CSE403 ● Software engineering ● sp12

Weeks 1 & 2				
Monday	Tuesday	Wednesday	Thursday	Friday
• Overview • Course plans & expectations	• Tools & tool questions (section)	• Lifecycle & project milestones • KNOW project overview	• No section	*Proposal descriptions & slides by 9:30AM* • Proposal presentations *Project & team preferences by 11PM* Teams announced by 11PM Saturday
• Requirements	Group meetings (JHN 75, 9^{30}-10^{20}AM)	• Teams	Sections (JHN 75, 9^{30}-10^{20}AM)	• TBA

David Notkin ● Kıvanç Muşlu (TA) ● Anton Osobov (TA)

web ● calendar ● discussions ● gradebook


We need mentors for new CSE students!

We still need about **20 CSE** majors to serve as mentors for new students this week. The Welcome Night is this Wed 430pm - 6pm (food included) in the Allen Center Atrium.

Please sign up via catalyst:
https://catalyst.uw.edu/webq/survey/cseadv/162148

Two definitions of SE

SE ≡ software engineering ● SW ≡ software

1. SW from womb to tomb
2. …the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of SW [IEEE]

Strengths and weaknesses of the definitions?

The first student to put an accurate summary on Catalyst GoPost summaries/discussions (after lecture) gets extra credit – maximum three per student during the quarter

Software engineering: engineering?

- SWEBOK – SE Body of Knowledge 2004 (v3 ongoing)
  - SW requirements, SW design, SW construction, SW testing, SW maintenance, SW configuration management, SE management, SE process, SE tools and methods, SW quality, related disciplines,…
  - Fifth of five purposes: "…to provide a basis for certification and licensing of software engineers"
    - Engineers are licensed per-state; only Texas licenses software engineers: 56349 PEs: ~20K CIV, ~8K ELE, 56 SWE (11 from universities)
- "Designing and constructing … works of public utility" <OED>; making software "machines to serve useful purposes in the world" <MICHAEL JACKSON>

An Assessment of Software Engineering Body of Knowledge Efforts: A Report to the ACM Council – Notkin, Gorlick, Shaw (2000)

Software engineering: same as programming?

Extra credit

A standard SE introduction

Software Crisis
- SW projects are too expensive and cancelled too often
- SW quality is appalling
- … it's a crisis! …

Engineering
- Make SE a *real* engineering discipline
- Define strong mathematical basis, standard of practice, etc.
- …

Process
- Define and adhere to a standard lifecycle, methodology, …
- State requirements, design, etc. precisely and rigorously
- …

Appalling quality and cost: examples

- Zune leap year bug – 30MB models failed to boot
- Mars Polar Lander crash
- …

Any other favorite examples?

My SE introduction

Software Crisis
- For the AIDS crisis, we'd like to eliminate AIDS
- For the Cuban missile crisis, try to eliminate missiles in Cuba
- If SW is a crisis, give us another technology like it!

Engineering
- Many of the goals and steps toward SE are reasonable
- Many of the analogies to traditional engineering are flawed
- We need to improve our ability to efficiently produce high-quality SW, but SW has fundamentally different characteristics

Process
- Processes help when consistent with the problem to be solved
- No process/method applies to all problems in all situations
- Many processes and methods, in pursuit of SE, try to take the "soft" out of SW

Why is SW challenging to engineer?

- Discrete nature of software
- Scale and complexity of software – even given abstraction
- Ability to adapt software – and subsequent pressures to do so
- Astonishing demand for software
- Exceedingly rapid changes in the underlying technologies
- Frequent lack of clarity about requirements
- Communication among teams can be difficult
- …

So, what about 403?

- Most of you likely rationally understand the distinctions between programming and software engineering
- Experience, however, shows that few of you are likely to understand the distinctions viscerally
- Thus, our primary vehicle for the course is a group project – groups of about six who take high-level requirements through implementation

- **The overarching intent of the project is to spread this understanding from your brain to your belly**

What's a 10 week project to do?

