Computer Science
UNIVERSITY OF TORONTO

csc302h: engineering large software systems

matt medland

matt@cs.utoronto.ca

http://www.cs.utoronto.ca/~matt/csc302

about me

- DCS undergrad here a long time ago
- graduate student here…twice!
- mix of academic & industry background
  - approximately 40% / 60% split
- developer, manager, director, consultant, chief software architect
- small startup up to company of 4000+
- mainframe, desktop, web, cloud, mobile, embedded

your TAs

- Ahmed Shah Mashiyat
  - Ph.D. Candidate here @ DCS
- Andrew Danks
  - 4th year undergraduate
  - extensive experience developing large applications at Avid Life Media, Marin Software & Yelp

about the course

- home page:
  - http://www.cs.utoronto.ca/~matt/csc302
  - syllabus & marking scheme
  - course announcements
  - assignments
  - lecture slides will be posted after each session
  - link to forum on piazza
  - etc.

about the course (2)

- textbooks
  - none required.
  - supplemental material provided on course page
  - recommended: *“The Agile Planning Horizon in Professional Software Development”,* by David A. Penny, Ph.D.

about the course (3)

- lectures, tutorials, office hours
  - l: tuesdays & thursdays @ 10am, GB248
  - t: thursdays @ 11am, GB248 (for now)
  - o/h: tuesdays @ 11am, BA4237
- course project
  - work with an existing open source project
  - work in teams of 6 or 7
  - four assignments, one report per team for each
  - peer evaluation

about the course (4)

- build on and apply what you learned in csc301h
  - Scale up to larger projects
- topics chosen from:
  - advanced UML, patterns, architecture, refactoring, software evolution, reverse eng., SDLC models, project mgmt. (planning, risks, estimation, prioritization), requirements analysis, v&v, testing, quality, managing a team, formal methods, etc.
  - some topics will get more attention than others

Computer Science
UNIVERSITY OF TORONTO

about the course (5)

- 2014 professional lecture series
  - invite professionals from industry to talk about topics relevant to the course
  - gain multiple perspectives from the real practitioners!
  - gain some industry contacts!

assessment

- four assignments 10% each, done in teams
  - reverse engineering & design discovery
  - analysis of change requests: development plan
  - requirements analysis & test plan
  - implementation of plan & post-mortem
- two tests: midterm 15%, final exam 35%
- participation & peer evaluation 10%

