



International University, VNU-HCMC

School of Computer Science and Engineering

Software Engineering - IT076IU

Instructor: Nguyen Thi Thuy Loan

nttloan@hcmiu.edu.vn, nthithuyloan@gmail.com

<https://nttloan.wordpress.com/>

1



International University, VNU-HCMC

Course Website

- Blackboard IU
- Please check frequently for updates!

Nguyen Thi Thuy Loan, PhD

2

2



Lectures

- Theories (15)
Dr. Nguyen Thi Thuy Loan
- Labs (8)
Dr. Nguyen Thi Thuy Loan and TA



Instructor

- Nguyen Thi Thuy Loan (Loan Nguyen)
 - nttloan@hcmiu.edu.vn
 - <http://nttloan.wordpress.com/>
 - office: O1. 610
- About myself
 - PhD in CS, Wroclaw, Poland. 2015
 - Postdoc: University of Warsaw, 2017
 - Assoc. Prof. IT, 2020
 - Joined IU in December 2018
 - Research interests:
 - Data Mining and Knowledge Discovery
 - Class Association rules
 - Pattern mining



Specific goals for the course

- Apply the principles and methods of software engineering in practice;
- Apply critical and analytic thinking to the planning of the software development process;
- Apply critical and analytic thinking to the execution and evaluation of the software development process;
- Use automated tools to support the software development process;
- Demonstrate creative thinking in the design of software solutions;
- Communicate effectively to diverse audiences;
- Apply teamwork skills in a software engineering project.



Learning outcomes

- Apply the principles and methods of software engineering in practice
- Apply critical and analytic thinking to the planning of the software development process
- Apply critical and analytic thinking to the execution and evaluation of the software development process



Requirements

- Students are expected to attend every class section on time and prepared.
- Students are expected to spend *time outside of class* to complete assignments.
- If you cannot complete an assignment, contact the instructor (BEFORE THE DUE DATE)
- Tests are to be taken at the day and time scheduled. DO NOT copy from another students.



Logistics

- Homework submission: Blackboard IU
 - All enrolled students are already there
- Discussion forum: Blackboard IU
 - All enrolled students are already there
 - Send me an email if you have not received an email from Blackboard IU
- Lecture slides will be uploaded before the class
 - It will be updated after the class



Grading

- Midterm Test: 25%
- Lab + Project Assignment: 30%
- Final Exam: 40%
- Class participation: 10%
- Bonus: 5%



Grading Strategy

- Relative grading
 - The actual grade distribution at the end will depend on the performance of the entire class on all the components.
 - Topper of the class gets 100 points irrespective of the number, and only “above expectation” performances get 100 points.
 - No fixed lowest grade or grade distribution.
 - Everyone can get good grade by working hard!



Schedule - Important Dates

- Week 9 or 10: Midterm Exam (by university schedule)
- Weeks 15, 16, and 17: Presentation
- Weeks 19 and 20: Final Exam (by university schedule)

These times are fixed. If you miss a test, there is no opportunity to retake it.



Exams

- Midterm test - Week 9, 10: It will cover the materials from lectures and sections before that day.
- Open a notebook that is written by hand.
- Total weight: 25%
- Exams will test your understanding of the material
- Final exam is comprehensive
 - would include every lecture up to the end.



Projects/ Assignments

- 20% weight
- In groups of 6 to 8 people
 - You can look for group members through Blackboard IU by announcing your general area of interest or if you have a problem in mind
 - Each group member should do approx. equal work
- Project topics: will be posted on Blackboard IU, but we expect you to be creative with a new idea.



Choosing a project

Client

- A client can be any person expect yourself (e.g., a member of a department, a local company, or other external organization, a member of faculty of faculty or staff,...)
- The client should have a firm intention to use the software in production. Aim for a minimum of a three-year production life with many users.



Choosing a project

Special types of client

- Remote clients
- Student organizations
- Entrepreneurship projects



Choosing a project and forming a team

In selecting a project, think broadly

- The project can be an application, system software, or even a toolkit. Software engineering covers everything from smartphones to supercomputers.

The only conditions are that there must be a real client and real users.

- You are encouraged to identify your own project.
- Some potential projects and clients will be suggested by the course team.

At the beginning of the first few classes, there will be time to describe possible projects or to announce that you are setting up a team.



Milestones

The projects are divided into three parts, each of which ends in a milestone.