Can approximate

- Ill-defined requirements
- Customers
- Time-pressure
- Teamwork
- Different team roles
- Control over design
- …

Can't approximate

- Global, distributed teams
- Full womb-to-tomb
- Competitors
- Project cancellation, extensions
- Feedback from real users
- …

Your biggest challenges are to define an appropriate scope for the project and to structure your team, your process, and your product to allow for planned and unplanned adjustments

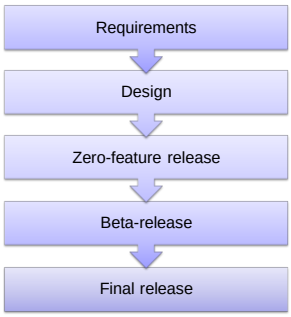
Project selection process

Topics & constraints
KNOW • your own ideas
form your own *proposal* teams of 3+

Proposals
description & slides
due 9^{30}AM Fri then presented/posted

Project & team preferences
due 11^{00}PM Fri; announced by Sat 11^{00}PM

After proposals: milestones



- Where does testing fit in? Documentation? Team structure? Bug database? Etc.?
- Mostly everywhere – these milestones focus on keeping the customer's needs in the forefront
- The milestones, and associated documents, should be the basis for a great portfolio for potential employers

Pitfalls to watch for include…

- A slow start
- Insufficient team meeting time
- Choosing project software solely because you want to learn it
- Ignoring the importance of understanding the domain
- Too much time making non-critical decisions
- Too much time making critical decisions
- "Super-programmers" who try to take over and make it a "mere matter of programming"
- Too much/too little time getting tools to work
- Too much/too little focus on documentation
- Isolating or marginalizing one or more team members
- Assuming nothing will go wrong
- Overly high expectations about what is achievable
- Nothing works unless everything works
- …

Keep your eyes on the prize

- I value a working system that does less over a non-working system that potentially does more
- I value a system that reflects realism over unrealistic conceptual beauty – but this is a tough line to toe
- I value a team that coordinates continuously over occasional "catching up" with each other
- I value a team that surfaces and deals with rather than hides and tries to avoid any difficulties
- I value a team that asks for help when they need it over a team that doesn't

Grading

Individual ● 35%		**Group ● 65%**	
Project proposal	2%	Weekly status reports	5%
Reading summaries/questions	8%	Requirements	10%
Midterm I	10%	Design & planning	10%
Midterm II	10%	Zero-feature release	10%
Class participation	2.5%	Beta-release	10%
Individual retrospective	2.5%	Final release	15%
Extra credit is considered after the basic course grades are assigned		Final presentation	2.5%
		Team retrospective	2.5%

Members of a group get the same grade, except in unusual circumstances

Readings

- A lot has been written about software engineering both from industrial and also from academic perspectives. The intent of the reading assignments is to have you see how experts in the field look at various issues and problems in software engineering
- There will be eight weeks of readings – the first three weeks are posted
- Each week there will be questions to be answered sometimes in a one-page document and sometimes via a Catalyst quiz (due Mondays at 11PM)
- Some readings will require you to login via the lib.washington.edu site

Exams and end of year

- Notkin (June 1-9) & Muşlu ✈ ICSE 2012 Zürich
- No final
- Project presentations May 31 (Th) and June 1 (F)
- Individual and team retrospectives due finals week
- Grades from Switzerland or 40,000'

- Two midterms (in class, open note, open book, closed electronics, closed neighbors) April 25, May 23

Today	Tomorrow	Wednesday	Thursday	Friday
• Overview • Course plans & expectations • **Form project proposal groups ASAP**	• Tools & tool questions (section)	• Lifecycle & project milestones • KNOW project overview	• No section • **Meet with your project proposal groups**	**Proposal descriptions & slides by 9:30AM** **Posted on web ASAP** • Proposal presentations **Project & team preferences by 11PM** Teams announced by 11PM Saturday

Any questions?