L3 –Engineering for Usability

Dr Samit Bhattacharya

Computer Science and Engineering

IIT Guwahati

Recap

- What we learned so far
  - General introduction to the field
  - Historical evolution
  - Core concern
  - Idea of usability

This lecture – software development life cycles

Introduction

- User-centric software – core design concern
  - Design usable system (to cater to the needs and expectations of the "layman" users)

What We Need

- A “systematic” approach!

Software Development Life Cycle

- To **comprehensively capture** and **represent** design and development activities

Engineering a Software

- Software development life cycles – **build software in stages**

Engineering a Software

- Example – let us try to create a calendar app

- **What should we do?**

Calendar App - A Possible Design

- Create a grid-like structure typically found on a physical calendar

- Put headings on each cell in the grid (representing the name of the month)

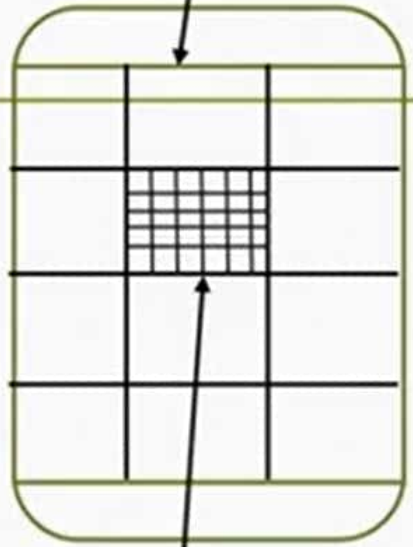
Calendar App - A Possible Design

- Create sub-grids in each cell to hold the dates
- Render the entire structure on the screen

Calendar App - A Possible Design

- Highlight the current date and month

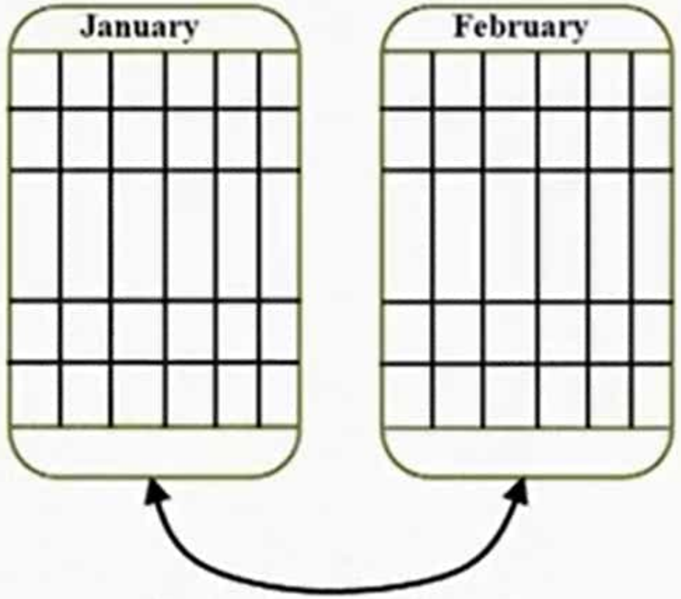
One Possible Design



A 3×4 grid structure. Each cell holds the dates for a month.

Each cell is sub-divided into a 6×6 sub-grid (to hold maximum 31 dates). The current date is highlighted.

Another Possible Design



Each screen displays dates of one month (with the current date highlighted). Screen changes through tap/swipe

Design Alternatives

- There can be many possibilities
  - Challenge - how to chose the right one
- Require a systematic approach
  - SDLCs help

SDLC

- There are many
  - Waterfall model
  - Spiral model
  - Evolutionary model
  - …

