UNIT-5
User Interface Design

The Golden Rules

User Interface Analysis and Design

- Interface Analysis and Design Models
  1) User Model
  2) Design Model
  3) User’s Mental Model
  4) Implementation Model
- The Process
  1) Interface analysis and modeling,
  2) Interface design
  3) Interface construction
  4) Interface validation.

Interface Analysis

1) User Analysis
2) Task Analysis and Modeling
3) Analysis of Display Content
4) Analysis of the Work Environment

Interface Design Steps

WebApp Interface Design

Design Evaluation

THE GOLDEN RULES

Three golden rules:

1. Place the user in control.
2. Reduce the user's memory load.
3. Make the interface consistent.

1. **Defines a number of design:** Principles that allow the user to maintain control:

Define interaction modes in a way that does not force a user into unnecessary or undesired actions: An interaction mode is the current state of the inter-face. For example, if spell check is selected in a word-processor menu.

Provide for flexible interaction. For example, software might allow a user to interact via keyboard commands, mouse movement, a digitizer pen.

Allow user interaction to be interruptible and undoable.

Streamline[classified by individual class]interaction as skill levels advance and allow the interaction to be customized.

Hide technical internals from the casual user.

Design for direct interaction with objects that appear on the screen.

2. **Reduce the user's memory load:** Defines design principles that enable an interface to reduce the user's memory load:

Reduce demand on short-term memory.

Establish meaningful defaults.

Define shortcuts that are intuitive[complex and long].

The visual layout of the interface should be based on a real-world metaphor[situation].

Disclose information in a progressive fashion.

3. **Make the Interface Consistent**

Defines a set of design principles that help make the 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 ANALYSIS AND DESIGN

Interface Analysis and Design Models:

There are **FOUR** different models:

1) User Model
2) Design Model
3) User's Mental Model
4) Implementation Model

1) User Model:

Establishes the profile of end users of the system

Users can be categorized as:

I. **Novices.[Beginners]** No syntactic knowledge of the system and little semantic knowledge of the application or computer usage in general.

II. **Knowledgeable, intermittent users.** Reasonable semantic knowledge of the application but relatively low recall of syntactic information necessary to use the interface.

III. **Knowledgeable, frequent users.** Good semantic and syntactic knowledge that of-ten leads to the "power-user syndrome"; that is, individuals who look for short-cuts and abbreviated modes of interaction.

2) Design Model: the software engineer creates a design model, using different models.

3) User's Mental Model:

It is the image of the system that end users carry in their heads. For example, if the user of a particular word processor were asked to describe its operation, the system perception would guide the response.

4) Implementation Model: Know the user, know the tasks."

Combines the outward manifestation of the computer-based system (the look and feel of the interface), coupled with all supporting information (books, manuals, videotapes, help files) that describes interface syntax and semantics.

The Process: The analysis and design process for user interfaces is iterative and can be represented using a spiral model.

The user interface analysis and design process encompasses:

(1) interface analysis and modeling,

(2) interface design

(3) interface construction

(4) interface validation.

(1) Interface analysis:

