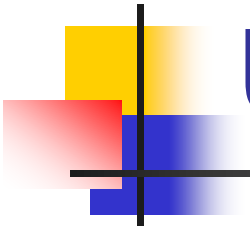
Usability Guidelines

- Anticipation. An application should be designed so that it anticipates the user's next move.
- Communication. The interface should communicate the status of any activity initiated by the user.
- Consistency. The use of navigation controls, menus, icons, and aesthetics (For example, color, shape, layout) should be consistent throughout.
- Controlled Autonomy. The interface should facilitate user movement throughout the application, but it should do so in a manner that enforces navigation conventions that have been established for the application.
- Efficiency. The design of the application and its interface should optimize the user's work efficiency.



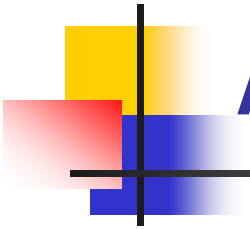
Usability Guidelines

- Flexibility. The interface should be flexible enough to enable some users to accomplish tasks directly and others to explore the application in a somewhat random fashion.
- Focus. The interface (and the content it presents) should stay focused on the user task(s) at hand.
- Human Interface Objects. A vast library of reusable human interface objects has been developed for both Web and mobile apps. Use them.
- Latency Reduction. Rather than making the user wait for some internal operation to complete (for example, downloading a complex graphical image), the application should use multitasking in a way that lets the user proceed with work as if the operation has been completed.



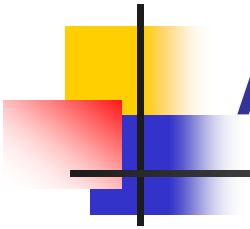
Usability Guidelines

- Learnability. An application interface should be designed to minimize learning time and, once learned, to minimize relearning required when the app is revisited.
- Metaphors. An interface that uses an interaction metaphor is easier to learn and easier to use, as long as the metaphor is appropriate for the application and the user.
- Readability. All information presented through the interface should be readable by young and old.
- Track State. When appropriate, the state of the user interaction should be tracked and stored so that a user can log off and return later to pick up where he left off.
- Visible Navigation. A well-designed interface provides the illusion that users are in the same place, with the work brought to them.



Accessibility Guidelines

- Application Accessibility. Software engineers must ensure that interface design encompasses mechanisms that enable easy for people with special needs.
- Response Time. System response time has two important characteristics: length and variability. Aim for consistency to avoid user frustration.
- Help Facilities. Modern software should provide online help facilities that enable a user to get a question answered or resolve a problem without leaving the interface.
- Error Handling. Every error message or warning produced by an interactive system should: use user understandable jargon, provide constructive error recovery advice, identify negative consequences of errors, contain an audible or visual cue, and never blame user for causing the error.



Accessibility Guidelines

- Menu and Command Labeling. The use of window-oriented, point-and-pick interfaces has reduced reliance on typed commands. However, it is important to: ensure every menu option has a command version, make commands easy for users to type, make commands easy to remember, allow for command abbreviation, make sure menu labels are self-explanatory, make sure submenus match style of master menu items, and ensure command conventions work across the family of applications.
- Internationalization. Software engineers and their managers invariably underestimate the effort and skills required to create user interfaces that accommodate the needs of different locales and languages.