Syllabus - Software Development

instructor

Zachary O. Dugas Toups
z@cs.nmsu.edu; 575-646-1605
office hours: M 2:00–4:00pm & by appt.; Science Hall 136

teaching assistant

Essa Imhmed
essa@nmsu.edu
office hours: F 10:00am–noon & by appt.; SH 149

original course designer

Jonathan Cook

text & software

required text

Shari Lawrence Pfleeger and Joanne M. Atlee. *Software Engineering: Theory and Practice*, 4th Edition. Printice Hall. 2010.

software

Other than requiring a compiled language, no language is specified for your project, but you will need to identify one and set up your own development environment.

You are required to use GitHub to manage your team project files. The core of Git is available below (you may have it, or it may be bundled with a graphical user interface):

Git 2.9.2 or higher. Online: **https://git-scm.com**

It is recommended that you have a GUI available; the following are suggested:

SourceTree 2.1.3 or higher. Online: **https://www.sourcetreeapp.com**. Available for OS X and Windows.

SmartGit 7.1.4 or higher. Online: **http://www.syntevo.com/smartgit/**. Available for all platforms.

course description

This course concerns the development of large systems that take multiple people to create. We will look at the process of developing software, what techniques can be used to reliably develop good software. We will look at specific techniques for determining what the program should do (requirements and specification), how it should do it (design), how to build it (implementation), how to verify that the software in fact does do what it should (testing), and how to determine if software has been suitably developed for its purpose (evaluation).

During this course, students will work in teams to apply the techniques learned in class to the development of a non-trivial system. Lab time in the course will be used for project work and to learn about tools that are available to help software developers create robust, reliable, and efficient systems.

objectives

Students completing this course will be able to:

- Understand the major software development activities of requirements analysis and specification, high- and low-level design, implementation, verification and validation (testing), deployment, and maintenance.
- Understand the different styles of software development processes that structure the activities.
- Be able to construct a domain model using UML.
- Understand the use of other UML diagrams (especially use cases, statecharts, sequence charts) in both requirements and design activities.
- Understand the importance of architectural styles and design patterns.
- Understand the difference and white and black box testing, and specific techniques within each.
- Understand the availability of tools that support the software development process and specific activities; be able to use a version control system in a project and be exposed to automated build and testing tools.
- Understand how programming practices effect and are effected by the other activities in the software development process.
- Communicate software design and evaluation with presentations, demos, and reports.

prerequisites

A C or better in CS 272, CS 273, and CS 278.

team-based learning

This course borrows materials from team-based learning, but does not implement it completely. You can expect that:

- You will be responsible for readings outside of class.
- There may be a quiz on every reading at the start of class; if so, you will take it individually, then take it with your team.
- The lectures will be short. Students will present some work.
- There will be a team project, with deliverables throughout the semester, that will take you from concept to complete system.

policies

team formation

Teams will be formed after students have identified their areas of interest through the Micro Proposals assignment. Teams are limited to 4 students, although teams of 3–6 will be considered when necessary.

reading / quizzes / exams

You are responsible for reading all materials prior to class. Throughout the semester, students will take readiness assurance quizzes based on the reading. Missed quizzes **cannot** be made up. The course will include a comprehensive exams that will cover all topics up to that point. There will be no final exam.

total		100	
readiness assurance tests*	2.5		
team readiness assurance tests	2.5		
exams*	25		
homeworks / programs / labs	30		
participation*	10		
final project	30	←	100%
micro proposal*	1.5		5%
problem statement	1.5		5%
initial user stories	0.75		2.5%
functionality schedule	0.75		2.5%
initial requirements	1.5		5%
design document & updated reqs.	1.5		5%
test report	1.5		5%
final documentation	3		10%
final implementation	3		10%
demo 0	0.75		2.5%
demo 1	1.5		5%
demo 2	1.5		5%
demo 3	2.25		7.5%
demo 4	3		10%
repository usage*	3		10%
peer evaluation*	1.5		5%
final project individual scale*	1.5		5%

attendance / class participation

Attendance is expected at every class. Students should be present both physically and mentally, asking questions, discussing, and not otherwise engaged (in a device). A student with more than one unexcused absences will lose points from the Presentation Questions / Class Participation component of his/her grade.

assignments

Normally, assignments are due before class on the day of presentations, and by midnight for written work (as specified in Canvas). **Late work will be accepted, but with a 10% penalty per day late (or 20% per class day, if the work is to be presented).** No extensions will be granted for failure to install required software.

