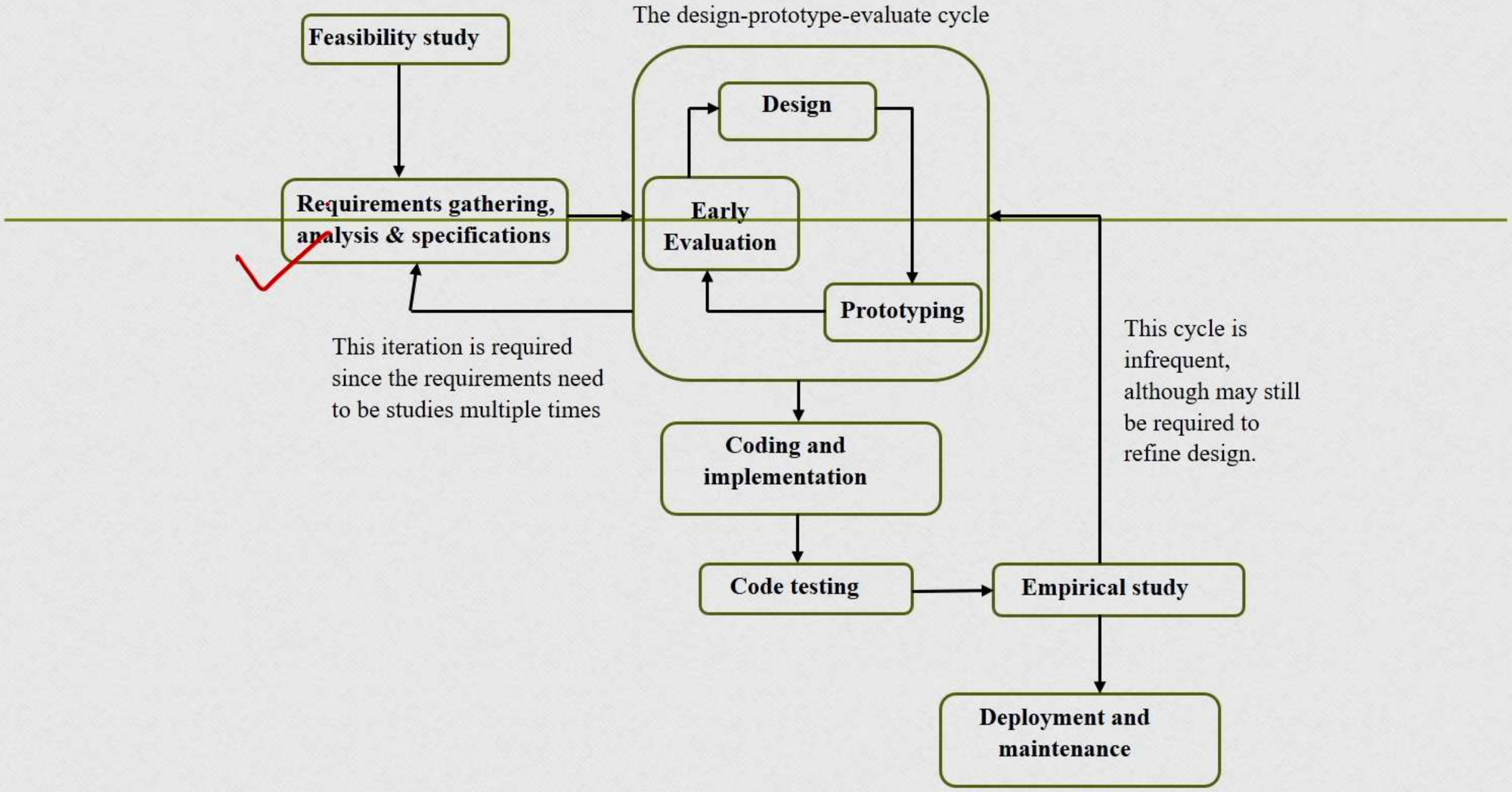
Recap

- Last lecture we learned requirement gathering through CI and design with guidelines
- This lecture we shall learn prototyping techniques
- First let us recall SDLC for UCD


The design-prototype-evaluate cycle
Feasibility study
Requirements gathering, analysis & specifications
Design
Early Evaluation
Prototyping
This iteration is required since the requirements need to be studies multiple times
Coding and implementation
Code testing
Empirical study
This cycle is infrequent, although may still be required to refine design.
Deployment and maintenance

