

- Search results "software engineering" (12/18/08)
 - Google: ~40,000,000 ["Barack Obama": ~111,000,000]
 - Yahoo: ~563,000,000 ["Barack Obama": ~628,000,000]
- Jobs with "software engineer" in title (12/18/08)
 - Monster.com: 3091
 - Yahoo! hotjobs: 1251 (posted within last 60 days)
 - Seattle Times: 105
 - Google: 80 (in U.S.)
 - Boeing: 51 ("software engineering" job classification)
 - Microsoft (U.S.): 209 "SDE", 137 "SDE in Test"

CSE403: Software Engineering

David Notkin
Winter 2009

But what is software engineering?

- Software engineering is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software [IEEE].
- The systematic application of scientific and technological knowledge, through the medium of sound engineering principles, to the production of computer programs, and to the requirements definition, functional specification, design description, program implementation, and test methods that lead up to this code [answers.com].
- Software engineering is intended to mean the best-practice processes used to create and/or maintain software, whether for groups or individuals, in attempt to address some of the usual haphazard methods that have plagued the software industry [practicalprocess.com, SEVP].
- The design, development, and documentation of software [pcmag.com].
- Software from tomb to tomb.

CSE403 W09

2

What software engineering isn't

- *It is not a mere matter of programming*
- In groups of two or three, take two minutes to add some further reasons for this
 - Building the right software system is as important as building the software system right.
 - Scale causes the problems to qualitatively differ.
 - ...your group's reasons...

Time's up!

CSE403 W09

3

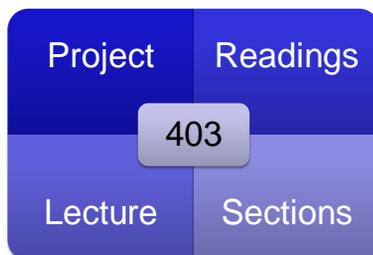
Today

- What, when, why and how
 - Lectures, project, sections, readings
- Project overview
 - Requirements, team information, software, pitfalls
 - Wednesday: team structure, milestones, etc.
- Software engineering introduction

CSE403 W09

4

Our approach



CSE403 W09

5

Our approach: the project

- 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
- The overarching intent of the project is to spread this understanding from your neurons to your viscera

CSE403 W09

6

Our approach: the lectures

- Core material
 - Lifecycle: design, testing, ...
 - Approaches/Methods: information hiding, agile processes, ...
 - Hot topics, research topics, licensing, ethics, ...
- Milestone reviews (one per project group) and perhaps some group meeting time
- The overarching intent of the lectures is to identify software engineering problems and corresponding approaches for which there is evidence of utility in given contexts

CSE403 W09

7

Our approach: the readings

- Two categories of software engineering course books
 - Many largely disconnected approaches to better engineer software
 - Coherent descriptions of a particular approach to engineering software
- Weaknesses for 403: little guidance on how to narrow down a big menu or little guidance on applicability (N.B.: There are reasons I haven't written a book!)
- Instead, various readings – some classic, some "newer"
- The overarching intent of the readings is to use the experience and evidence of others in helping you to better understand the software engineering problem and solution spaces and to develop your own sense of these and related issues

CSE403 W09

8

Our approach: the sections

- Forming groups
- Tool mini-tutorials
- Group meetings
- The overarching intent of the sections is to reduce, as much as possible, the overhead of using a variety of software engineering tools for your project

CSE403 W09

9

Project: budget game tool

- Government budgets are complex, combining political, social and economic policies
- It's hard to think about the core issues because it's hard to understand the budgets and the consequences of potential decisions
- Budget games allow people to make coarse decisions about what programs to reduce or eliminate and which revenue streams to create or increase, showing the consequence of these decisions on the budget
- High-level budget analysis "games" examples include
 - How would you cut the Massachusetts budget by \$12.6 billion?
 - The Budget Game (Texas, NYC)
 - National Budget Simulation United States
- Links on 403 web
- Wiki for sharing other examples across groups

CSE403 W09

10

High level requirements

- Build a budget game for a selected government entity
 - Federal, state, county, city, public university with an annual budget of at least one billion dollars
- Use real, publicly available data: share sources across groups via wiki
- Support multiple levels of granularity for viewing the data and for investigating alternatives (e.g., focused cuts vs. across-the-board cuts)

CSE403 W09

11

Criteria

- I value what the game allows users to do more than how it lets them do it: that is, function dominates user interface.
- I value a working system for lesser function over a non-working system with greater (potential) function.
- I value a system that reflects realistic warts of the selected budget over unrealistic conceptual beauty.
 - Einstein said: "Everything should be made as simple as possible, but not one bit simpler."
 - Although you cannot make it fully realistic, take this notion to heart.

