Software Engineering (ICT 3159)

Faculty details

- Sucheta Kolekar
- 9008773043
- Sucheta.kolekar@manipal.edu
- Whats APP/MS Teams/Mail/Calls/ Cabin number 12

Subject Overview

- Software Engineering: 4 hrs. per week (3 Theory classes+1Tutorial)

Syllabus Theory

- Introduction to Software Engineering
- Software Engineering Process Models:
- Modeling using Unified Modeling Language(UML)
- Design Engineering
- Architectural and Design Patterns
- Software Testing Strategies and Testing Techniques
- Software Configuration Management and Risk Management
- Project Management: Scheduling and Estimation

Course Objectives: Theory

- To learn the Concepts of Software Engineering
- To understand the Software Development Life Cycle
- To learn the basic principles behind software configuration and risk management
- Model software requirements for application development
- To learn architectural, design patterns for design of complex, scalable software systems

Course Outcomes

- Understand the basics of software development life cycle
- Understand the basic principles behind software configuration and risk management
- Explore the importance of requirement analysis through scenario-based exercise
- Adapt software design strategies using object-oriented concepts
- Identify a suitable testing strategy to validate a given software application

Study Materials

1. Roger S. Pressman “Software Engineering A practitioner's approach”- McGraw Hill, 7th/8th edition, 2015

2. Ian Somerville, “Software engineering”, Addison Wesley, 10th Edition, 2017.

3. Grady Booch, James Rumbaugh, Ivar Jacobson, “The Unified Modeling Language User Guide”, Pearson Education, 2nd Edition, 2015.