You will work with a team most of the semester, and team activities make up most of your grade. **Students are only eligible for the team portion of the grade if they earn at least 70% of their individual grade.**

grade	%
A	93.5–100%
A-	89.5–93.4%
B+	86.5–89.4%
B	83.5–86.4%
B-	79.5–83.4%
C+	76.5–79.4%
C	73.5–76.4%
C-	69.5–73.4%
D	59.5–69.4%
F	<59.4%

TENTATIVE schedule

date	topic	reading	activity	lab	due
1/19 Th					
1/24 Tu	Why software engineering? + Motivation	ch1, ch11.1			
1/26 Th			Git lecture	lab 1 (L1): initial Git	homework 1 (HW1): setup accounts [BEFORE LAB]
1/31 Tu	Modeling the process & life cycle				program 1 (P1)
2/2 Th		ch2		project pitches & team formation	micro proposals [BEFORE LAB]
2/7 Tu	Planning and managing the project				
2/9 Th		ch3			
2/14 Tu			flash presentations		P2
2/16 Th	[prof. @ NSF]				
2/21 Tu	Capturing the requirements				HW2
2/23 Th		ch4			project: problem statement
2/28 Tu	[prof. @ GDC]				proj: initial user stories
3/2 Th					
3/7 Tu			exam 1	demo 0	HW3: plan to graduation
3/9 Th	Designing the architecture	ch5			proj: functionality schedule
3/14 Tu	"				proj: initial requirements
3/16 Th	Designing the modules	ch6		demo 1	iteration 1 survey (I1S)
3/21 Tu	Spring Break				
3/23 Th					
3/28 Tu	(Designing the modules)				
3/30 Th			exam 2		
4/4 Tu	Writing the programs	ch7			HW4
4/6 Th			architecture presentation		peer evaluation 1 (PE1), I2S
4/11 Tu	Testing the programs	ch8		demo 2	
4/13 Th					proj: design document & updated reqs.
4/18 Tu	Testing the system	ch9			P3 (coverage)
4/20 Th					I3S
4/25 Tu	[sick]				PE2, P4 (JUnit)
4/27 Th	Evaluating...	ch12		demo 3	
5/2 Tu	"	"			
5/4 Th			exam 3		test report
5/11 Th	final exam 10:30am–12:30pm [prof. @ CHI]			demo 4 (final)	
5/14 Su	(materials due date)				final doc.; final project imp.; final project individual scale

university policies

academic honesty

Plagiarism is using another person's work without acknowledgment, making it appear to be one's own. Intentional and unintentional instances of plagiarism are considered instances of academic misconduct and are subject to disciplinary action such as failure on the assignment, failure of the course or dismissal from the university. The NMSU Library has more information and help on how to avoid plagiarism at http://lib.nmsu.edu/plagiarism/.

As programmers, reuse is an essential part of our work. You are welcome to use existing libraries and reuse your own code, but must make certain to appropriate document and provide licenses. You must adhere to any licensing terms and are responsible for any fees for software you choose to license.

disability notice

Section 504 of the Rehabilitation Act of 1973 and the Americans with Disabilities Act Amendments Act (ADAAA) covers issues relating to disability and accommodations. If a student has questions or needs an accommodation in the classroom (all medical information is treated confidentially), contact:

Student Accessibility Services (SAS), Corbett Center, Rm. 208
sas@nmsu.edu; (575) 646-6840; website: http://sas.nmsu.edu.

discrimination policy

NMSU policy prohibits discrimination on the basis of age, ancestry, color, disability, gender identity, genetic information, national origin, race, religion, retaliation, serious medical condition, sex, sexual orientation, spousal affiliation and protected veterans status.

Furthermore, Title IX prohibits sex discrimination to include sexual misconduct: sexual violence (sexual assault, rape), sexual harassment and retaliation.

Office of Institutional Equity (OIE), O'Loughlin House, 1130 University Ave.
equity@nmsu.edu; (575) 646-3635; website: http://www.nmsu.edu/~eeo.