CSE403 W09

12

What's a 10 week project to do?

Not a mere matter of programming!

Can approximate <ul style="list-style-type: none"> • Ill-defined requirements • Time-pressure • Teamwork • Different team roles • Control over design • ... 	Can't approximate <ul style="list-style-type: none"> • Global, distributed teams • Full-worm-to-tomb • Pressure from competitors • 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

CSE403 W09 13

Project groups

- Each group should have 4-6 people
- Form groups primarily on your own; I reserve the right to make adjustments as needed
 - Final groups will be in place by the end of this Thursday's section
 - By 4PM on Wednesday, you must send email to the staff (Cse403-ta@cs.washington.edu) with your group status – even if you're a singleton
 - One email per (sub)group, please
- Experience shows that an inability to find consistent and common group meeting times is a *huge* minus

CSE403 W09 14

Software requested for CSE labs

Unix	cunit • nunit • Eclipse • gcc, gdb • JDK • jsunit • junit • SVN clients	Unix PCs & Attu Cluster
Unix	Ruby on Rails • Bugzilla • wiki • LAMP dev environment	cubist
Windows	Tortoise SVN • JDK • cunit • junit • C++ • eclipse • ASP.NET • jsunit • nunit	Desktop PCs
Windows	ASP.NET dev environment	iisqslrv
	SVN server	project space server

CSE403 W09 15

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
- Extended differences of opinion about alternative 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
- Overly high expectations about what is achievable
- Nothing works unless everything works

CSE403 W09 16

The usual SE introduction

Software Crisis

- Software projects are too expensive and cancelled too often
- Software quality is appalling
- ...

Engineering

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

Process

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

CSE403 W09 17

Your turn: appalling quality and cost

- Zune leap year bug
- Mars Polar Lander crash
- ...

Time's up!

CSE403 W09 18

Notkin's responses

Software Crisis

- For the AIDS crisis, we'd like to eliminate AIDS
- For the Cuban missile crisis, we wanted to eliminate missiles in Cuba
- For the sub-prime loan crisis, we'd like to eliminate the effect of sub-prime loans
- If software is a crisis, give us another technology like it!

Engineering

- Many of the goals and steps toward Software Engineering are reasonable
- Most of the analogies to traditional engineering disciplines are flawed
- We need to improve our ability to efficiently produce high-quality software, but these analogies distract rather than enhance

Process

- Process and methods are good, when they are consistent with the problem to be solved
- No process or method is appropriate in all contexts to help solve all problems
- Many processes and methods, in pursuit of Software Engineering, seem to wish to take the "soft" out of software

CSE403 W09

19

Is it Software Engineering

- *Absolutely*, in the sense of "designing and constructing ... works of public utility" [OED] or that we "make [software] machines to serve useful purposes in the world" [Michael Jackson]
- *Not yet*, in conventional terms as a recognized and licensed profession. SWEBOK (Software Engineering Body of Knowledge) says that the "legitimization of professional authority" indicates that "the knowledge and competence of the professional have been validated by a community of ... peers" and that this "knowledge rests on rational, scientific grounds"
- *May never be*, if viewed as "engineering all kinds of software"

CSE403 W09

20

An alternative introduction

software

- Software is a powerful vehicle for building machines, or parts of machines, that benefit society
- Software's importance has grown wildly over the past 50 years
- Software's malleability is its strength – It can be molded in incredible forms
- Software systems are intertwined in society in amazing ways: this is a testament to the power of software and the ingenuity of software engineers

engineering

- To even better satisfy the needs of society, we must be relentless at improving the properties of the software we build
- We should look to traditional engineering disciplines, as well as to other product-development approaches (such as drug design), for ideas about how to do so
- However, software is at least in most ways a fundamentally different medium – so we should anticipate finding entirely new approaches in this pursuit

CSE403 W09

21

Administrivia

- Grading
 - Two exams: 15% each
 - Project: 70%
- TAs – introduce thyselfes!

CSE403 W09

22

Questions?

CSE403 W09

23