



Introduction to Software Engineering Administrativa

CSE115a – Winter 2025
Richard Jullig



The Four Questions



Video: <https://www.youtube.com/watch?v=TSIajKGHZRk>

- A
- B
- C
- D



Acknowledgements

- The lecture notes for this course are significantly inspired by
 - Linda Werner, UCSC
 - CMPS115 lecture notes
 - Construx courses
 - Walter Tichy, Karlsruhe Institute of Technology
 - with Mathias Landhäußer and Andreas Biersch
 - Softwaretechnik I lecture notes
 - Other sources as noted



Software Engineering – Short Definition

Software Engineering:

body of knowledge concerning the *managerial* and *technical aspects* of software *system* construction and maintenance

This course provides essential knowledge to software **engineers**, i.e., software **professionals** that work in **teams** to construct, deploy, and maintain **software systems**.



Teaching Staff

■ *Instructor*

Richard Jullig, PhD
Continuing Lecturer

■ *Contact*

- Office: E2-247B
- email: rjullig@ucsc.edu

■ *Office hours*

- Monday/Thursday 10:00am-11:00am PST
- “Office” Zoom link:
<https://ucsc.zoom.us/my/jullig.virtual.office?pwd=RjZEBG9PaDdsSGxlc1pyNC9zYXJLdz09>



Teaching Staff

■ *Teaching Assistants/Project Coaches*

- | | |
|----------------------|-------------------|
| ■ Scott Davis | okdogulu@ucsc.edu |
| ■ Mathesh Thirumalai | mthiruma@ucsc.edu |
| ■ Bruce Bai | brbai@ucsc.edu |
| ■ Eric Xie | ejxie@ucsc.edu |



Course material

■ Course material

- available on Canvas
- Lecture notes
- Textbooks (posted under Canvas>Pages>Reading Material – Software Engineering)
 - Bruegge_2012_Object_Oriented_Software_Engineering_Using_UML_Patterns_and_Java_3rd_2012
 - Meyer_Agile_2014
 - ... and much more ...
- Additional reading material
- Read-and-Comment (RAC) assignments (Homework)
- Project resources
- Document and presentation templates
- ...



Course Work

■ Individual work (50%)

- (~20%) 4 Read-and-Comment assignments
 - Read paper
 - Submit written summary/commentary
 - via Crowdgrader (details soon)
 - Peer review 4 submissions
- (~5%) Weekly team status reports (TSR) (Week3 through Week10)
- (~5%) Project reflection essay (at end of course)
- (~20%) 3 tests
- Possibly other reading/viewing assignments

■ Note:

- percentages are subject to (slight) change
- Individual work items will appear as Canvas assignments



Course Work (2)

- **Team** project (50% of grade)
 - 5+/-1 team members
 - Initial project presentation (in class; attendance mandatory)
 - Software development (outside class)
 - using **Scrum** project management (required)
 - Final project presentation (in class; attendance mandatory)
 - Project Review (equivalent to final exam; attendance mandatory)

- **Note:**
 - Team members' project scores will vary depending on contribution
 - To pass the class, both your individual work and your team project work must be minimally satisfactory i.e., at least 50% level

Syllabus and Schedule

- posted on Canvas > Syllabus





Project Site: Grepthink.com

- **Grepthink.com**
 - Site to propose or request to join projects
 - All projects must be posted here
- **Register** at Grepthink right away (as soon as possible)
 - Use your **UCSC user id**
 - Example:
if your UCSC email is asmith@ucsc.edu
then your UCSC user id is *asmith*
 - **Remember your password** (password recovery is broken)
- After registering at Grepthink,
join the class 2025W cse115a
 - Join code: ***jximKlyzJP*** (*/* could be lower case L or upper case i)
 - See also: Canvas > Pages > Course Platforms



Forming project teams

- Use the project site: GrepThink.com
(cf. links on Canvas > Pages > Course Platforms)
 - **Propose** project ideas
 - The person proposing the project usually becomes the **Product Owner (PO)**
 - Be bold, creative: you don't need to have all the answers
 - **Request to join** proposed projects
 - Reach out to PO of project you are interested in joining
 - Explore more than one project
 - Once you join a project,
let other projects (you requested to join) know that you are no longer interested
- External project sponsors
 - Presentations: possibly later this week
- All projects must be posted on Grepthink



