SOFTWARE ENGINEERING

CO3001

COURSE OUTLINE

Anh Nguyen-Duc
Tho Quan-Thanh

WEEK 1

Adapted from https://iansommerville.com/software-engineering-book/slides/

AIMS

- ✓ The goal of this course is to provide undergraduate students with
  - Knowledge (concepts, terms, processes, models)
  - Skill (methods, techniques)
- ✓ for requirement, analysis, design, implementation and testing of software-intensive systems.

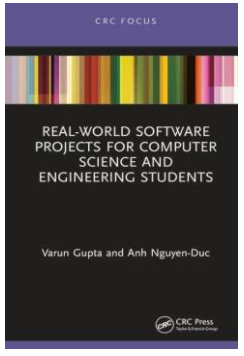
Anh Nguyen-Duc

Email: angu@usn.no

Universitetet i Sørøst-Norge

VIETSE

Software Startups
Global Research Network



✓ Academic experience:

- Assc. Prof. in University of South Eastern Norway
- Dr. in Experimental Software Engineering

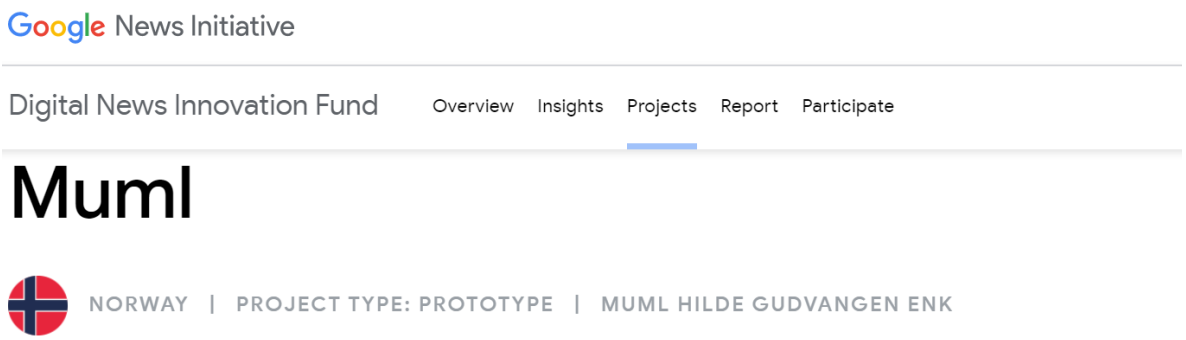
✓ Academic services:

- Vice chair of Software Startup Research Network
- Co-Founder of VietSE – Vietnamese Software Engineering Network
- Steering member NOKOBIT (Norsk konferanse for organisasjoners bruk av informasjonsteknologi)
- Co-chairs of NOKOBIT 2020, ICSOB 2019, IWSS 2018, 2017, PROFES 2016, ICE 2016
- Book author – Fundamentals of Software Startups
- Guest editors & PCs in 20+ SE conferences and journals

✓ Research interests:

- Software Startups
- Lean Startup Education
- Global Software Development
- Security in Internet-of-things

