



SECOND SEMESTER 2019-2020

Course Handout Part II

Date: 06-01-2020

In addition to part-I (General Handout for all courses appended to the time table) this portion gives further specific details regarding the course.

Course No. : **IS F341**
Course Title : **Software Engineering**
Instructor-in-Charge : **Prof. Narasimha Bolloju**

Scope:

Type of Software, Application of Software, Software Life Cycle Model, Agile Modeling, Requirement Engineering, Object Oriented Analysis and Design (using UML), Introduction of Web Engineering, Software Integration and Testing, Support Processes and Software Quality, Software change Management, Software Product and Process Metrics Measurement, Software Project Management, Scheduling, Staff Measurement, Risk Management, Component based Software Engineering, Quality and Reliability Model, Maturity Models, Deployment Models, Ethics of Software Engineering (ACM/IEEE code of Ethics).

Objectives:

This course provides the knowledge and skills necessary to effectively participate and contribute to project teams in software development following a suitable methodology. Specifically, the course has the following objectives:

- to offer widely used software development methodologies so that the students will be able to select an appropriate software process model and architecture for a given type of development project,
- to elaborate the relationship of software development to overall software product engineering, estimate time and costs,
- to explain the rationale behind various activities related to planning and management of software development in teams, and
- to make students build a software system in small teams adhering to a widely used agile methodology by practicing relevant techniques pertaining to activities such as requirements elicitation and analysis, requirements modeling and specification, software development, testing and project management.

Course Plan:

Lecture Nos.	Learning objectives	Topics to be covered	Chapter in the Text Book
1-6	<ul style="list-style-type: none">- Explain challenges in software development- Explain typical characteristics of agile methods	Introduction software engineering (SE); Challenges associated with SE; Types of software systems; Plan-based vs. agile development; Agile project management and Scrum; Work systems (Socio-technical	T1-Chapters 1, 3, and 4; R1 – Chapter 3



	<ul style="list-style-type: none"> - Prepare system scope and top level system requirements 	systems); Requirements engineering overview; functional and non-functional requirements; Business rules; Requirements specification with user stories	
7-12	<ul style="list-style-type: none"> - Practice requirements modeling skills using UML through a UI/UX design methodology 	Introduction to a UI/UX design methodology and UML; Identifying user profiles; Developing personas; Defining requirements in terms of user stories and use case diagrams; Interaction design with Use case descriptions and Activity diagrams; Information architecture; Mind maps and card sorting for UI design; Low fidelity prototyping with wireframes	T1-Chapters 4 and 5; R3
13-18	<ul style="list-style-type: none"> - Understand and apply object modeling skills - Develop object models 	Domain/object modelling; Attributes and different types of relationships in object models; Different types of operations/methods; NoSQL; schema-less databases; CAP theorem; Document modelling	T1-Chapter 5; R4 – Chapters 1 and 2
19-21	<ul style="list-style-type: none"> - Understand basic software testing concepts - Learn and practice effort estimation techniques 	Introduction to software testing; Acceptance tests; Test planning; Validation vs. verification; Estimating effort in plan-driven methods; Estimating effort in agile methods	T1-Chapter 8
22-27	<ul style="list-style-type: none"> - Explain and compare different architectural and design choices - Apply patterns for effective and efficient object modeling 	Software architectures; Architecture principles; Design principles; Component diagrams; Cohesion and coupling; Object modeling with transaction pattern	T1-Chapter 6
28-30	<ul style="list-style-type: none"> - Explain various software testing concepts and their role in software engineering 	Usability testing; A/B testing; Software testing concepts (validation/verification; types of testing); Test cases and test data; Unit testing; Integration testing; Interface testing, Acceptance testing; Regression testing	T1 - Chapter 8
31-36	<ul style="list-style-type: none"> - Understand and compare different types of process models - Explain effort estimation techniques 	Software development life cycle; Process models for software development; Component based development; Suitability of process models to a given project; Kanban and Scrumban methodologies; Agile methods for large distributed teams	T1-Chapters 2, 3, and 23
37-39	<ul style="list-style-type: none"> - Understand planning and management of development processes 	Project planning; Quality management; Risk management; Configuration change management; Process improvement	T1-Chapters 23 to 25
40-42	<ul style="list-style-type: none"> - Explain design patterns and their application - Appreciate ethical issues in software engineering 	Design patterns and applications of popular design patterns; Ethics for software engineers	T1-Chapters 1, 7

Text book:

T1 - Software Engineering (10th edition) by Ian Sommerville, Pearson Education, 2017.

Reference books:

R1 - The work system method: connecting people, processes, and IT for business results by Alter S., Work System Press, 2006.

R2 - Essential Scrum – A practical guide to the most popular agile process by Rubin, K. Addison-Wesley, 2013

R3 - UX Design and Usability Mentor Book: With Best Practice Business Analysis and User Interface Design Tips and Techniques by Yayici, E., UX Services, 2014.

R4 - NoSQL distilled: a brief guide to the emerging world of polyglot persistence by Sadalage PJ, and Fowler M. Pearson Education; 2013.

Evaluation Scheme:

Component	Duration	Weightage (%)	Date & Time	Nature of Component
Mid Term Exam	90 minutes	20%	6/3 1.30 -3.00 PM	Closed book
Class/lab exercises	-	20%	Spread through out the semester	In class/lab & take home exercises
Project and presentation	-	30%	Spread through out the semester	Group project
Comprehensive	3 hours	30%	13/05 FN	Closed book

As part of the group project, students will work on a software development project following Scrum - a popular agile method - in small groups and submit associated reports. Each group will also make a presentation including the demonstration of the developed software product at the end of first or second sprint. After the third and final sprint, a demo video will be submitted by each team for viewing and for soliciting inputs from other students of this course.

Group project work requires full stack development with currently used tools/environments (e.g., React JS, Node.js, MongoDB). In addition to the development tools groups are required to use project management tools (e.g., PivotalTracker), cloud-based modeling environments for UML and UI/UX design (e.g., LucidChart), and other supporting tools for testing and test data generation (e.g., Jest and Mockaroo).

Chamber Consultation Hours: To be announced.

Notices: All notices pertaining to this course will be posted on CMS. Announced in the Lectures.

Make-up Policy: Make-up will be granted strictly based on prior permissions and on justifiable grounds only. There is no make up for the Assignments/Presentation component.

Academic Honesty and Integrity Policy: Academic honesty and integrity are to be maintained by all the students throughout the semester and no type of academic dishonesty is acceptable.

INSTRUCTOR-IN-CHARGE