Forming project teams: Team Size and Schedule

- Teams should consist of 5 people
 - +/- 1 acceptable (but not desirable)
 - **Note:** larger teams are harder to coordinate
 - Schedule **coordination** is **important** for team **success**
 - Coordination/communication complexity increases with the **square** of team size
 - **Schedule coordination** is **#1 problem** for cse115a teams
- Project planning and coordination
 - Synchronous meetings critical (and therefore required)
 - Ensure schedule feasibility for meetings and work



Forming Teams: with Friends or Strangers?

■ Friends

- Easier to get started with
- May live closer
- Easier to talk to at first
- Easier to get off track
- Harder to confront
 - Not showing up
 - Being late
 - Shoddy work
 - No work

■ Strangers

- May end up being friends
- May live further away
- Harder to talk to at first
- Easier to stay focused
- Harder to understand
 - Not showing up
 - Being late
 - Shoddy work
 - No work



Project Work Space

- Lab space in Basking Engineering
 - Possible rooms: BE-302, BE-316, BE-340A, or BE-340B
 - To be announced
 - Key access granted after enrollment is stable
 - 24/7 access



Forming project teams – KAIST clause

- KAIST students
 - At most two KAIST students in the same project
- **In general:**

Any team may have at most two members with the same native language, unless the native language is English.



What's your experience with group projects?

- Have you've been part of a group project before?
- How did it go?
- In this class: **Team** projects
 - Team: an **organized, collaborative** group

This course may be different (from what you are used to)



- Not another programming course
- Wide range of material
 - technical aspects
 - managerial/organizational/social/human aspects
- Questions/problems/tasks
 - less well defined
 - you will have to define them
- Answers
 - not unique; no single best answer
 - you will have to make trade-offs

This course may be different (from what you are used to) (2)



- Responsibility for your work lies with you
 - even when (you think) no one is checking
 - many different tasks and deadlines to track (your job)
 - just as “in the real world” (outside the university)
- Opportunity to build and exercise your autonomy
 - Intrinsic motivation
 - Proactive commitment
 - Leadership by example
- It's OK if you are not perfect
 - Use “perfect” as a verb, not as an adjective
- The teaching staff are primarily here to help

This course may be different (from what you are used to) (3)



■ Check your mindset

What do I have to do
to get the grade I want

?

(minimum)

The *Homework* Mindset

What can I contribute
to make my team/the
course successful

?

(optimum)

The *My Work* Mindset



Useful Material

- Two TED talks by Barry Schwartz
 - Some overlap between talks (each about 20min.)
 - View at least one
 - See Canvas > Pages > Reading Material - General

- Movie *Invictus*
 - Intro to Rugby, [Scrum](#)
 - Team building and leadership
 - Inspiration and aspiration (and lots of perspiration)
 - Perseverance

This course may be different (from what you are used to) (4)



- Why are you taking this course? Cui bono?

Non scholae sed vitae discimus.

This course may be different (from what you are used to)



- The gift of your attention

I do not have the skill to be clear
for someone who is not attentive.

(Jean-Jacques Rousseau, Social Contract,
Preface of Volume II)

- The best place and time for learning

Here and Now



Questions – Please ask!

- Questions are welcome
 - during and outside of lectures
- Terminology in Software Engineering
 - confusing (even to native English speakers)
 - changing over time
- “Pay” attention, but if you are lost: Ask!
 - especially if you are shy and introverted (like me)
- If you can’t hear me, let me know right away
- If you can hear me but don’t understand me, let me know.



Getting to know each other

- Find two people you don't know
 - At least one other person from another country/state

- Find out
 - Their name
 - Where they are from
 - Something they like about computers or computer science
 - Something they like to do for fun
 - Anything else you're curious about (favorite food, ...)

- Be ready to introduce “your” people to the class