Prototyping for Feedback

- Instead, we go for a prototype, a lightweight implementation in a sense, to test our idea

Prototyping for Feedback

- Once its purpose is served, a prototype can be discarded

Prototyping for Feedback

- In the best case, the prototype can be **incrementally refined** (and **tested**) as we make progress in the development till the end, at which point it becomes the **fully-implemented product** in itself

Prototype Categories

- **Horizontal** - the entire interface is depicted at the surface level without any functionality
  - Interaction is not prototyped
  - No real work can be done
  - Suitable to discuss, brain-storm or elicit feedback on the interface look and feel, primarily

Prototype Categories

- **Vertical** - designed to represent interaction
  - Few selected features implemented in-depth, starting from the first screen to the screen after the last action is performed
  - Suitable for analysis of interactions and features

How to Create Prototypes

- **Low fidelity**
  - No technological intervention - made with paper, cardboards, woods and so on
  - A popular example are the paper mock-ups for interface look, feel and even functionality
  - Quick and cheap to make and modify
  - Good for horizontal prototypes and used to brain-storm on alternative designs and get user response on the idea

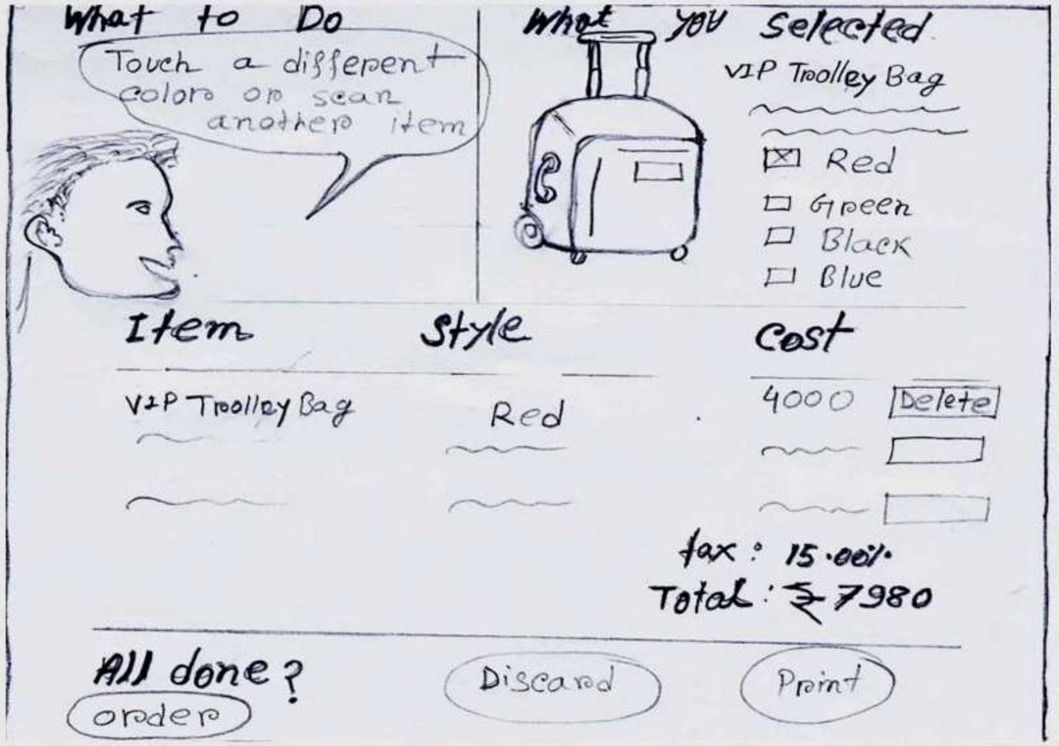
Low Fidelity Prototype: Interface Sketches

- Interface sketches offer another low-fidelity prototypes
  - Drawings depicting major components of an interface
  - Provide way to envision appearance of the interface

