

California State University, San Bernardino

**CSUSB ScholarWorks**

---

Q2S Enhancing Pedagogy

---

Winter 2-27-2020

## **A Mock Software Company: For Teaching Software Engineering (CSE 455) Course**

Arturo Concepcion  
concep@csusb.edu

Follow this and additional works at: <https://scholarworks.lib.csusb.edu/q2sep>



Part of the [Computer and Systems Architecture Commons](#)

---

### **Recommended Citation**

Concepcion, Arturo, "A Mock Software Company: For Teaching Software Engineering (CSE 455) Course" (2020). *Q2S Enhancing Pedagogy*. 112.

<https://scholarworks.lib.csusb.edu/q2sep/112>

This Course Outline / Syllabus is brought to you for free and open access by CSUSB ScholarWorks. It has been accepted for inclusion in Q2S Enhancing Pedagogy by an authorized administrator of CSUSB ScholarWorks. For more information, please contact [scholarworks@csusb.edu](mailto:scholarworks@csusb.edu).

**SCHOOL OF COMPUTER SCIENCE & ENGINEERING**  
**CALIFORNIA STATE UNIVERSITY, SAN BERNERADINO**  
Spring Term 2019

**Course No.** : CSE 455  
**Course Title** : Software Engineering  
**Prerequisite** : CSE 330 (*Data Structures*)  
**Units** : 4 units (3 hours lecture, 3 hours laboratory)  
**Meetings** : 8:30-9:45 am CS 122 MW (lecture)  
10:00- 12:45 pm JB 359 M (lab)  
10:00- 12:45 pm JB 359 W (lab)

**Instructor/Office/Phone/E-mail/Office Hours:**

Dr. A.I. Concepcion  
JB343  
Voice: 909.537.5330  
[concep@csusb.edu](mailto:concep@csusb.edu)  
By appointment only

**Teaching Assistants/E-mail/Office Hours:**

Chaz Hampton, [hampc301@coyote.csusb.edu](mailto:hampc301@coyote.csusb.edu), Mon, 3 :45 – 5 :45 pm.  
Sagar Patel, [sagarp@coyote.csusb.edu](mailto:sagarp@coyote.csusb.edu), Tue, Thu, 2 :15 – 3 :15 pm.

**Objectives:**

Computer hardware productivity continues to increase by leaps and bounds, while software productivity seems to be barely holding its own. Software projects continue to grind out production-engineered code at the same old rate of 30 to 40 delivered LOC per man-hour.

Software engineering encompasses five areas of studies: models of software systems, methods of software development, management of software projects, architecture & design, and analysis of software artifacts (Software Engineering Institute, Carnegie Mellon University), see <[mse.isri.cmu.edu/software-engineering](http://mse.isri.cmu.edu/software-engineering)>.

The general objective of the course is to gain practical and realistic experience in the design, development, and maintenance of software.

The specific objectives of the course are:

- (a) To be able to write software requirements specification (SRS), software quality assurance plan (SQAP), and software project management plan (SPMP) documents.
- (b) To be able to derive the software design from the SRS and learn an OOA/D methodology.
- (c) To be able to apply verification and validation techniques for software testing.
- (d) To be able to perform software maintenance.
- (e) To learn the techniques of team development and processes.
- (f) To understand the use Agile software development, which is iterative

and incremental development.

- (g) To distinguish the difference between the following terms: software engineering, software life-cycle, and software process.

**Text:**

-- B. Bruegge & A. Dutoit, "Object-Oriented Software Engineering: Using UML, Patterns, and Java," 3<sup>rd</sup> Edition, Prentice Hall, 2010.

-- M. Fowler, "UML Distilled: A Brief Guide to the Standard Object Modeling Language," 3rd Ed., Addison-Wesley, 2004.

**References:**

-- I. Sommerville, "Software Engineering," 10<sup>th</sup> Ed., Pearson Publishing, 2016.

**Requirements:**

For the lecture, the whole class will learn the principles and best practices of software engineering. What they'll learn in the lecture will directly apply to the software projects for the lab part of the class.