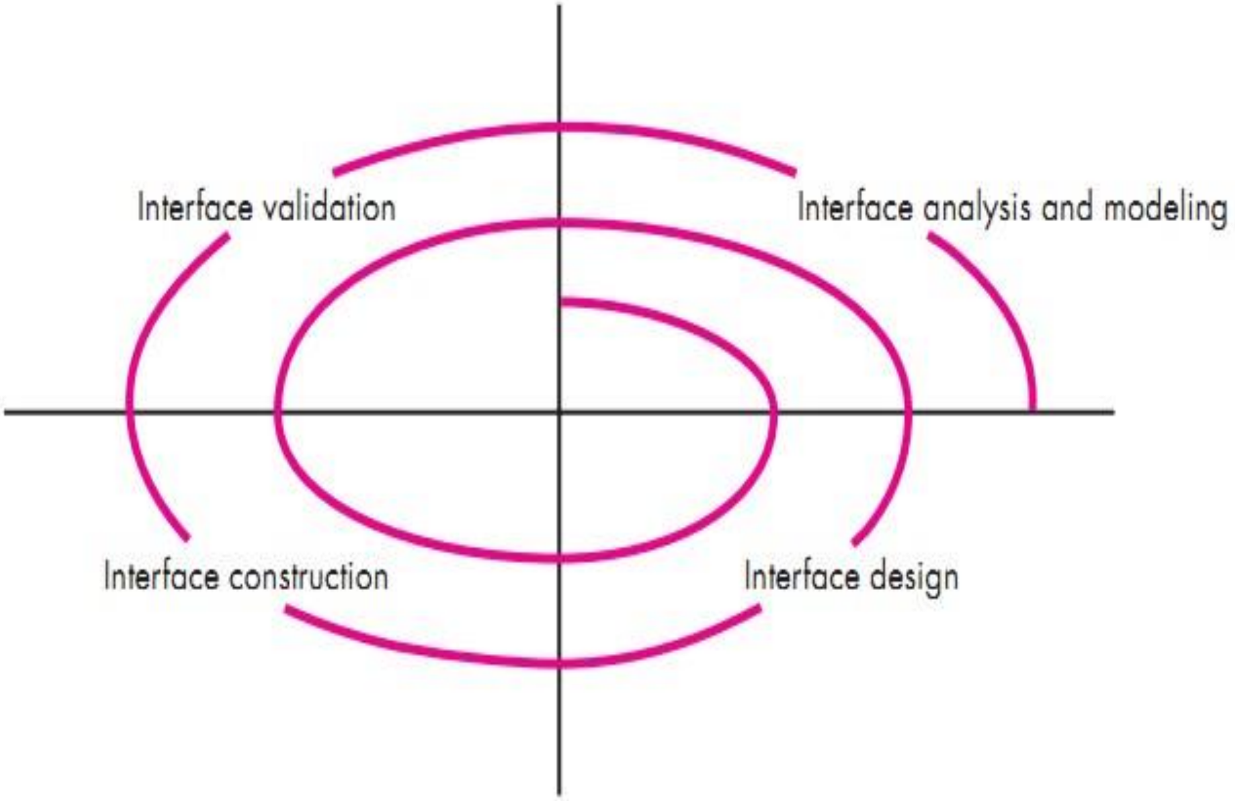
focuses on the profile of the users who will interact with the system. Skill level, business understanding, and general receptiveness to the new system are recorded; and different user categories are defined.

The information gathered as part of the analysis action is used to create an analysis model for the interface.

(2) Interface design: is to define a set of interface objects and actions (and their screen representations) that enable a user to perform all defined tasks

(3) Interface construction: begins with the creation of a prototype that enables usage scenarios to be evaluated.

(4) Interface validation: focuses on (a) the ability of the interface to implement every user task correctly, (b) the degree to which the interface is easy to use and easy to learn, and (c) the users' acceptance of the interface as a useful tool in their work.



[Figure of-Spiral Process Model]

INTERFACE ANALYSIS

A) User Analysis:

Information related to user interface from a broad array of sources can be used to accomplish this:

- User Interviews
- Sales input
- Marketing input
- Support input

To understand the users of a system, questions are analyzed :

- Are users trained professionals, technicians, clerical, or manufacturing workers?
- What level of formal education does the average user have?
- interface analysis
- Are the users capable of learning from written materials or have they expressed a desire for classroom training?
- Are users expert typists or keyboard phobic?

- What is the age range of the user community?

B) Task Analysis and Modeling:

I) Use cases: When used as part of task analysis, the use case is developed **to show how an end user performs some specific work-related task.** From it, you can extract tasks, objects, and the overall flow of the interaction.

II) Task elaboration: Stepwise elaboration (also called stepwise refinement) as **a mechanism for refining the processing tasks** that are required for software to accomplish some desired function.

The **design model** of the interface should accommodate each of these tasks in a way that is consistent with the **user model** (the profile of a "typical" interior designer) and **system perception** (what the interior designer expects from an automated system).

