

COURSE SYLLABUS

CSI 3370 – Software Engineering and Practice

Location: MSC 120

Winter 2018

Course Catalog Description

Introduction to software engineering and practice. Topics include software process models, project management, requirements analysis, software quality assurance, and testing.

Course Detail: The field of software engineering includes problem definition, analysis and design methodologies, engineering processes, quality assurance, and maintenance. This course will cover various engineering aspects of developing software in context of web or mobile software system.

Prerequisite: Major standing and CSI 230

Class Timing: 3:30PM – 5:17PM Monday & Wednesday
MSC 120

Instructor: Dr. Nilesh Patel

Office Location: EC 524

Office Hours: By Appointment or 3:00PM – 3:30PM Wednesday

Email: npatel@oakland.edu

Office Phone: (248) 370-2247

Course Web Page: Oakland University Moodle URL

Text book:

- **Software Engineering by Ivan Marsic (Available on Moodle for download)**
- **Slides and Material Uploaded on Moodle**

Reference Books:

- Object Oriented and Classical Software Engineering by Stephen R. Schach – 8th Edition
- Play for Java Nicolas Leroux and Sietse de Kaper, Softbound print: January 2014 (est.) | 325 pages ISBN: 9781617290909
- Engineering Long-Lasting Software: An Agile Approach Using SaaS and Cloud Computing by Armando Fox and David Patterson (Available as eBook)

Course Contents

Week Of	Day of the Week	Topic
1/3/18	Wednesday-II	Introduction and Syllabus
1/8/18	Monday-I	Java and Motivation for Software Engineering
	Wednesday-II	Software Lifecycle Models
1/15/18	Monday-I	Dr. Martin Luther King Jr. Day
	Wednesday-II	Software Config. Mngt. - I
1/22/18	Monday-I	Software Config. Mngt. - II
	Wednesday-II	Requirements Engineering - I
1/29/18	Monday-I	Requirements Engineering - II
	Wednesday-II	UML - I
2/5/18	Monday-I	UML - II
	Wednesday-II	Requirements Analysis - I
2/12/18	Monday-I	Requirements Analysis - II
	Wednesday-II	Project Work Starts
2/19/18	Monday-I	Winter Break
	Wednesday-II	
2/26/18	Monday-I	Domain Modeling - I
	Wednesday-II	Domain Modeling - II
3/5/18	Monday-I	Iteration - I (student Presentation)
	Wednesday-II	Quality Assurance - II
3/12/18	Monday-I	Quality Assurance - II
	Wednesday-II	Iteration - II (Students Presentation)
3/19/18	Monday-I	Project Estimation & Risk Management - I
	Wednesday-II	Project Estimation & Risk Management - II
3/26/18	Monday-I	Iteration - III(Students Presentation)
	Wednesday-II	Defect Management
4/2/18	Monday-I	Project Release Management
	Wednesday-II	Software Metrics
4/10/17	Monday-I	Final Project Presentation (Group - 1)
	Wednesday-II	Final Project Presentation (Group - 2)
4/17/17	Monday-I	Exam Preparation
4/19/17	Wednesday-II	Final Exam

Grading Item	%Weight
1. Short Quizzes	30%
2. Assignments	30%
3. Team Project	
3.1 Sprint 1	5%
3.2 Sprint 2	5%
3.3 Sprint 3	5%
3.4 Peer evaluated weighted grade(Final Demo/Documentation)	15%
4. Final Exam	10%

Final Grade

Weighted average score will be scaled to assign final Digit grades according to the following policy:

- For students receiving weighted grade between 71 and 100 Final grade will be decided based on distribution curve and it will be between 4.0 to 2.0
- Students having weighted total grade < 70% will fail the class

Final Grade

Students must prepare themselves for a possible quiz/assignment every week. I will try to give a short quiz randomly. Quiz may contain between 10 and 20 multiple-choice questions. **Since quiz will be conducted on moodle, everyone shall be able to give it irrespective of his/her presence in the class. It will be students' responsibility to check moodle for a quiz every week. No second attempt will be allowed on a missed quiz, unless a medical certificate is presented.**

Workload per week:

Approximately 6-8 hours or more based on prior knowledge of prerequisites and self-motivation.

