

Software Requirement Engineering

Muhammad Waseem

Lecture 1

Course Key se311/

Date:10.02.2014

Department of Computer Science and Software Engineering, FBAS
,International Islamic University Islamabad

Objective of this Lecture

- Discussion on
 - course Outline
 - Quizzes Method
 - Attendance Policy
 - Assignments Guide line

Course Objective:

- Gain understanding of
 - what is **requirement engineering**,
 - Different **stages and concepts** of requirement engineering.
- To describe in detail the
 - different **activities of requirement engineering**.
- To give you a practical knowledge on
 - how to make certain requirement engineering **artifacts**.

Syllabus

- **Introductory stuff**
 - What are Requirements?
 - What is Engineering?
 - What is a System?
- **Basic RE activities**
 - Planning and Eliciting Requirements
 - Modeling and Analyzing Requirements
 - Communicating and Agreeing Requirements
 - Realizing and Evolving Requirements
- **Advanced Topics**
 - Inconsistency and Uncertainty in RE
 - Use of Formal Methods in RE [**optional**]

Introductory Stuff

- **What are Requirements?**
 - Scope (for this course):
 - Separating the Problem from the Solution
 - What Requirements Engineers do
- **What is Engineering?**
 - Engineering as a profession
 - Engineering projects
 - Engineering lifecycles
 - Engineering design
- **What is a System?**
 - Formal foundations of software systems
 - Conceptual foundations of information systems
 - Empirical foundations of human activity systems
 - Observability of systems

Eliciting and Planning

- **Elicitation Targets**
 - Stakeholders & User Classes
 - System boundaries
 - Goals
 - Scenarios
- **Elicitation techniques**
 - Interviews, questionnaires, surveys, meetings
 - Prototyping
 - Ethnographic techniques
 - Knowledge elicitation techniques
 - Conversation Analysis
 - Text Analysis
- **The Feasibility Study**
 - Types of Feasibility
 - Cost/benefit analysis
- **Risk Analysis**
 - Identifying and managing risk

Modeling & Analyzing

- **Basics of modeling**
 - Notations and their uses
 - Formality and Expressiveness
 - Abstraction and Decomposition
 - Model management and viewpoints
 - Types of Analysis
- **Enterprises**
 - Business rules and organizational structures
 - Goals, tasks and responsibilities
 - Soft Systems analysis
- **Information Structures**
 - Entities and Relationships
 - Classes and Objects
 - Domain Ontologies
- **Behaviors**
 - Activities and Interactions
 - States and Transitions
 - Concurrency
- **Quality Requirements**
 - Taxonomies of NFRs
 - Performance
 - Usability
 - Safety
 - Security
 - Reliability
 - Maintainability

Communicating & Agreeing/ approving

- **Validation**
 - Refutable descriptions
 - Role of contracts and procurement
 - Role of organizational politics
- **Documenting Requirements**
 - Properties of a good specification
 - Documentation standards
 - Specification languages
 - Making requirements testable
- **Prototyping and Walkthroughs**
 - Throwaway prototyping
 - Operational prototyping
 - Walkthroughs of operational models
- **Reviews and Inspections**
 - Effectiveness of Inspection
 - Conducting an Inspection
 - Collaborative Requirements Workshops
- **Negotiation and Prioritization**
 - Representing argumentation and rationale
 - Computer-supported negotiation
 - Trade-off analysis
 - ! Release planning

Realizing and Evolving

- Software Evolution
 - Laws of evolution
 - Release planning
 - Product families
 - Requirement Reuse
- Requirements and Architectures
 - Architectural Patterns and Description
 - Languages
 - Mapping requirements to architectures
 - Architectural Robustness
- Managing Change
 - Baselines and change requests
 - Configuration management and version control
 - Impact Analysis
- Traceability and Rationale
 - Pre- and Post- traceability
 - Capturing Design Rationale
 - Traceability techniques
- Managing Inconsistency
 - On the inevitable intertwining of
 - inconsistency and change
 - Learning from inconsistency
 - Feature interaction
 - Living with inconsistency

Marks Distribution:

- Quizzes: 10%
- Assignments: 15%
- Midterm: 15%
- Final Exam: 60%

Rules to follow

- No **plagiarism** will be allowed, if **full or part of assignment** or **quiz** is copied, the student will be given zero mark.
- A student **10 minutes** late in class will be considered absent.
- **75% attendance** policy will hold.
- There will be surprise quizzes.

Many books on RE exist

no one textbook covers the field well

- Student textbooks

- A. Davis, Software requirements: objects, functions and states, Prentice Hall, 1993.
- G. Kotonya and I. Sommerville, Requirements Engineering: Processes and Techniques, Wiley, 1998.
- P. Loucopoulos and V. Karakostas, System Requirements Engineering, McGraw Hill, 1995.
- L. A. Macaulay, Requirements Engineering, Springer Verlag, 1996.
- R. J. Wieringa, Requirements Engineering: Frameworks for Understanding, Wiley, 1996.
- Flynn, D., Information Systems Requirements: Determination and Analysis, McGraw Hill, 1992

- Practitioner textbooks

- J. Andriole, Managing Systems Requirements: Methods, Tools, and Cases, McGraw-Hill, 1996.
- D. C. Gause and G. M. Weinberg, Exploring Requirements: quality before design, Dorset House, 1989.
- D. C. Gause and G. M. Weinberg, Are Your Lights On?: How to Figure Out What the Problem Really Is, Dorset House, 1990.
- J. O. Grady, System Requirements Analysis, McGraw Hill, 1993.
- I. S. Graham, Requirements Engineering and Rapid Development: A Rigorous, Object-Oriented Approach, Addison-Wesley, 1998.
- B. L. Kovitz, Practical Software Requirements: A Manual Of Content And Style, Manning Publications, 1998

Questions



dreamstime.com