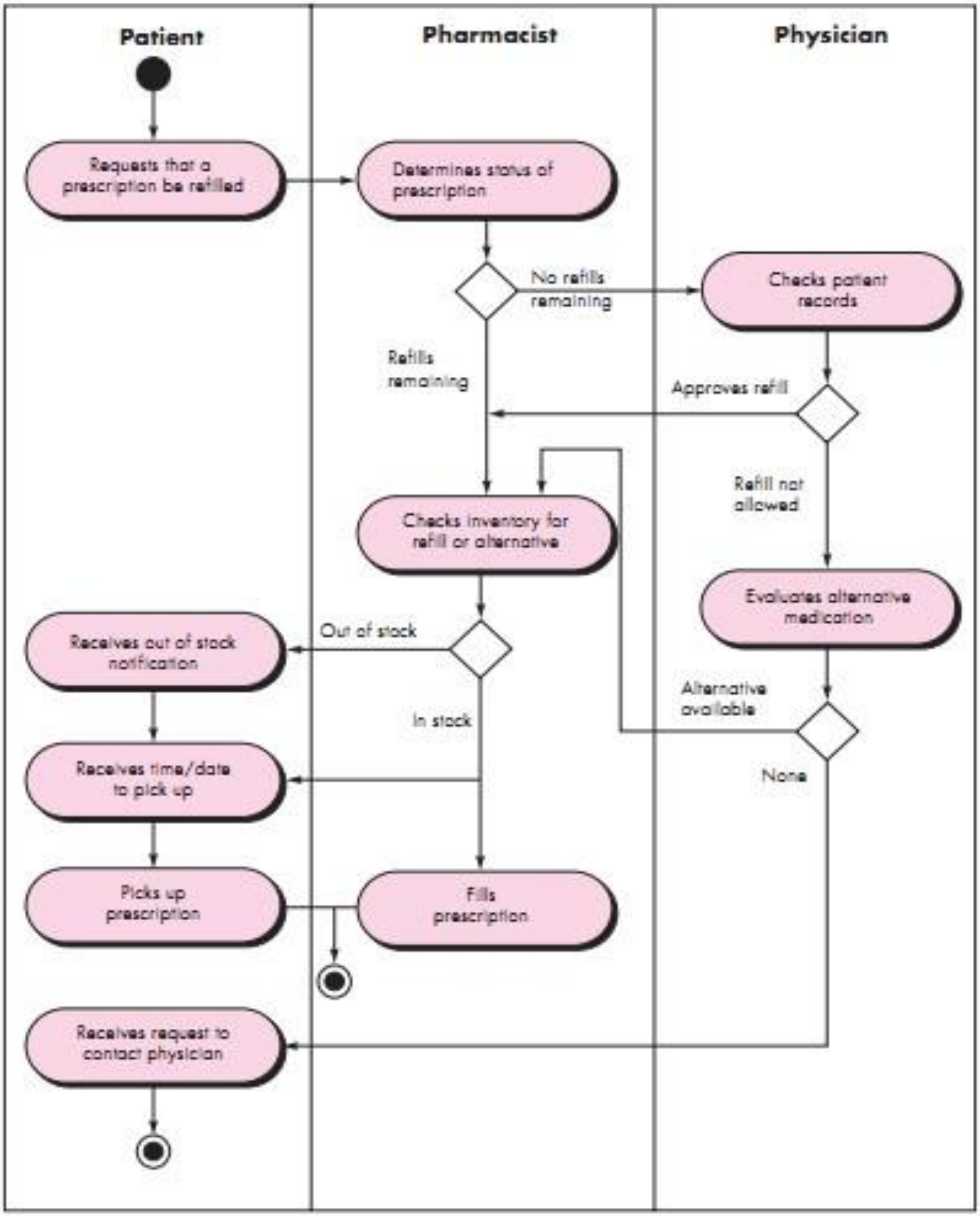
III) Object elaboration: Examine the use case and other information obtained from the user and extract the objects that are used by the interior designer. These **objects** can be categorized into **classes. Attributes** of each class are defined, and an evaluation of the actions applied to each object provide a list of operations.

