INTRODUCTION

• What is Software Engineering?

• How is Software Engineering different from Programming/development?

• What do Software Engineers do?

• Ethical considerations as Software Engineers
What Do Software Engineers Do?

• Requirements
  – Gathering
  – Analysis
• Design
• Salesmanship
• Project management
  – Budgeting
  – Staffing
  – Facilitate discussions & decisions
  – Progress tracking
  – Setting development standards
• Risk management
• Evaluation
• Documentation
• Support

What Do Software Engineers Do?

• Missing Anything?

• Which parts have you been involved in?

• Which parts have you not been involved in?
What is a software lifecycle model?

Why does it matter?
Life-cycle models

Waterfall Model
Royce 1970

Requirements Definition

System & SW Design

Implementation & Unit Testing

Integration & System Testing

Operation & Maintenance

What about mistakes? refinement?

Product: class models +
Product: requirements specifications

Step n+1: Design

complete targeted requirements

Step n+2: Implement

Product: code +

Step n+3: Test

Product: test results +
Life-cycle models

Waterfall Model

Incremental development
(each release adds more functionality)

Evolutionary development
(each version incorporates new requirements)

V Models

“analyse and design”

detailed design

“test and integrate”

code and debug

system requirements

software requirements

system integration

acceptance test

software integration

test

design

code

test

Product: test results

Step 0: Analysis requirements

Step 1: Test

Product: requirements specifications

Step 2: Design

Product class models

Implementation & Test 2

Implementation & Test 1

System & SW Design

Requirement Definition

What about models? inference?
Life-cycle models

- Requirements definition
- System design
- Sub-system development
- System integration
- System installation
- System evolution
- System decommissioning

From Sommerville
Agile/Extreme Programming?

• What is Agile/XP?

Agile/Extreme Programming?

• Key techniques & practices of XP
  – Rapid, Iterative development cycle
  – Continuous integration
  – Test-first mentality
  – Pair-programming
  – Code review
  – Flat structure
  – Planning Poker
  – Prototyping
Work breakdown & scheduling

• What have you covered?
• What is involved in this?

Work breakdown & scheduling

• Activities, milestones & deliverables
• Task dependencies & scheduling
• Activity networks
• Gantt charts
• Margins for error (slack in the schedule)?
Cost estimation

- Types of costs
  - Hardware Costs
  - Travel & Training costs
  - Effort costs

- Magnitude, overhead, margins
- General approaches to cost estimation

Managing team

- What have you covered?
- What is involved in this?
Managing Team

- Costs/problems of adding new programmers to existing project
- Diminishing returns on programmer productivity
- Problems to be overcome in adding new people to a project
Requirements gathering

- Ethnographic methods
- Interviewing
- Questionnaires
- Focus groups
- Studying documentation and artifacts
- Participatory design
- Prototyping
- Goal-oriented analysis
- …etc

Requirements analysis

- Scenarios
- Use-cases
- “Profiles”
- IEEE requirements specification documents

- Goal-oriented analysis
- Prototypes
Modeling requirements

• Data-flow model
• Composition model
• Architectural model
• Classification model
• State transition model

Problems with requirements?

• Consistency problem
• Completeness problem
• Ambiguity/lack of clarity problem
• ....

• Solution?
  – Formal specifications...?