Software Engineering - Introduction

Samit Bhattacharya

Comp Sc & Engg

Indian Institute of Technology Guwahati

Basic Issues

- Why it is needed?
  - —Suppose I want an adder: it'll give me sum of two input numbers. I ask you to help me get one. What you will do?
  - —What are the issues

- Difference between program vs product
  - —Program: small codes
  - —Product: large (& complex) systems

Styles of programming

- Exploratory style of programming
  - Think in terms of coding (keep on coding as you try to solve the problem)
  - We do this often
  - We should avoid this as this is more suitable for small codes (not for products)

- Modern style (our focus)
  - Follows different stages of development (coding is only a part of it)
  - Required for maintenance & customer satisfaction
  - Key concerns: modularity & life cycle

What we shall learn

- How to build large systems following the SDLC stages
  - —Different SDLC models
  - —Feasibility study
  - —Specification
  - —Design
  - —Prototyping
  - —Coding in module and combining those
  - —Testing
  - —Project management, cost estimation etc
- Note how coding is only one part (involves lots of documentations)

Things to do & marks distribution

- Project ideas to be floated (to be done in groups of 3/4)

- Theory and lab grading will be together (to get good grade, you have to perform well in both)
  - Periodic evaluations (exams): 25%
  - Class interaction (including project progress review presentations): 25%
  - Project implementation: 50%
  - Theory grade (exam+interaction), lab grade (project+interaction)
  - Attendance important (one grade less if <75%, 2 grades less if < 60%)

References

- Rajib Mall – Fundamentals of S/W Engineering
- Roger Pressman –S/W Engineering: A Practitioner's Approach
- Samit Bhattacharya – Human Computer Interaction: User Centric Computing for Design
- Handouts: as applicable
- My NPTEL lectures