De La Salle University
College of Computer Studies
Software Technology Department

# COURSE SYLLABUS

## SPSWENG / Software Engineering (3 units)

College / Department : College of Computer Studies / Software Technology
Prerequisite : OBJECTP, INTRODB, SWDESPA
Prerequisite to : Practicum
Class Days & Time / Room : S17A TH 1100-1230 / A911
S18A MW 1100-1230 / A911
S18B MW 1100-1230 / G212
S19A MW 915-1045 / G211
S19B MW 1245-1415 / G210
X23 MW 915-1045 / MRW403

Professor : Briane Paul Samson (briane.samson@dlsu.edu.ph)
Ethel Ong (ethel.ong@delasalle.ph)
Nathalie Rose Lim-Cheng (nats.lim@delasalle.ph)
Lance Alcabasa (lance.alcabasa@dlsu.edu.ph)
Ryan Dimaunahan (dimaunahan.ryan@gmail.com)

Consultation Hours : (tentatively) TH 0930 - 1100
Course Site / Group : edmodo.com Group Code: 2cyrkv

---

**COURSE DESCRIPTION:**

This is an introduction to the fundamental concepts, principles and techniques involved in the primary activities of software development – requirements elicitation and specification, design, implementation, and testing. Major emphasis are on the various approaches to requirements analysis and review activities, methods for modeling software systems, architectural design and detailed design using object-oriented techniques, and principles of user-centered design, all of which are necessary for an effective and efficient production of software systems.

Through team projects involving local firms, students gain first-hand experience in performing all the aforementioned software activities following the unified process (UP) model of software development. Agile methodologies are also infused into the course as well to provide the additional perspective to the students regarding this area. In addition, this course also provides an introduction to project management in software engineering and use.

**LEARNING OUTCOMES (LO)[1]:**

Upon completion of the course, students will be able to:

LO1 : Formulate a requirement specification based on the analysis of customer and user's needs to reduce ambiguity, and improve the consistency and completeness of a development team's vision of the intended software solution. [CCT, TCPL]

LO2 : Select and apply appropriate techniques, modern technologies, and practices in requirements engineering,

[1] The learning outcomes are defined with the Expected Lasallian Graduate Attributes (ELGA) as instituted by the University in mind. These attributes are Critical and Creative Thinker (CCT); Effective Communicator (EF); Reflective Lifelong Learner (RLL); Technically Proficient and Competent Professionals and Leaders (TCPL); and Service-Driven, Ethical and Socially Responsible Citizen (SEC)

software design and construction, and software verification and validation. [CCT, RLL, TCPL]

LO3 : Create a software solution that is functional, reliable, efficient, usable and maintainable. [CCT, SEC, RLL, TCPL]

LO4 : Critically evaluate and reflect on own assumptions, thinking and values related to software development in order to plan and engage in independent learning to advance their professional development. [CCT, RLL]

LO5 : Function effectively and responsibly as a Lasallian individual, in contributing positively to the success of a software development project through Christian leadership practices and collaborative processes. [SEC, EF]

**GRADING SYSTEM:**

To pass this course, a student must earn **at least 60%** through the course requirements. Grade breakdown as well as Grade Point Equivalence, are given below:

| | |
|---|---|
| Role Accomplishment | 2.0 |
| Reflection | 1.0 |
| Project Sign Off | 0.5 |
| Exceptional Software Solution | 0.5 |
| | 4.0 |

**LEARNING PLAN:**

| Topic | Week | Learning Activities / Topics |
|---|---|---|
| Introduction | 0 | • Introductions<br>• Professional Developer vs Amateur Developer |
| Requirements Engineering | 1 - 3 | • Gathering vs Elicitation<br>• Problem Analysis<br>• Specification<br>• Verification<br>• Project Planning |
| Week 3b: Start of Iteration 1 | | |
| Software Design | 3 | • Class and ER Diagrams<br>• Software Design Principles |
| Software Construction | 4 | • Key Construction Decisions<br>• Collaborative Construction<br>• Integration |
| Software Verification and Validation | 5 - 7 | • Principles of Testing<br>• Testing Processes and Types<br>• Test Case Design<br>• Defect Reporting and Analysis |
| Week 7b: Iteration 1 Demo | | |
| Reflection on Iteration 1 | 8 | |
| Week 8b: Start of Iteration 2 | | |
| Usability Design | 8 - 9 | |
| Professional Software Engineering | 10 - 11 | • SDLC<br>• Mythical Man Month<br>• Software Ethics |
| Week 12: Iteration 2 Demo | | |

**FINAL COURSE OUTPUT:**

As evidence of attaining the above learning outcomes, the student is required to do and submit the following during the indicated dates of the term.

| LEARNING OUTCOME | REQUIRED OUTPUTS |
|---|---|
| LO1 | 1. Software Requirements Specifications<br>2. User Interface<br>3. Test Plan<br>4. Defects Reports<br>5. Project Plan |
| LO2 | |
| LO3 | |
| LO4 | |
| LO5 | |

**OTHER REQUIREMENTS AND ASSESSMENTS:**

Aside from the final output, the student will be assessed at other times during the term by the following:

- Unit Homework
- Project Supporting Documents
- Online Quiz

**GENERAL COURSE POLICIES**

- Accessing Facebook, Twitter and other forms of social network, as well as unrelated sites is STRICTLY prohibited in the lab. Anyone caught doing so will result to deductions from the closest departmental exam.
- Not following instructions in the requirements for submission may result to non-acceptance of the submission (and thus, it is considered a missed exercise/submission)
- The student who was absent in previous meeting/s is expected to catch up on the missed discussion himself (by asking his peers what he has missed).
- Only valid and verifiable reasons for missed examinations will be considered for make-up.
- Comply with Section 4.13 Policy on Academic Honesty of the DLSU Student Handbook. (Link: http://www.dlsu.edu.ph/offices/mco/publications/student_handbook/default.asp)

**MATERIALS & REFERENCES:**


- Brooks, F.P. (1986). *No Silver Bullet – Essence and Accident in Software Engineering"*. Proceedings of the IFIP Tenth World Computing Conference: 1069 - 1076
- Brooks, F.P. (1995). *The Mythical Man-Month: Essays on Software Engineering (Anniv. ed.)*. Massachusetts: Addison-Wesley.
- Software Engineering Code of Ethics and Professional Practice. http://www.acm.org/about/se-code
- Code Complete by Steve McConnell. Microsoft Press; 2nd edition (August 31, 2004)
- The Art of Software Testing by Glenford J. Myers, Corey Sandler & Tom Badgett (November 2011)