For the lab, the class will have mobile app projects, which will be developed for the spring term. The class will be organized into teams, one for each project. Each team will have a management team consisting of a project manager and an assistant project manager. The rest will be the development team. There will be a QA team in-charge of testing the prototypes developed by each team. The QA team will be led by a team lead. The set of deliverables are: SRS, SPMP, SQAP, software architecture, detailed design, unit, integration, and system test plans, maintenance manual, documented source code, and a public exhibition presentation and demonstration in the finals.

The TAs assigned to the class will provide help in the development of mobile app projects. Additional help will be available from the interns of the Mobile App Dev (MAD) team. They will act as consultants and expert advisors on any questions or issues you might have in the development of the mobile app projects assigned to you. Be sure to contact the TAs when you have problems. The MAD team is responsible for mobile apps both developed for the campus and clients with commercially viable apps. For more information, see the MAD homepage at <https://mobileappdev.academic.csusb.edu>,

UML (Unified Modeling Language) notations will be used throughout the software design of your mobile app projects. You will need to learn how to use DIA in order to draw the diagrams of the object-oriented analysis and design (OOA/D) methodology.

Each team will be graded according to the deliverables and their being submitted and stored on-time on the development repository. Since the software project is to be developed and maintained, the teams must apply software engineering concepts, methods, and techniques learned from the lectures.

The following are the details of your grade for the course:

Deliverables	Management Team	Development Team
SRS	20%	20%
SPMP, SQAP, & Architecture	20%	
Detailed Design		20%
Integration & System Test Plans	10%	
Unit Test Plan		10%
Maintenance Manual	5%	
Documented Source Code		5%
Tutorial Exercises	5%	5%
Attendance	10%	10%
Instructor & Peer Evaluation	10%	10%
Testing Prototype 1 & 2	20%	20%
Total	100%	100%

The lecture hours will be used to learn the principles and best practices of a software engineer while the laboratory hours will be used mainly as meeting and development times for the teams. The actual work will be done outside of the class hours, which is estimated to be about 4-6 hours per week.

The course involves several reading assignments and references other materials. You will be responsible for reading them.

### **Grading:**

<i>Final Average</i>	<i>Grade</i>
<i>94 and above</i>	<i>A</i>
<i>90-93.9</i>	<i>A-</i>
<i>87-89.9</i>	<i>B+</i>
<i>84-86.9</i>	<i>B</i>
<i>80-83.9</i>	<i>B-</i>
<i>77-79.9</i>	<i>C+</i>
<i>74-76.9</i>	<i>C</i>
<i>70-73.9</i>	<i>C-</i>
<i>67-69.9</i>	<i>D+</i>
<i>64-66.9</i>	<i>D</i>
<i>60-63.9</i>	<i>D-</i>
<i>Below 59.9</i>	<i>F</i>

### **Students with Disabilities:**

If you are in need of an accommodation for a disability in order to participate in this class, please contact Services to Students with Disabilities at UH-183, 909.537.5238.

### **Papers Published for CSE 455 Software Engineering Course (a little bit of bragging):**

[1] A.I. Concepcion and B. Singh, "Modified Personal Software Process (mPSP: An Experiment in Teaching Software Engineering," *In Proc. of the 8<sup>th</sup> International Conference on Computer Science Education: Innovation & Technology (CSEIT 2017)*, 9 – 10 Oct 2017.

[2] A.I. Concepcion, S. Lin, T. Chiang, P. O'Connor, M. Korcha, and D. Shumpert, "A Faculty/Staff/Student Team for Collaboration in Developing Mobile Applications in the Software Engineering Course," *In Proc. of the 6<sup>th</sup> International Conference on Computer Supported Education (CSEDU'14)*, 1 – 3 Apr 2014.

[3] W. James, P. Lucas, J. O'Connor, and A.I. Concepcion "Re-Engineering the AlgorithmA Project for Long-Term Maintenance," *In Proc. of the 23<sup>rd</sup> ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA '08)*, 19-23 Oct 2008.