Waterfall Model

Waterfall Model

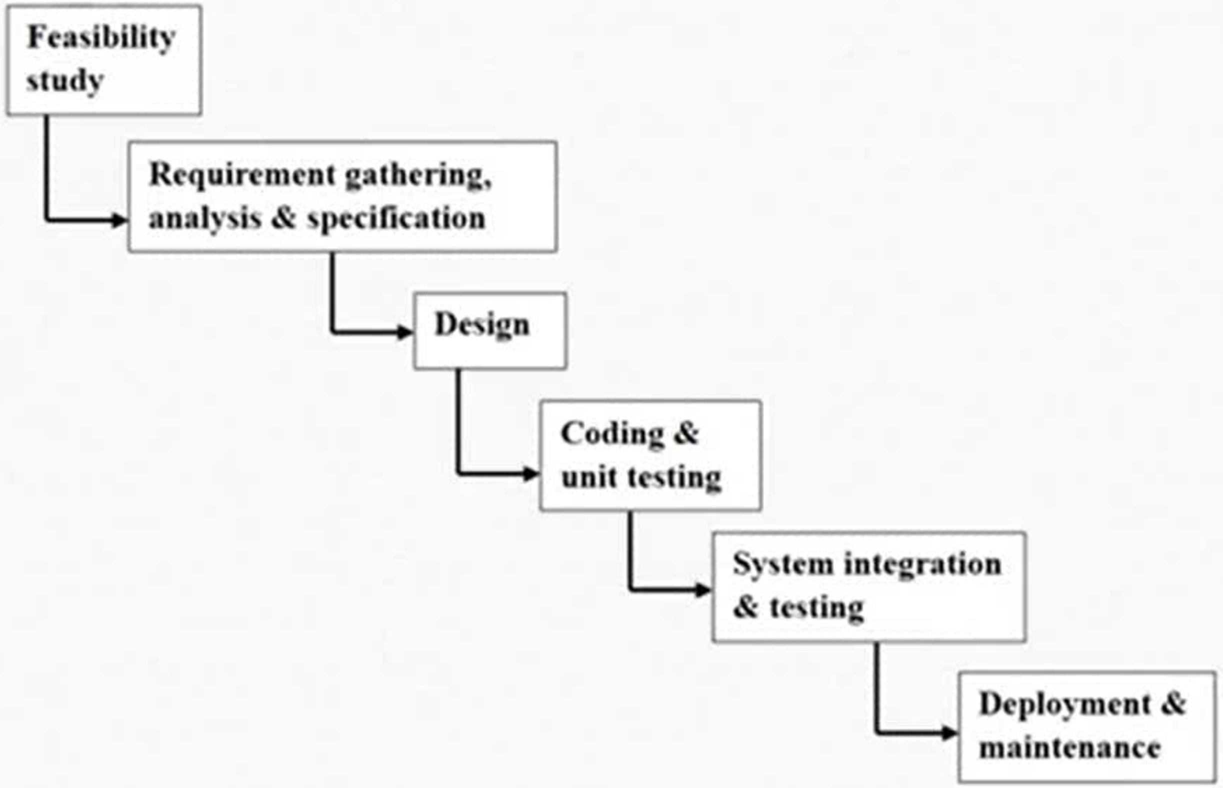
- Most well-known SDLC model (although rarely used in practice)

Waterfall Model (Classical)

- SEVEN major stages
  - Feasibility study
  - Requirement gathering, analysis and specification
  - System design
  - Coding (implementation) and unit testing
  - Integration and system testing
  - Deployment
  - Maintenance

Waterfall Model

- The stages are depicted as a "waterfall" (hence the name)