Anh Nguyen-Duc



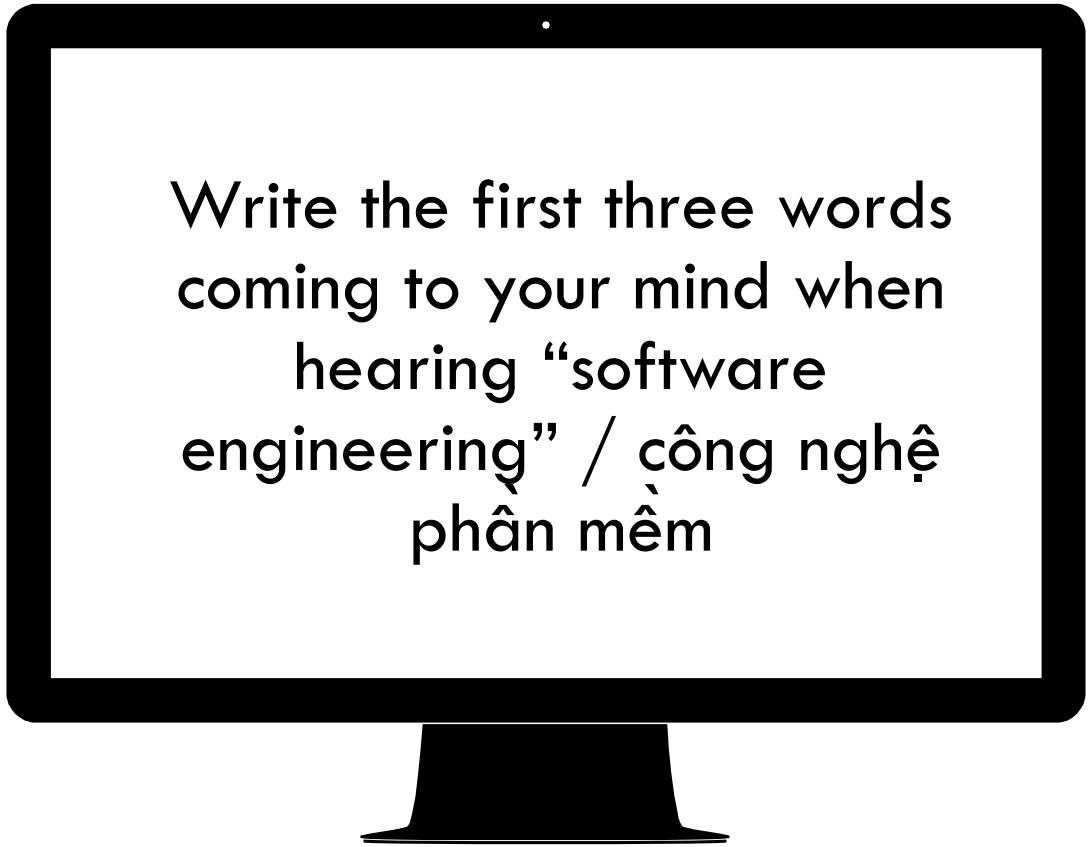
MBITION

VNNOR

COMARTEK

- ✓ **Teaching:**
  - WEB6101N - Web Development and Human User Interaction (2017-2020), USN
  - TDT4920 - Customer Driven Projects (2011-2020), NTNU
  - Green IT (expected Autumn 2021), USN
  - MIS405 - Managing IT Projects (Spring 2020,2021), USN
  - Practical Project Management (Spring 2018-2020), USN
  - SYS1000B - Software Engineering (Spring 2018), USN
  - TDT4140 - Software Engineering, (2016-2017), NTNU
- ✓ **Entrepreneur:**
  - 2015: Muml – Hyper-local news platform
  - 2019: VVN – eKYC solution
- ✓ **IT & Startup consultancy**

Menti.com


Write the first three words coming to your mind when hearing “software engineering” / công nghệ phần mềm

OUTLINE

- ✓ An introductory course to the field of software engineering.
- ✓ The goal is to provide techniques, methods and processes for the development of software-intensive systems.
- ✓ Help getting familiar with software engineering activities: requirements elicitation, software specification, architectural & detailed design using design patterns.
- ✓ Also cover software implementation and software testing
- ✓ Use extensively the UML modeling language

STUDENT LEARNING OUTCOMES

Knowledge:

- ✓ L.O.1. Understand that software systems need to be developed methodologically and professionally;
- ✓ L.O.2. Elicit requirements & perform architectural design;

Competence:

- ✓ L.O.3. Cary out detailed design, coding, testing;
- ✓ L.O.4. Use the UML language effectively in software development.