course policies

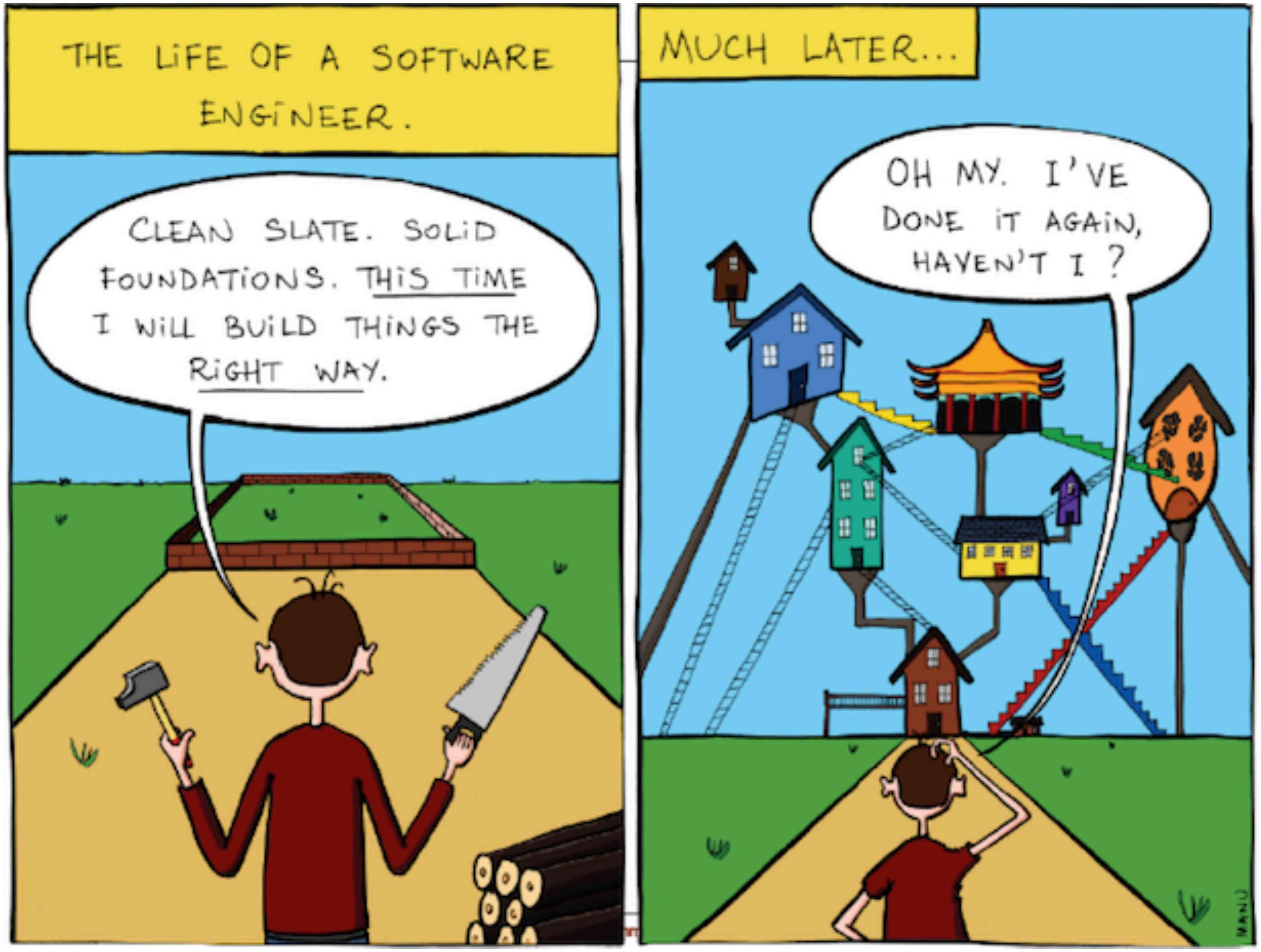
- re-grading
  - requires written request to instructor with explanation
  - will be done by the Instructor, mark may go up or down!
- late assignments
  - due date/time posted on the assignment and webpage
  - daily penalties will apply to late work (10% per day, up to 7 days, then a mark of 0 will be assigned)
- academic integrity
  - don't plagiarize. ok to discuss, but don't take notes.
  - properly site sources in your reports
- communication
  - only email me if it's confidential – put csc302h in subject
  - use discussion forum for questions, answer other's questions!

project

- can’t “engineer a large software system” in a semester, so lets pick one!
- will choose an existing open source project
- will likely use matplotlib
  - cool python 2D plotting library
  - check out the website http://matplotlib.org

why do we need a course on engineering large software systems?

why csc302h?


THE LIFE OF A SOFTWARE ENGINEER.
CLEAN SLATE. SOLID FOUNDATIONS. THIS TIME I WILL BUILD THINGS THE RIGHT WAY.
MUCH LATER...
OH MY. I'VE DONE IT AGAIN, HAVEN'T I ?
MANU

why csc302h? (2)

- historically, humans are very bad at engineering large software systems
  - see “why software fails” under readings
- how bad can we be? software is everywhere! some of us must be ok at it… right?
- annually, $bn are wasted on failed or over-budget large software projects.
  - lots of room for improvement!

why csc302h? (3)

- national programme for IT in the NHS (UK national health service) NPfIT:
  - cancelled project took 9 years and cost £12bn
  - from computerworlduk.com, 11/09/2011:

“The [UK] government will formally announce the scrapping of the National programme for IT in the NHS…to “urgently dismantle” the health service IT scheme comes after a series of damming reports…The final nail in the £12 billion scheme will be announced this morning…set up in 2002, is not fit to provide services to the NHS. ‘There can be no confidence that the programme has delivered or can be delivered as originally conceived,’”

why csc302h? (4)

- is it because they didn't use agile?
- agile projects can fail too! and just as bad
  - universal credit is Britian's plan to consolidate all welfare payments into one.
  - touted as the world's biggest agile software project, now close to total failure!
  - original budget = £2.2bn, cost so far = £12.8bn
  - article:

    http://news.slashdot.org/story/13/05/25/139218/worlds-biggest-agile-software-project-close-to-failure