IV) Workflow analysis: When a number of different users, each playing different roles, makes use of a user interface, task analysis and object elaboration and apply workflow analysis.

This technique allows you to understand how a work process is completed when several people (and roles) are involved.

Workflow can be represented effectively with a UML swimlane diagram (a variation on the activity diagram).

Example


Patient
Pharmacist
Physician
Requests that a prescription be refilled
Determines status of prescription
No refills remaining
Checks patient records
Refills remaining
Approves refill
Refill not allowed
Checks inventory for refill or alternative
Evaluates alternative medication
Out of stock
Receives out of stock notification
Alternative available
In stock
None
Receives time/date to pick up
Picks up prescription
Fills prescription
Receives request to contact physician

[FIGURE OF SWIMLANE DIAGRAM FOR PRESCRIPTION REFILL FUNCTION]

V) Hierarchical representation: a task hierarchy can be defined for each user type. The hierarchy is derived by a stepwise elaboration of each task identified for the user. To complete the task, three subtasks are defined. One of these subtasks, provide identifying information, is further elaborated in three additional sub-subtasks.

C) Analysis of Display Content

During this interface analysis step, the format and aesthetics of the content (as it is displayed by the interface) are considered.

Among the questions that are asked and answered are:

- Are different types of data assigned to consistent geographic locations on the screen (e.g., photos always appear in the upper right-hand corner)?
- Is proper on-screen identification assigned to all content?
- If a large report is to be presented, how should it be partitioned for ease of understanding?

The answers to these (and other) questions will help you to establish requirements for content presentation.

D) Analysis of the Work Environment

In some applications the user interface for a computer-based system is placed in a “user-friendly location” (e.g., proper lighting, good display height, easy keyboard access).

The interface designer may be constrained by factors ease of use

INTERFACE DESIGN STEPS

A) Applying Interface Design Steps:

Create Use case

Defined and elaborated objects and actions & categorized by their type.

Create screen layout

Perform interface design activities.

B) User Interface Design Patterns:

A design pattern is an abstraction that prescribes a design solution to a specific, well-bounded design problem.

E.g. calendar tool

C) Design Issues:

Response time: System response time is the primary complaint for many interactive applications. In general, system response time is measured from the point at which the user performs some control action (e.g., hits the return key or clicks a mouse) until the software responds with desired output or action

Help facilities: detailed research in a multivolume set of “user manuals” may be the only option. In most cases, however, modern software provides online help facilities that enable a user to get a question answered or resolve a problem without leaving the interface.

Error handling: Error messages and warnings are "bad news" delivered to users of interactive systems when something has gone awry. At their worst, error messages and warnings impart useless or misleading information and serve only to increase user frustration.

Menu and command labeling: The typed command was once the most common mode of interaction between user and system software and was commonly used for applications of every type. Today, the use of window-oriented, point-and-pick interfaces has reduced reliance on typed commands, but some power-users continue to prefer a command-oriented mode of interaction.

Application accessibility: As computing applications become ubiquitous, software engineers must ensure that interface design encompasses mechanisms that enable easy access for those with special needs. Accessibility for users (and software engineers) who may be physically challenged is an imperative for ethical, legal, and business reasons

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. The challenge for interface designers is to create "globalized" software. Localization features enable the interface to be customized for a specific market.

WEB APP INTERFACE DESIGN

Interface Design Principles and Guidelines:

In order to design WebApp interfaces that exhibit these characteristics identifies a set of overriding design principles:

1. Anticipation[expectation]: A WebApp should be designed so that it anticipates the user's next move.

2. Communication: The interface should communicate the status of any activity initiated by the user.

3. Consistency: The use of navigation controls, menus, icons, and aesthetics (e.g., color, shape, layout) should be consistent throughout the WebApp.

4. Controlled autonomy: The interface should facilitate user movement throughout the WebApp, but it should do so in a manner that enforces navigation conventions that have been established for the application.

