

CEN 6075 Software Requirements Engineering

Credits: 3 credits

Text book, title, author, and year: <<The Essence of Software Engineering: Applying the SEMAT Kernel>>, Ivar Jacobson , Pan-Wei Ng, Paul McMahon , Ian Spence, Svante Lidman, Addison Wesley, Fall, 2012 (Draft/Rough Cut available), ISBN-13: 978-0321885951, App. 250 pages <<Requirements Engineering: Fundamentals, Principles, and Techniques>> by Klaus Pohl, Springer; 1st Edition. (July 23, 2010), 832 pages

Reference materials: Lecture notes, working environments, tooling, and other references will be posted on Blackboard

Specific course information

Catalog description: Principles of requirements elicitation, specification and analysis. A broad range of methods will be presented in the context of how they support these principles. Both functional and non-functional requirements will be addressed. Other topics include problem analysis, modeling, requirements documentation, and prototyping.

Course Special Features:

The course is founded on the theory of software engineering as represented in the recent work of SEMAT, a community effort for the establishment of Software Engineering Method and Theory, and the resulting Essence input to an emerging standard from OMG – on the Essentials of Software engineering. The course uses SEMAT and Essence principles to present and teach the area of Requirements Engineering. It also shows how Requirements Engineering fits into Enterprise Architecture frameworks like DODAF/MODAF/UPDM, ISO RM/ODP, TOGAF and Zachman.

The Requirements Engineering framework of the course is based on the framework of IREB, the International Requirements Engineering Board, focusing on Requirements artifacts (Goals, Scenarios, Solution-oriented Requirements), Core Activities (Documentation, Elicitation, and Negotiation), and System Context (with Customer, System and Endeavour). Alternative requirements engineering methods and practices will be presented and discussed. Goal modeling will be demonstrated through BMM, i*, RML and KAOS. Scenarios will be created using Use cases, user stories, BPMN, UML and the new Use cases 2.0. Domain models will be demonstrated through structured natural language expressions, fact models, business rules, UML models and ontology models.

The course will contain practical project exercises and use of tools. A special feature of this course is that we will have exclusive academic use of Ivar Jacobson International (IJI) commercialized Essential Unified Process and Use Case 2.0 material and tooling in our class.

The course starts with an introduction to the Essential Unified Process as a context for showing model-based requirements engineering through the use of UML and Use cases. Agile Requirements Engineering is then presented and contrasted, with the use of a backlog of user stories, features and epics. Goal modeling and scenario modeling is introduced next, together with system level requirements for data, function and behaviour. Non functional requirements, constraints and risk analysis practices is following after this. The course ends with a summary and discussion of the introduced practices for requirements elicitation, documentation, negotiation,

validation and management, and concludes with how the participants can customize their own requirements engineering method through and Essence composition of best practices.

Prerequisites: CEN 4010, CEN 5035, or another introductory course in software engineering

Specific goals for the course: Proficiency in the areas of software design and development, data structures, and operating systems

An ability to plan and execute an engineering design to meet an identified need

An understanding of the overall human context in which engineering and computing activities take place

Brief list of topics to be covered:

1. Course Overview
2. Software Engineering Bootcamp
3. SEMAT and Essence
4. Essential Unified Process
5. Essential Unified Process and EA (TOGAF/UPDM/ODP)
6. FDD (as an Essence method) in Ericsson Development
7. Agile and the ARE - Agile Requirements Engineering method (Customer, Solution, Endeavor)
8. Business/Enterprise/Customer practices – Business Use Case practice
9. Scenarios – BPMN 2.0 models – Business process practice
10. Domain models/Concept models/Ontologies/UML class diagram/ Facts/Rules
11. Structured Natural language documentation – FactExpress/Business Rules
12. Goal Modeling (RE), i*, KAOS, BMM practices
13. Scenarios – User Stories and Agile RE practices
14. Scenarios – UI Mockups Interaction practices – Prototypes - Balsamiq
15. Scenarios – Use cases 2.0, practice
16. Use case driven Testing – TDD practices
17. System level requirements, Data/Function/Behaviour
18. Service specification with SoaML, Composition with BPMN
19. Non Functional Requirements practices– Constraints etc
20. Risk analysis practices - CORAS (chapter/book with tool)
21. Requirements ELICITATION and NEGOTIATION – practices (Pohl)
22. Requirements VALIDATION and MANAGEMENT – practices (Pohl)
23. SUMMARY OF COURSE – Making your own Requirements Engineering method of practices