STUDENT LEARNING OUTCOMES

No.	Course learning outcomes
L.O.1	Understand that software systems need to be developed methodologically and professionally;
	L.O.1.1 Understand principles and concepts of software engineering L.O.1.2 Understand methods and techniques of software engineering
L.O.2	Elicit requirements & perform architectural design
	L.O.2.1 Requirements elicitation L.O.2.2 Architectural design
L.O.3	Cary out detailed design, coding, testing
	L.O.3.1 Detailed design L.O.3.2 Coding L.O.3.3 Testing
L.O.4	Use the UML language effectively in software development
	L.O.4.1 UML use-case diagram L.O.4.2 UML sequence diagram L.O.4.3 UML class diagram L.O.4.4 UML activity diagram (or UML state-chart diagram)

TEXTBOOK/REFERENCE BOOK

- ✓ [1] Ian Sommerville (2015), Software Engineering (10th ed.), ISBN 978-0133943030, Pearson
- ✓ [2] G. Booch, J. Rumbaugh, I. Jacobson (1998), The Unified Modeling Language User Guide, Addison-Wesley.
- ✓ [3] E.J. Braude (2001), Software Engineering: An Object-Oriented Perspective, ISBN 978-0-471-32208-5, John Wiley.
- ✓ [4] Gamma, E., Helm, R., Johnson, R., Vlissides, J., Design Patterns: Elements of Reusable Object-Oriented Software, ISBN 978-0201633610, AddisonWesley Professional (Nov. 10, 1994)
- ✓ [5] State-of-the-art articles on Software Engineering

TEACHING ACTIVITIES

- ✓ Flipped classroom: read materials before the lectures
- ✓ Rehearsal section
- ✓ In-class activities
- ✓ Student presentation
- ✓ Capstone projects
- ✓ Final exam

EVALUATION

- ✓ In-class/online activities/quizzes: 10%
- ✓ Student presentation: 10%
- ✓ Group-based project: 30%
- ✓ Final writing exam: 50%

QUIZZES

- ✓ Online, every week. Two categories:
  - Before the lecture:
    - Read slides, prepare yourself and take the quizzes before the lecture.
    - The questions are simple, just review the content of the coming lecture.
  - After the lecture (named "... - advanced"):
    - Summarize the content of the lecture, give to cases for you to apply the knowledge of the lecture.
    - The questions are some more advanced. Take your time to reflect the lecture before taking the quizzes.
- ✓ Duration limit of 10', valid in a few days.
  - You can take a quiz twice to get the highest score.
- ✓ Please check the course website regularly

PROJECT

TENTATIVE SCHEDULE

Wk	Tuần DL 2023	Topic
1	35	Ch1. Introduction
2	36	Ch2. Software process
3	37	Ch3. Req. engineering
4	38	Ch4. Req. engineering (cont.)
5	39	Ch5. Introduction to OPP (đọc thêm). Ch6. System modeling
6	40	Ch5. Introduction to OPP (đọc thêm). Ch6. System modeling (cont.)
7	41	Review Project #1
	42	Midterm break
8	43	Ch7. Architecture design
9	44	Review Project #2
10	45	Ch8. Design and Implementation
11	46	Ch9. Quality assurance
12	47	Review Project #3
13	48	Ch10. Agile Software Development
14	49	Student presentation (Task 5.2)
15	50	Ch11. Continuous Integration/ Deployment (CI/CD), Ch12. Security in SE

CONTACT

- ✓ Lecturers:
  - Nguyễn Đức Anh (angu@usn.no)
  - Quản Thành Thơ (qttho@hcmut.edu.vn)

- ✓ Course website:
  - http://e-learning.hcmut.edu.vn

REFERENCE SOURCES OF THE SLIDES

- ✓ Slides in this course are adapted mainly from Sommerville 2015 [1]. Some slides are adapted from Braude 2001 [2].
- ✓ Slides of chapter "7.3. More on Implementation" are adapted from Braude 2001 [2].

[1] Ian Sommerville (2015), Software Engineering (10th ed.), ISBN 978-0133943030, Pearson
https://iansommerville.com/software-engineering-book/slides
[2] E.J. Braude (2001), Software Engineering: An Object-Oriented Perspective, ISBN 978-0-471-32208-5, John Wiley.