Team Project

Team project will focus on use of the concepts, tools and techniques described in the course to advance a project focused on the technical theme. Students, in self-selected, 5-7member teams, will go through planning, requirement elicitation, design and implementation of a desktop/web-based or Mobile application. Team may choose any development environment and application tools that is appropriate for project. The instructor will determine appropriateness. Team formation must occur before the proposal is submitted and membership will be considered fixed from that time —no team switching will be allowed.

By submitting/presenting a project, team members attest that they all participated in the conceptualization and accomplishment of the project. It is incumbent on team members to assure that no one on the team receives free rides through the project. If problems arise during the term, upon consultation with team members, the instructor will remove non-participating team members from their teams. Individuals removed from teams will not receive points on the team assignment.

The successful completion of the course should provide you with skills to develop software using defined processes, tools and methodologies.

Team Formation Guideline:

Each project team will not have no less than 5 or more than 7 members. You must ensure that team is comprised of complimentary skills. Computer Science and Information Technology projects require requirements elicitation, analysis and design, implementation and testing skills. Over the period of project development, each member must demonstrate skills in number of areas as per various efforts areas identified in **Efforts Matrix.**

Project Development Guideline:

We will follow an agile software development process. Each team will divide the feature set they plan to develop into 2 weeks long sprint. At the end of each sprint, on Monday or Wednesday class, team will capture various project artifacts. All artifacts are quantitatively measured as defined in **Effort Matrix.**

- Tools, technology and Algorithm Research
- Customer Statement of Requirements
- Glossary of Terms
- System Requirements
- Use case Requirements
- Class or Entity Relationship Diagram
- Interaction / State Diagram
- System Architecture
- User Interface Design
- Unit Tests
- System Test Designs
- Implementation
- Algorithms
- Project Plan Update
- Power point Preparation
- Test Execution and Bug Reporting

For each sprint, team members will divide responsibilities with non-overlapping efforts such that produced artifacts measure individual contribution and effort. Software activities are highly coupled and dependent on each other. Members who are responsible for requirements must complete their activity ahead of time to ensure that downstream activities get enough time for completion. To ensure planning of such dependency, all team must generate quality artifact to help team succeed.

Your work will be assessed in bi-weekly deliverables. At the end of each month, team will submit the code, documentation and other project artifacts. While working in harmony, real world projects do implement freedom and control to ensure that each member delivers the best artifact without influence of the other. Each team member is expected to give minimum between 8 to 10 hours per week of their time working on the project. **Weak deliverable by any team member will result in 0 grade for that week.**

Effort and Responsibility Matrix on Next Page with Bi-weekly Grading Mechanism

Your Name	John Doe					
Project Name	Example Template			You must provide accurate artifacts created by you. For example		
Sprint#	Efforts (Hrs)	Towards objectives	Brief Description of Accomplishments	Artifact Name	Location	Reference
Sprint # 1	16	B, L	Wrote usecases UC1, UC2. Developed practice and procedure for effectively using GIT	Requirements	Requirements.doc	Pages 11, 12
Sprint # 2				Class Diagram	Design.doc	Page 3,4
Sprint # 3				Powerpoint	Sprint2.ppt	Slide 4 to 6
Sprint # 4				Code	Someclass.java	New code(20), Updated(3)
Sprint # 5				Project plan	plan.prj	task 1 to 5
Sprint # 6				Defect	Bugzilla	Bug ID - 1, 3, 7
Sprint # 7				Unit test	testcase.java	New code(23), Updated(1)
Sprint # 8				Test data	testcase.java	New test data (5)
Sprint # 9				Research Report	Report.doc	page 1 to 20
Sprint # 10				Project Setup	Joining Team.doc	page 1 - 10
Sprint # 11				Deployment	build scripts..	Specifcs of the script and purpose
Sprint # 12						

Other Requirements:

No late submissions or makeup exams will be permitted except in cases of documented emergencies. All assignments artifacts must be compressed and submitted using naming convention: Assignment#-Your First Name-Your Last Name. File extension, for example, Assignment1-John-Doe.zip. Expanded folder after uncompressing the zip file must have folder name without zip extension. Assignments submitted without this naming convention will not be graded.

Statement on Academic conduct:

The Oakland University values academic honesty and integrity. Each student has a responsibility to understand, accept, and comply with the university's standards of academic conduct, as well as policies established by the schools and colleges. Cheating, collusion, misconduct, fabrication, and plagiarism are considered serious offences. Violations will not be tolerated and may result in penalties up to and including expulsion from the University.