Feasibility study

Requirement gathering, analysis & specification

Design

Coding & unit testing

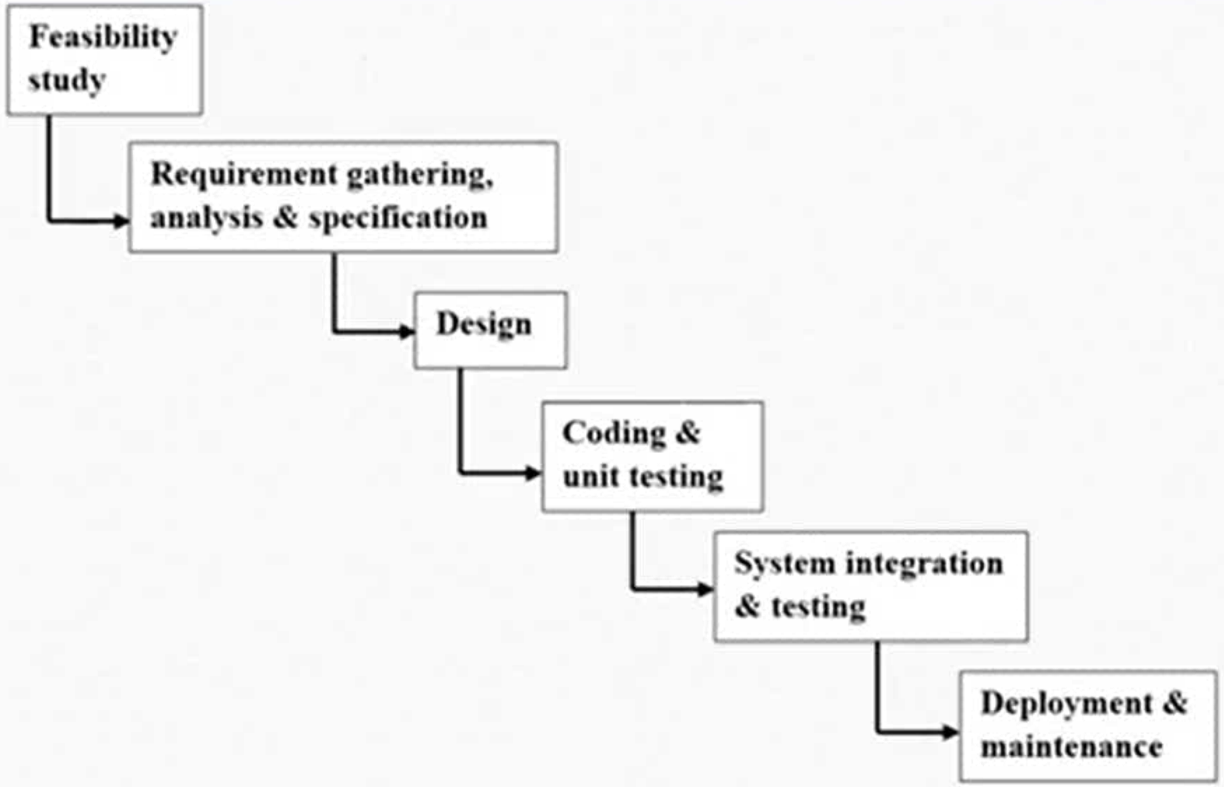
System integration & testing

Deployment & maintenance

Waterfall Model (Iterative)

- We can have iterations between the stages as well

Waterfall Model (Iterative)