5. Efficiency: The design of the WebApp and its interface should optimize the user's work efficiency, not the efficiency of the developer who designs and builds it or the client-server environment that executes it

6. Flexibility: The interface should be flexible enough to enable some users to accomplish tasks directly and others to explore the WebApp in a somewhat random fashion.

7. Focus: The WebApp interface (and the content it presents) should stay focused on the user task(s) at hand

8. Fitt's law: "The time to acquire a target is a function of the distance to and size of the target"

9. Human interface objects: A vast library of reusable human interface objects has been developed for WebApps. Use them. Any interface object that can be "seen, heard, touched or otherwise perceived"

10. Latency reduction: Rather than making the user wait for some internal operation to complete (e.g., downloading a complex graphical image), the WebApp should use multitasking in a way that lets the user proceed with work as if the operation has been completed

11. Learnability: A WebApp interface should be designed to minimize learning time, and once learned, to minimize relearning required when the WebApp is revisited

12. 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

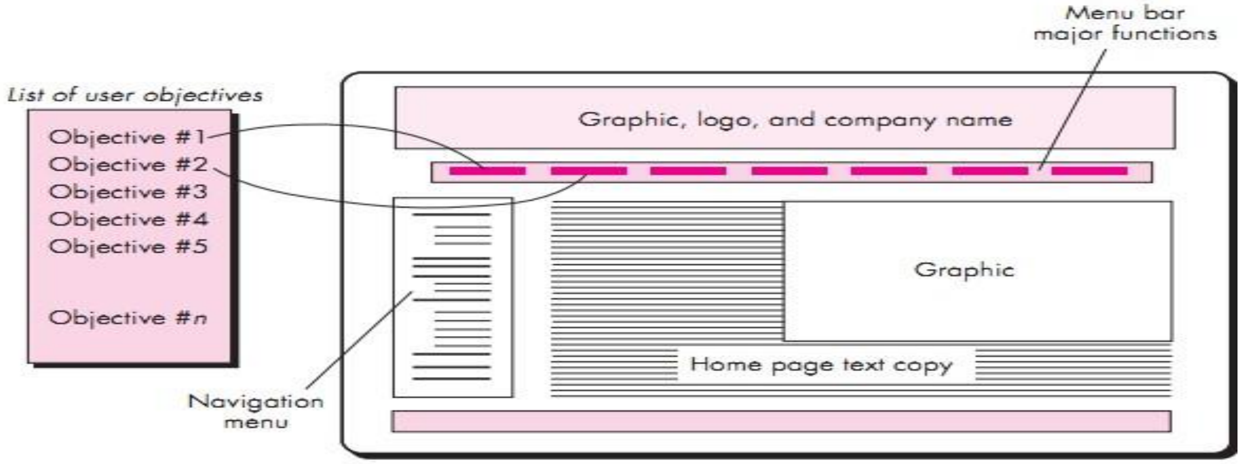
13. Maintain work product integrity: A work product (e.g., a form completed by the user, a user-specified list) must be automatically saved so that it will not be lost if an error occurs.

14. Readability: All information presented through the interface should be readable by young and old.

15. Track state: When appropriate, the state of the user interaction should be tracked and stored so that a user can logoff and return later to pick up where she left off

16. Visible navigation: A well-designed WebApp interface provides "the illusion that users are in the same place, with the work brought to them"