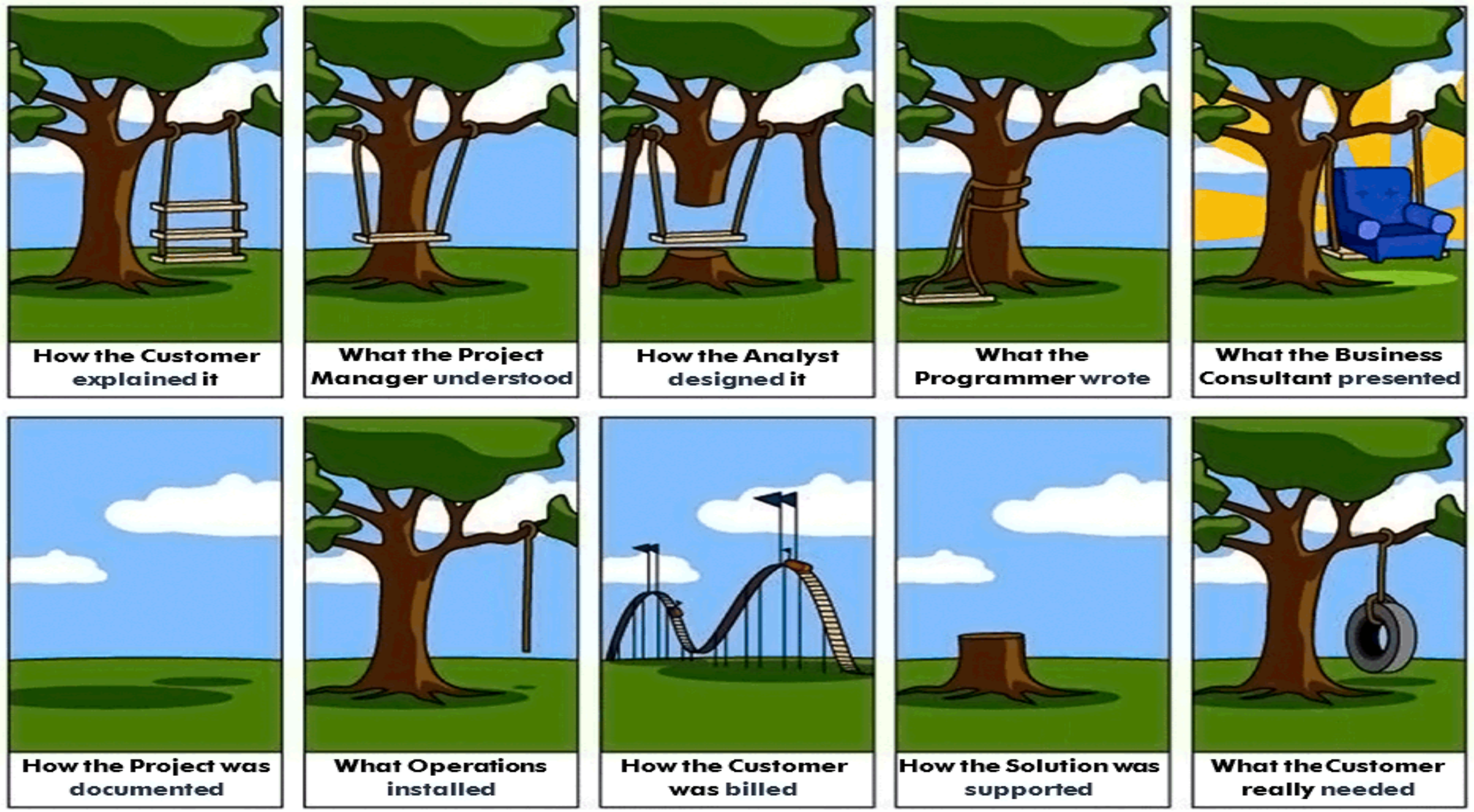
- Slides
- Recordings
- Your own notes

Theory: Mode of evaluation

- Internal Marks: 50
  - 3 Quizzes in 3rd, 5th and 7th week : 10 Marks
  - 2 Assignments in 4th and 6th week : 10 Marks
  - 2 Sessional Tests: 30 Marks (schedule will be received by September)
- External Marks: 50
  - No guidelines are provided yet

Mandatory Requirements

- Stationary: Pencil, Pen, Scale, Eraser, Sharpener, A4 Sheets (Many), Separate Notebook
- Two CR for controlling the meeting
- 6/7 volunteers for interaction during class in the meeting


How the Customer explained it
What the Project Manager understood
How the Analyst designed it
What the Programmer wrote
What the Business Consultant presented
How the Project was documented
What Operations installed
How the Customer was billed
How the Solution was supported
What the Customer really needed

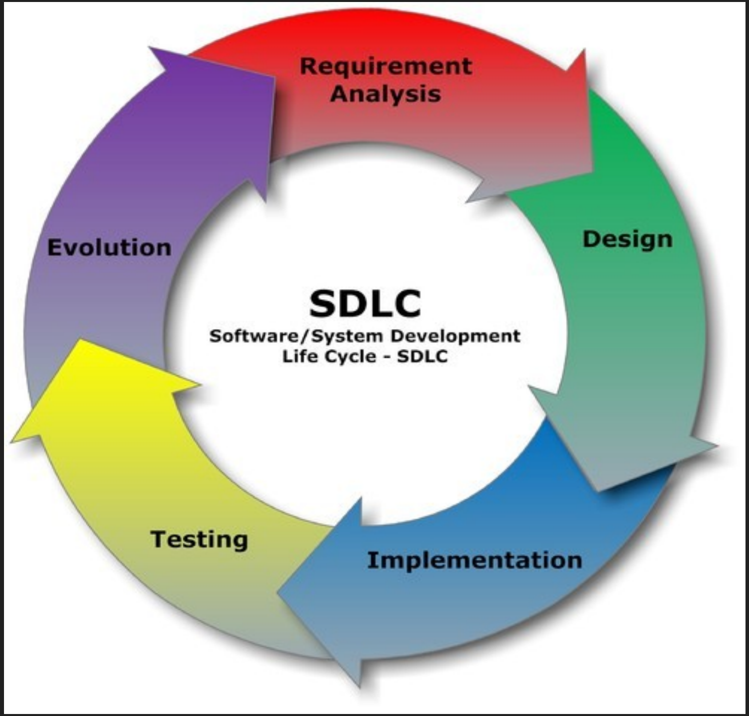
- $250 billion annually in US and approximately equal in India
- Over 175,000 projects!
- Complexity, size, distribution, importance push our limits.
- Business pushes these limits:
  - Great demands for rapid development and deployment

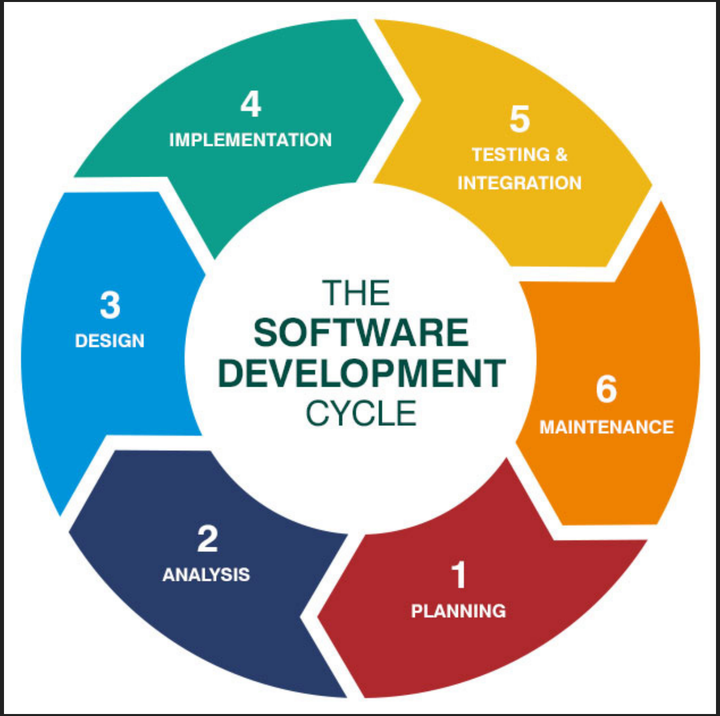
- Incredible pressure: develop systems that are:
  - On time,
  - Within budget,
  - Meets the users’ requirements

- Figures in the late 90s indicated that at most
  - 70% of projects completed
  - Over 50% ran over twice the intended budget
  - $81 billion dollars spent in cancelled projects!!