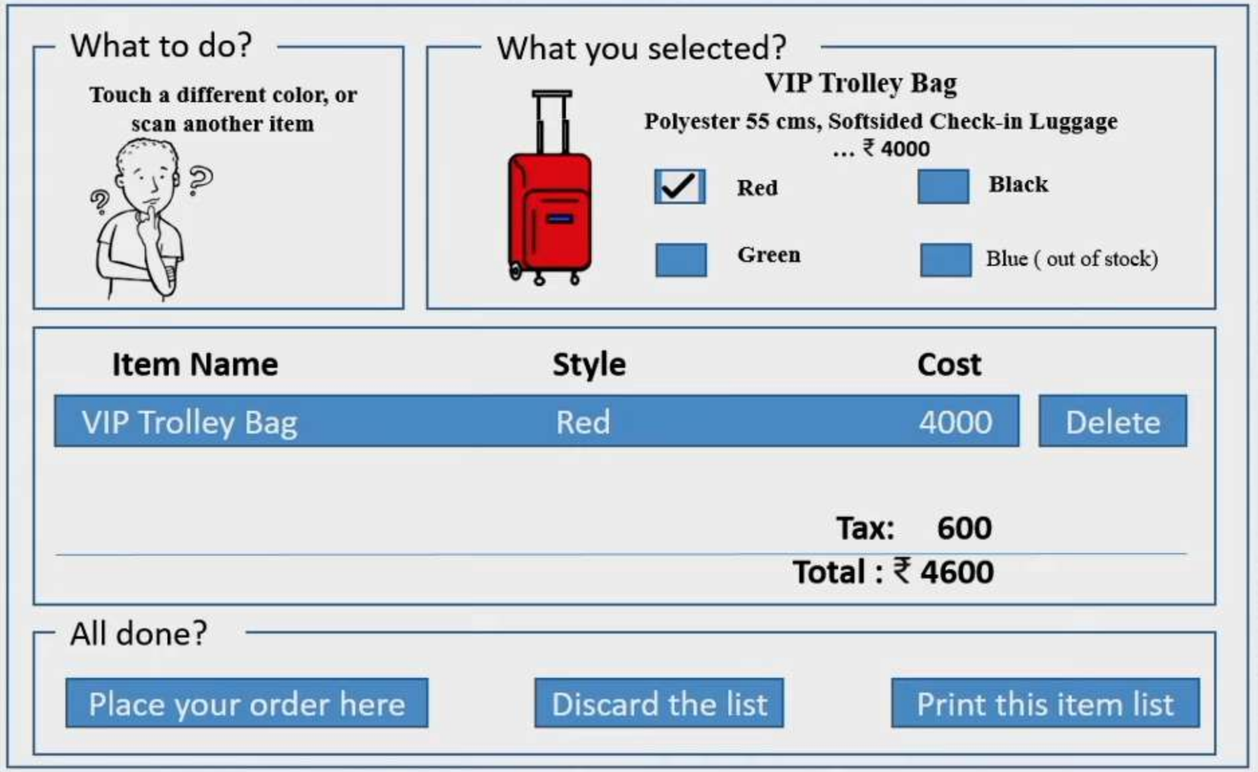
Low Fidelity Prototype: Interface Sketches

Example - consider a shop that sells various consumer items (clothes, bags and so on). They want an interactive system to generate bills based on the selection by a customer. The customer selection is “scanned” and identified by the system and a bill is generated. A sketch for the proposed interface design is shown along with one possible final design.

Low Fidelity Prototype: Interface Sketches



Low Fidelity Prototype: Interface Sketches



Low Fidelity Prototype: Interface Sketches

- What we need is **not a single sketch**, but a **series of sketches**, often called "storyboarding" to represent interaction

Low Fidelity Prototype: Storyboarding

- The idea originated from the film industry, where it is used to depict a scene

Low Fidelity Prototype: Storyboarding

- Each sketch in a storyboard represents a “key frame” (the terminology again borrowed from the movies), which are “snapshots” of the interaction at a particular point of time