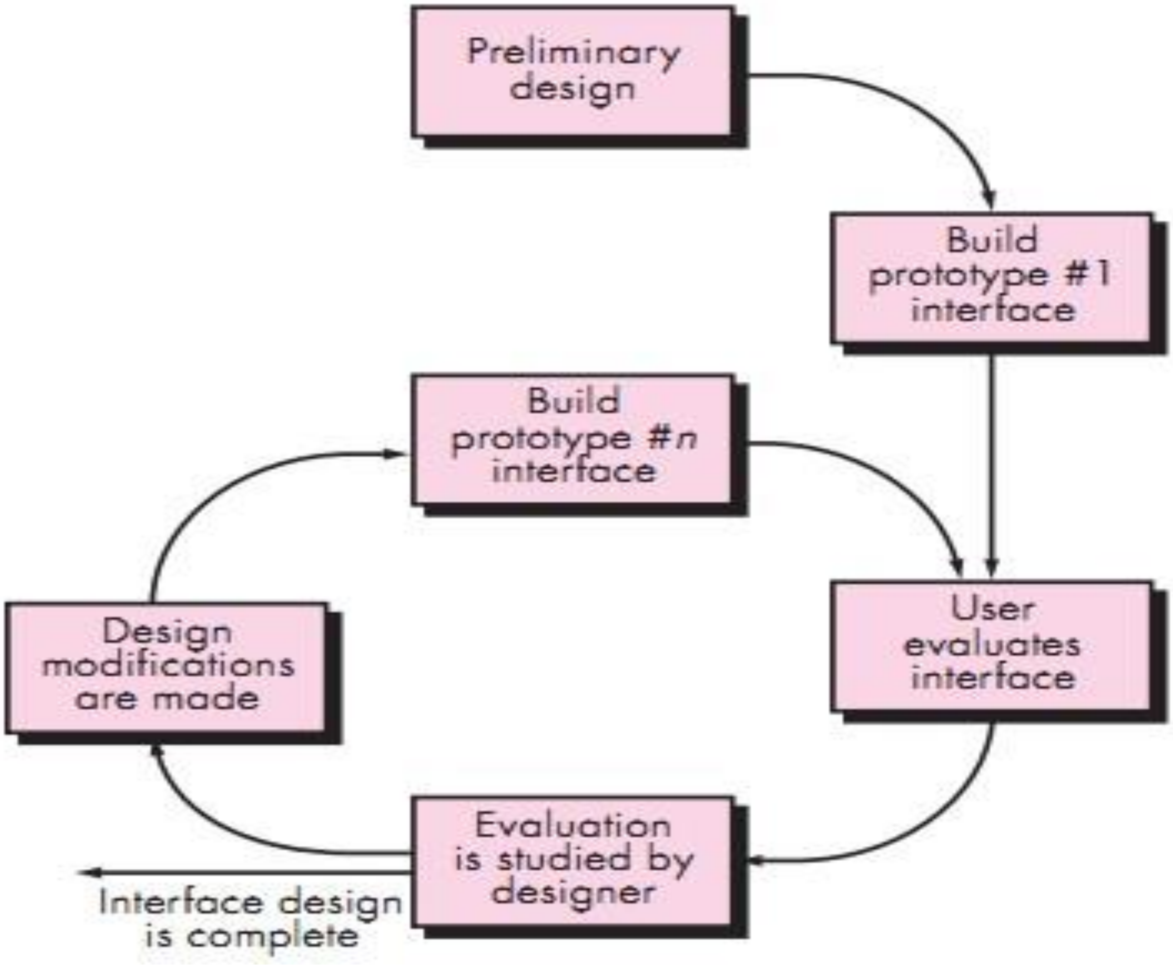
Interface Design Workflow for WebApps :



1. Review information contained in the requirements model and refine as required.
2. Develop a rough sketch of the WebApp interface layout.
3. Map user objectives into specific interface actions.
4. Define a set of user tasks that are associated with each action.
5. Storyboard screen images for each interface action.
6. Refine interface layout and storyboards using input from aesthetic design.
7. Identify user interface objects that are required to implement the interface.
8. Develop a procedural representation of the user's interaction with the interface.
9. Develop a behavioral representation of the interface.
10. Describe the interface layout for each state.
11. Refine and review the interface design model.

DESIGN EVALUATION

After creation of UI, it must be evaluated, whether it meets user needs or not.


Preliminary design
Build prototype #1 interface
Build prototype #n interface
User evaluates interface
Design modifications are made
Evaluation is studied by designer
Interface design is complete

[Figure-Interface design evaluation cycle]

Evaluation Criteria:

1. The **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. The 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. The 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.**