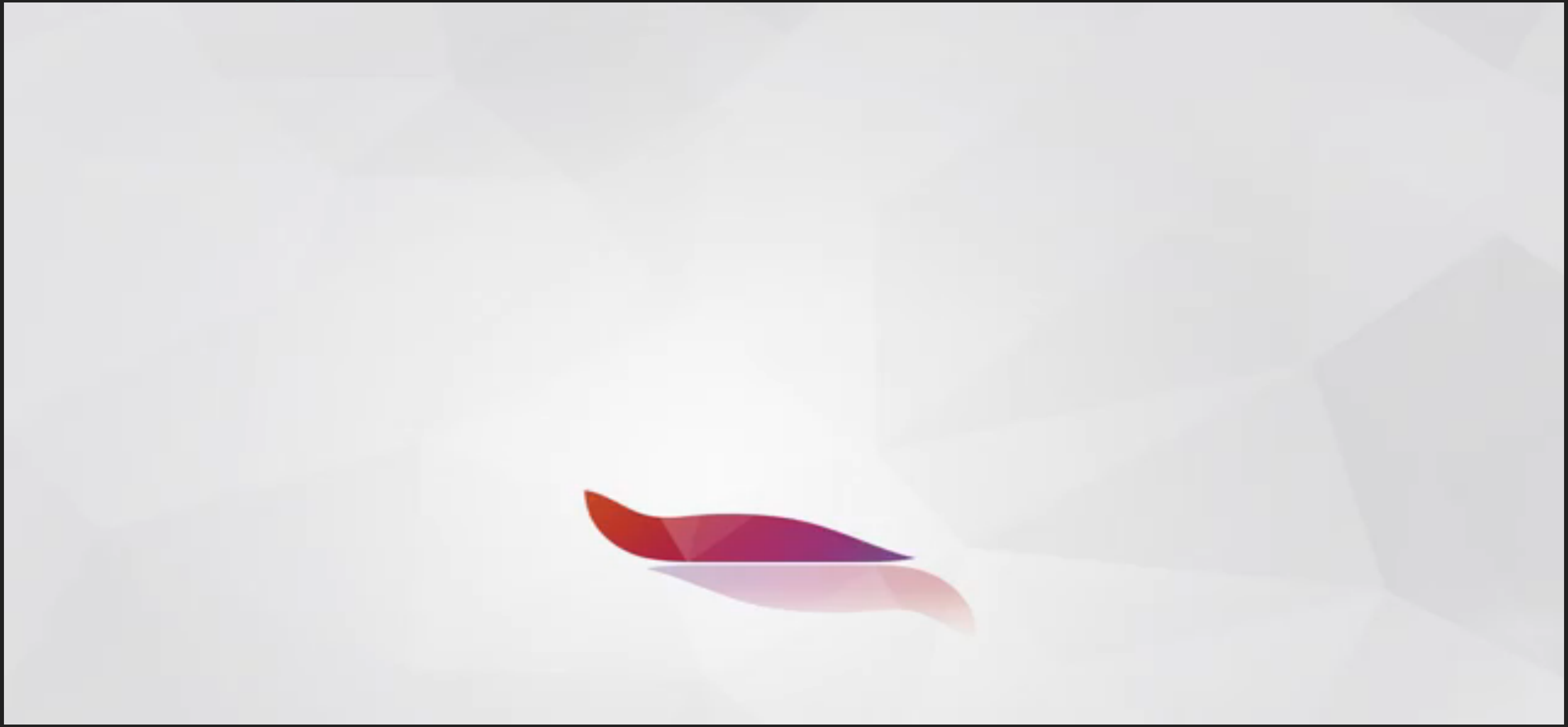
- Getting better, but we need systematic way and better tools and techniques!

Software Development Life Cycle (SDLC)





Software Engineer’s Life Cycle (SELC)



Basic Terminologies

- Software
- Engineering
- Software Engineering
- Requirements
- Design and modelling
- Testing
- Quality
- Coding
- Deployment
- Maintenance
- Updates
- Change/Version control
- Risk

Need of Software Engineering

- **Large software**
- **Scalability**
- **Cost**
- **Dynamic Nature**
- **Quality Management**

Activity

- Job Profiles based on Software Engineering and UML: Everyone should list atleast 10 job profiles.