Low Fidelity Prototype: Storyboarding

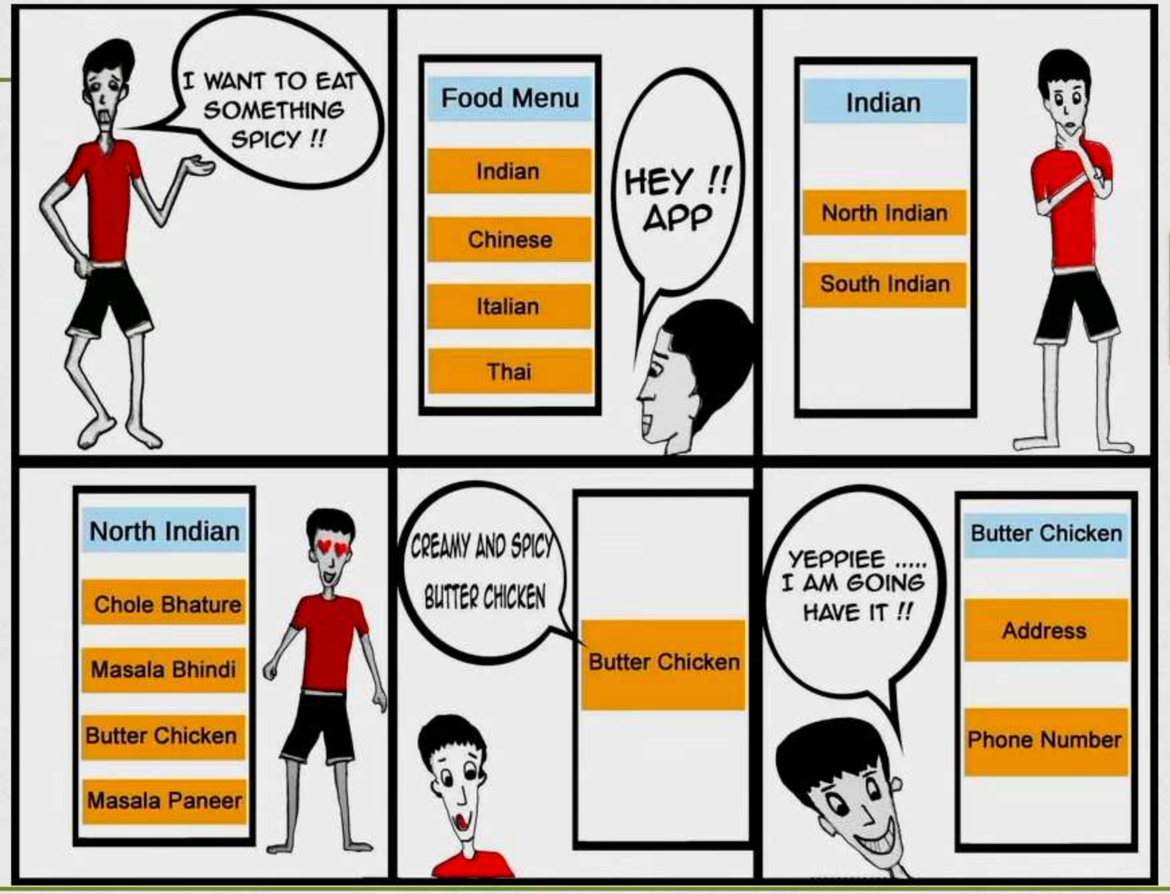
- The key frames allow us to visualize the nature of interaction in its entirety, albeit for a specific scene

Low Fidelity Prototype: Storyboarding

- Example - a proposed mobile app design to find “spicy food” in your neighborhood

Low Fidelity Prototype: Storyboarding

- Storyboard captures sequence of actions starting from the desire to find food to the actual finding of the food, along with the appearance of the intermediate screens



How to Create Prototypes

- **Medium-fidelity** - when we use computer to create simple (low fidelity) prototypes
  - If the computer is used to create static sketches only, **it still remains horizontal prototype** since no real functionality is there

How to Create Prototypes

- Ex - we can have an animation video (created with Adobe Flash tool, for example) to prototype the food selection app of the previous slide
  - In the video, simple controls can be provided to simulate the interaction

Medium-Fidelity Prototyping

- In fact, the interaction can also be simulated with as simple a tool as a Microsoft PowerPoinT™ slideshow
  - The key frames can be converted to slides
  - With simple controls (e.g., timers or key press), slide transition takes place depicting the interaction
  - In that way, the storyboard can be converted to a vertical prototype

How to Create Prototypes

- **Hi-Fidelity Prototypes** – prototypes created with computer programs (e.g., actual software development)
- More sophisticated and requires much more effort (including expertise in programming) than the other two categories

How to Create Prototypes

- There are many interface builders, “toolkits” and “wizards” to ease the programming efforts