The first milestone is a feasibility report.

For the second milestone, the team makes a presentation and submits a progress report to the client and the course team.

At the third milestone, the team demonstrates the working software and makes a presentation to the client and the course team, followed by a final report and handover of the completed project to the client.



Milestones

These are group projects, but you will also be rewarded individually for special contributions to the project, or failure to provide a fair share of the effort.

See the Assignments folders pages on the website for more information.



Project Deliverables

1. Project proposal (around 2 weeks, 1-3 pages)
 - problem selection is part of the project
 - 3 weeks from now
 - but start asap, look for problems, do related work study, find an interesting question, let me know your initial thoughts, all by the deadline
2. Midterm progress report (around 4 weeks, 3-5 pages)
3. Final project report (around 10 weeks, more than 10 pages)
4. A final 10-15 mins project presentation and/or demonstration (in the last 1-3 classes)



Project Evaluation Criteria

Scale of 110:

1. Well-motivated? 10
 2. Novel? 10
 3. Comprehensive related work survey? 10
 4. Quality of writing? 10
 - should reflect all other factors too except class presentation
 5. Class presentation/demo? 25
 - should reflect all other factors too except writing
 6. Technical contributions? 45
 - ERD/ Relational Database Schema/ Database/ Program/Testing/Evaluating, etc.
- Evaluating others' projects during the project presentation:
30% (total)



Overview

Every project includes all aspects of software development:

- feasibility study
- requirements
- system and program design
- coding
- reliability and testing
- delivery
- documentation for future maintenance
- software license



Sprint

In agile terminology, a sprint is a fixed period of time during which a team completes part of a software project.

- Every sprint ends with code that is ready to put into production
- A typical sprint might have a team of 6 to 8 people working for 2 to 4 weeks.
- If not released during the sprint, the software should be ready for release, or for integration into a large system.
- It should be fully tested, with documentation for maintenance.



Time box

A time box is a set period of time during which a development team completes part of a software project.

- Time: the project must be completed within one semester.
- Resources: the team size is fix (between 6 and 8 people)
- Scope: the scope must be chosen to fix within the time and resources available
- The scope depends on the experience of the team and the complexity of the project. The project team must agree a project scope with the client.
- If the project runs into difficulties (e.g., time delays), the scope has to be adjusted.



An agile sprint

Within the time box, the project will complete an agile sprint. This can be:

A complete system

- By the end of the time box, the system is either in operation, or fully tested and ready to be installed by the client.

Part of a larger system

- The project will create a completed section of a larger system. It will deliver code, fully tested, and documented.



Team organization

A team of 6 to 8 people cannot function without organization.

Every project should have:

- Regular meetings with the client (at least once a week).
- Regular team meetings.
- A project plan, kept up to date (e.g., a Gantt chart).
- A project management system for code and documentation, chosen in conjunction with the client (e.g., GitHub).



Within the time box

- The project is large enough that you will need a systematic process for developing the software.

Most project will use one of the following processes:

- Iterative refinement
- Modified waterfall model

Some project may use:

- An agile process with a sequence of short sprints



Organizing the project

The projects are team projects

Everybody should aim to contribute to every aspect of the project:

- Meeting with the client
- Taking notes at meetings
- Requirements and design
- Presenting at the presentations (required)
- Coding and testing
- Project management
- Writing reports and documentation
- etc.



Leadership

- Members of the team are jointly responsible for the success of the project.

Leadership

- There is no team leader. Everybody shares in organizing the project.



Project management

Members of the team are jointly responsible for project management.

- It is often useful to have a project manager, who manages the schedule, but the assignment of tasks is a shared activity.
- Suggestion: Some teams change project manager at regular intervals, e.g., every two weeks.
- Small groups: It is often useful to divided the team into small groups for some activities, but everybody should be prepared to help with the work of every group.
- Special expertise: If a team member has special expertise, make use of it, but try to get other members of the team involved.



Class Participation

- 5% weight
- Includes
 - Participation in class (Q/A)
 - Evaluating others' projects during the project presentation
- In general, actively participate in the class!



Reading Material

- Ian Sommerville Software Engineering, 9th Edition, 2011
- You do not have to buy the books, but it will be good to consult them from time to time
- You should be prepared to do quite a bit of reading from various books and papers
- <https://www.guru99.com/software-engineer-book.html>



Thank you for your attention!