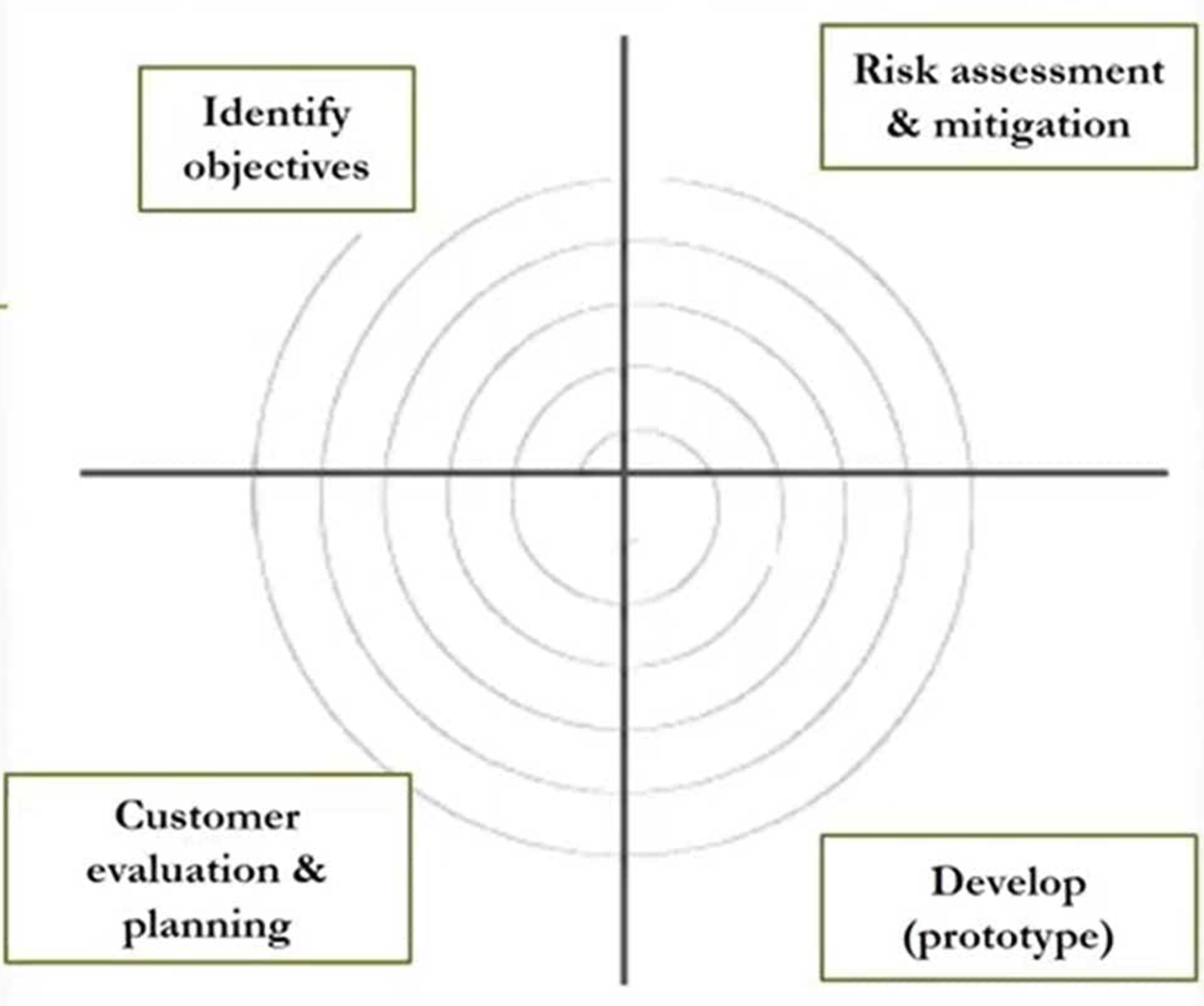

Feasibility study
Requirement gathering, analysis & specification
Design
Coding & unit testing
System integration & testing
Deployment & maintenance

Spiral Model

Spiral Model - Characteristics

- A meta model – encompasses other SDLCs

Spiral Model



Identify objectives

Risk assessment & mitigation

Customer evaluation & planning

Develop (prototype)

Spiral Model - Characteristics

- Multiple cycles (iterations)
- Each spiral – one iteration
- Each iterations – divided into 4 phases (quadrants)

Spiral Model - Quadrants

- 1st quadrant – identify objectives & risks
  - During the first quadrant, objectives of the iterative phase are identified
  - Also the risks associated with these objectives

Spiral Model - Quadrants

- 2nd quadrant - risk assessment and mitigation
  - Identified risks analyzed in details
  - Steps taken to reduce the risks
    - E.g., if there is a risk of inappropriate requirement specification, a prototype system may be developed

Spiral Model - Quadrants

- 3rd quadrant - development
  - Develop product (prototype) after resolving identified risks and evaluate

Spiral Model - Quadrants

- 4th quadrant – evaluation and planning
  - Review results achieved so far with customer and plan next iteration around the spiral

Spiral Model

- Through the iterations, progressively more complete version of the software gets built

Interactive System and SDLC

- Interactive systems should be “usable”
  - Requires users to take into account in every stage of the design
  - Brings in lots of iteration

Interactive System and SDLC

- Difficult to express with traditional SDLCs (e.g., waterfall model)

- In the next lecture, we shall learn about an "iterative" SDLC for interactive systems

Reference

- Rajib Mall (2018). **Fundamentals of Software Engineering**, 5th ed, PHI Learning Pvt Ltd. **Chapter 2**
- Roger S Pressman (2015). **Software Engineering: A Practitioner's Approach**, 8th ed, McGraw-Hill Education, New York, **Chapters 2-4**