How to Create Prototypes

- Popular examples include the Tcl/Tk toolkit, Visual Basic programming language, JAVA Swing library and so on

How to Create Prototypes

- The interaction is implemented through programming

How to Create Prototypes

- Hi-fidelity prototypes are **mainly used to create vertical prototypes**

How to Create Prototypes

- "Wizard of Oz" approach - an interesting prototyping technique

How to Create Prototypes

- In this technique, a human subject *believes to* interact with an *autonomous* computer
- In reality, however, the computer is operated by an "unseen" human being (the "Wizard")
- The term originated in the early 1980s, although the concept was there beforehand

How to Create Prototypes

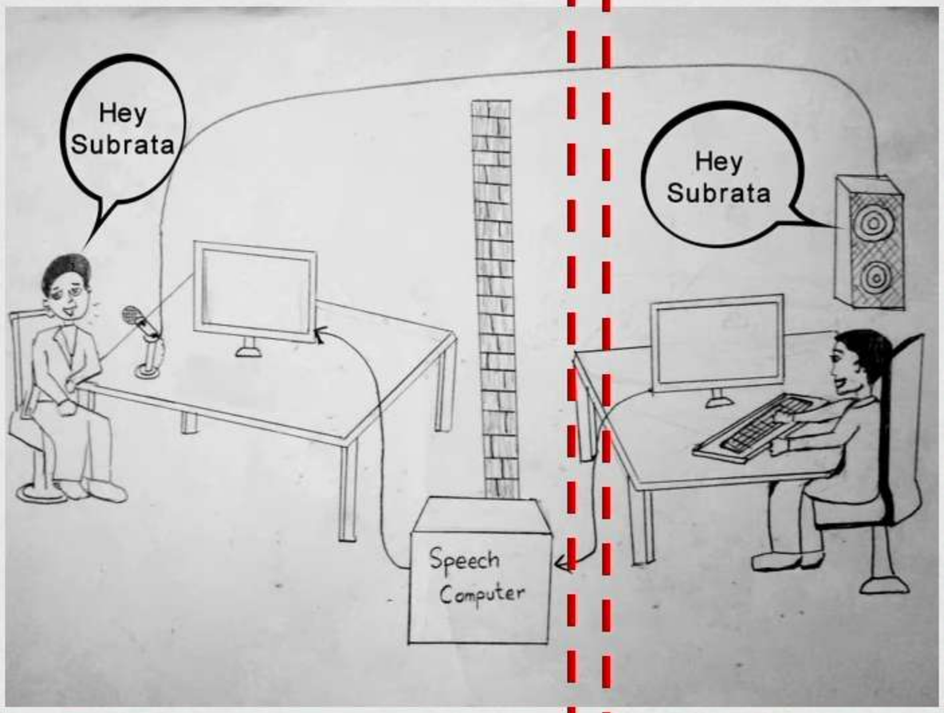
- Example - the "listening typewriter" experiment conducted by the IBM in 1984
  - IBM was trying to develop a speech recognition system
  - They wanted to test the acceptability of the idea, requiring a working system
  - **Problem - nobody had the speech recognizer and the company did not want to spend money and effort before testing the acceptability**