[4] A.I. Concepcion, D. DeMelo, and D. Hollingsworth, "ReMoTe: A Software Process Management Tool," *In Proceedings of the 22<sup>nd</sup> International Conference on Computers and Their Applications (CATA-2007)*, Honolulu, Hawaii, Mar 2007.

[5] A.I. Concepcion, "Recursively Estimating Multi-threaded Observation Technology Enterprise (ReMoTe)," *Presentation at the World's Best Technologies (WBT 2007) Conference*, Arlington, Texas, 15-16 May 2007.

[6] A.I. Concepcion, M. Bernstein, K. Fitzgerald, and J. Macdonell, "A Ten-Week Mock Software Company," *In Proceedings of the 36<sup>th</sup> ACM SIGCSE Symposium*, St. Louis, MO, Mar 2005.

[7] A.I. Concepcion, N. Leach, and A. Knight, "AlgorithmA 99: An Experiment in Reusability and Component-Based Software Engineering," *In Proceedings of the 31<sup>st</sup> ACM SIGCSE Symposium*, Austin, TX, Mar 2000.

[8] A.I. Concepcion, L. Cummins, E. Moran, and M. Do, "AlgorithmA 98: An Algorithm Animation Project," *In Proceedings of the 30<sup>th</sup> ACM SIGCSE Symposium*, New Orleans, Louisiana, Mar 1999.

[9] A.I. Concepcion, S. Lin, and S. Simon, "The RMT (Recursive Multi-Threaded) Tool: A Computer Aided Software Engineering Tool for Monitoring and Predicting Software Development Progress" *In Proc. of the 21<sup>st</sup> International Conference on Software Engineering*, Los Angeles, CA, May 1999.

[10] A.I. Concepcion, "Using an Object-Oriented Software Life-Cycle Model in the Software Engineering Course," *In Proceedings of the 29<sup>th</sup> ACM SIGCSE Symposium*, Atlanta, GA, Feb 1998.

**Course Outline:**

WEEK	TOPICS	DELIVERABLES
01 Apr	<i>Chap. 1: Introduction to Software Engineering</i>  <i>Chap. 2: Modeling with UML</i>	
08 Apr	<i>Chap. 3: Project Organization and Communication</i>  <i>Chap. 4: Requirements Elicitation</i>	
15 Apr	<i>Chap. 5: Analysis</i>	SRS
22 Apr	<i>Chap. 5: {continuation}</i>	SPMP, SQAP
29 Apr	<i>Chap. 6: Systems Design: Decomposing the System</i>	Revised SRS, SPMP, And SQAP
06 May	<i>Chap. 6: {continuation}</i>	Architecture
13 May	<i>Chap. 8: Object Design: Reusing Design Patterns</i>	
20 May	<i>Chap. 8: {continuation}</i>	
27 May	<i>Chap. 11: Testing</i>	
03 Jun	<i>Chap. 11: {continuation}</i>	
10 Jun	<i>Finals Day, 12 Jun (Wed)</i>	

**Laboratory Outline:**

WEEK	ACTIVITIES	DELIVERABLES
01 Apr	<i>Survey of Skills &amp; Organization</i>	
08 Apr	<i>Training &amp; Tutorials</i>	
15 Apr	<i>Training &amp; Tutorials</i>	
22 Apr	<i>Launch of Prototype #1 Development</i>	
29 Apr	<i>Meetings and Development</i>	
06 May	<i>Meetings and Development</i>	
13 May	<i>QA Testing</i>	Deliver Prototype #1
20 May	<i>Launch of Prototype #2 Development</i>	
27 May	<i>Meetings and Development</i>	
03 Jun	<i>QA Testing</i>	
10 Jun	<i>Finals Day, 12 Jun (Wed)</i>	Deliver Prototype # 2 Exhibit Presentation & Demo

**Student's copy, keep.**

By affixing my signature below, I have fully read the course syllabus for CSE 455 (Spring 2019) and I understand and accept the requirements of the course.

---

Printed Name

---

Signature and Date



**Detach and return to instructor.**

By affixing my signature below, I have fully read the course syllabus for CSE 455 (Spring 2019) and I understand and accept the requirements of the course.

---

Printed Name

---

Signature and Date