How to Create Prototypes

- So, they decided to go for a “Wizard of Oz” experiment

Wizard of Oz Experiment

What the user sees



The wizard

How to Use Prototypes

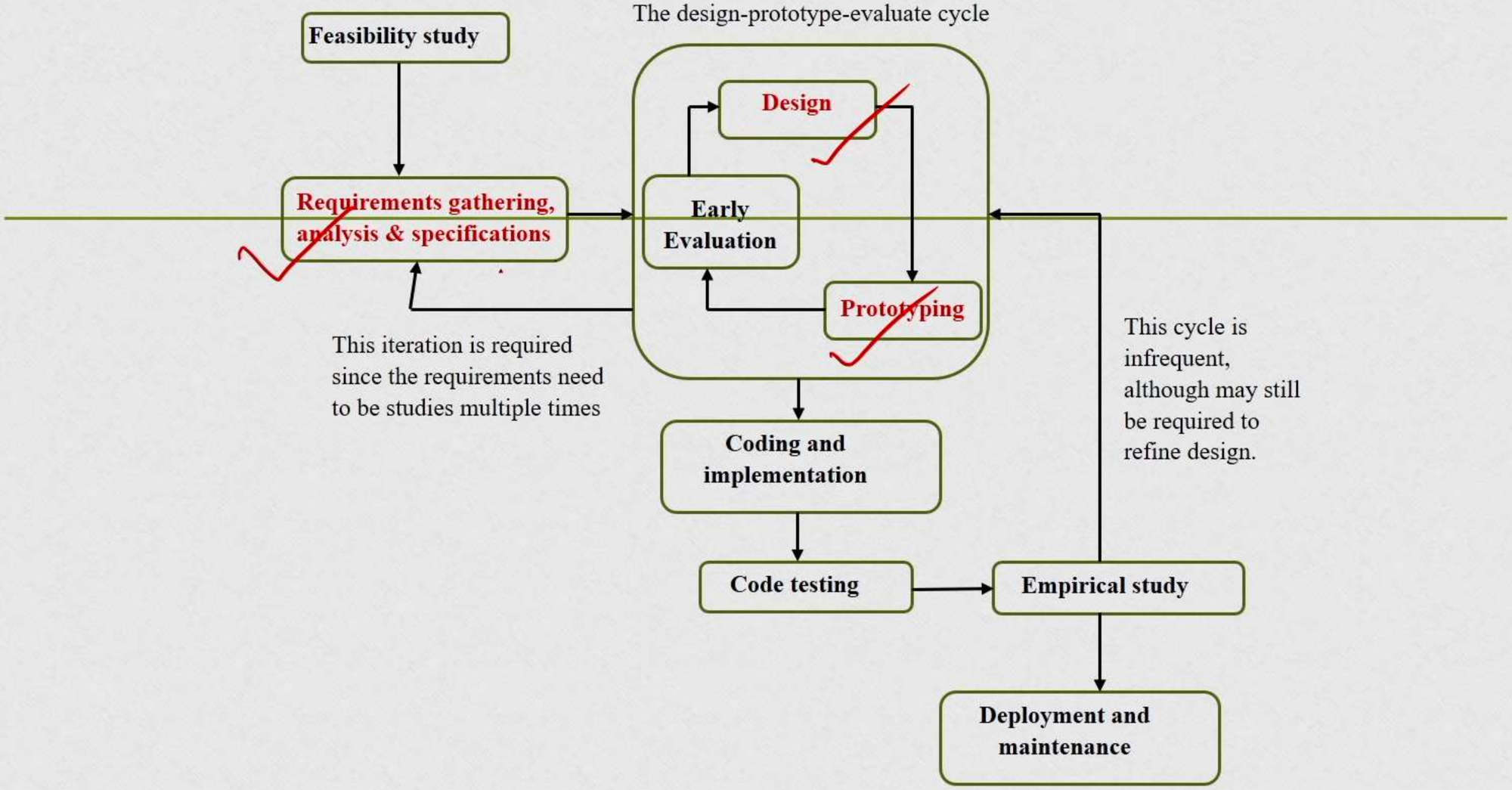
- It can be **“thrown away,”** i.e., discarded after use (Cardboard mock-ups and paper sketches suitable for such treatments, since these are low-cost)
  - We should be able to create such prototypes rapidly (otherwise, it becomes too expensive to discard a prototype made over a period of time with considerable man-hour spent)

How to Use Prototypes

- In the **"incremental"** approach, the system is designed into units (modules)
  - Each unit is separately prototyped and tested
  - Afterwards, it is integrated into the system

How to Use Prototypes

- The other approach is **“evolutionary”** – the whole system is prototyped and tested
  - Based on the testing, the prototype is altered
  - Eventually, it becomes the final product


The design-prototype-evaluate cycle
Feasibility study
Requirements gathering, analysis & specifications
Design
Early Evaluation
Prototyping
This iteration is required since the requirements need to be studies multiple times
Coding and implementation
Code testing
Empirical study
This cycle is infrequent, although may still be required to refine design.
Deployment and maintenance

Book

- **Bhattacharya, S**. (July, 2019). Human-Computer Interaction: User-Centric Computing for Design, McGraw-Hill India
  - Print Edition: ISBN-13: 978-93-5316-804-9; ISBN-10: 93-5316-804-X
  - E-book Edition: ISBN-13: 978-93-5316-805-6; ISBN-10: 93-5316-805-8

Chapter 2, Sec 2.4.5