- I bet the welfare recipients could have used that £12.8bn!

why csc302h? (5)

2003 blackout

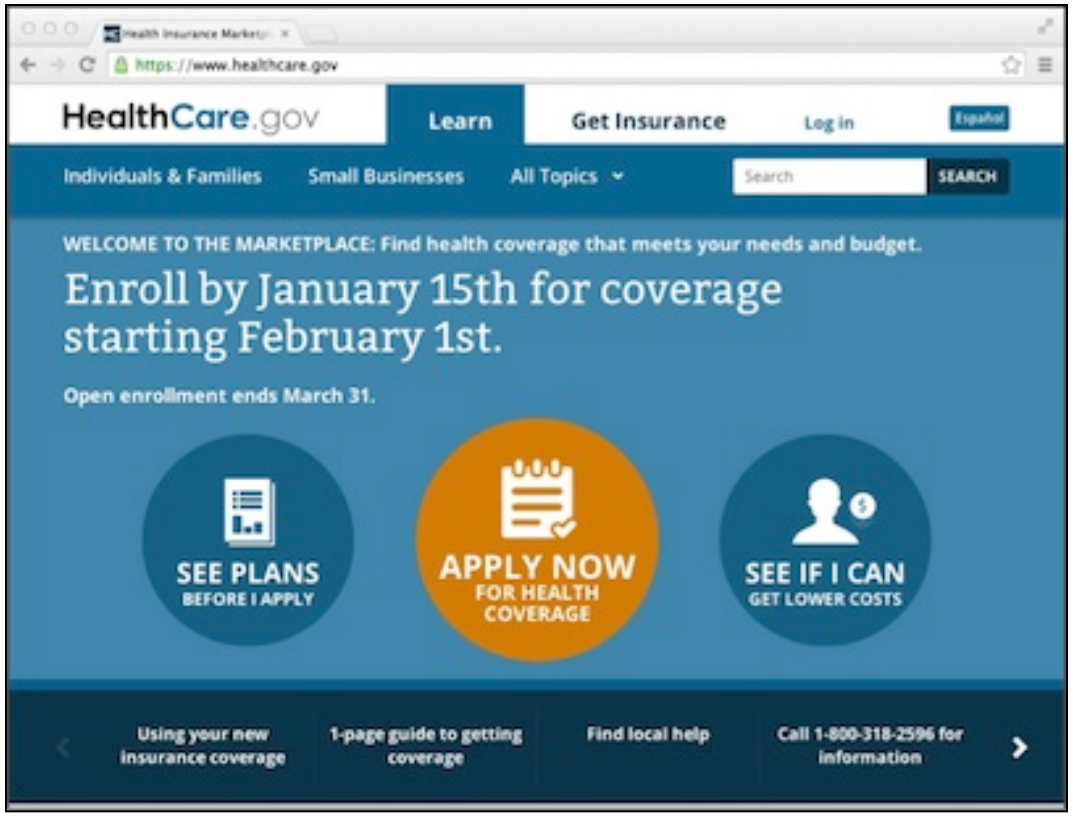
race condition on computer system in Ohio caused stalling of audio/visual alerts

primary system failed, then shortly after, the backup also failed (d’oh!)

Computer Science
UNIVERSITY OF TORONTO

why csc302h? (6)

HealthCare.gov


https://www.healthcare.gov
HealthCare.gov
Learn
Get Insurance
Log in
Español
Individuals & Families
Small Businesses
All Topics
Search
SEARCH
SEE PLANS BEFORE I APPLY
APPLY NOW FOR HEALTH COVERAGE
SEE IF I CAN GET LOWER COSTS
Using your new insurance coverage
1-page guide to getting coverage
Find local help
Call 1-800-318-2596 for information

WELCOME TO THE MARKETPLACE: Find health coverage that meets your needs and budget.

Enroll by January 15th for coverage starting February 1st.

Open enrollment ends March 31.

$500m for a website? are you serious?!?!

why csc302h? (6)

- author of *Why Software Fails* states that:

  "Studies indicate that large-scale projects fail three to five times more often than small ones."

- article:

  http://spectrum.ieee.org/computing/software/why-software-fails

what is large?

- what do we mean when we say a software system is ***"large?"***
  - class discussion
  - some examples
  - largest software you have ever worked with?

what is large? (2)

- what makes a software system ***"large?"***
  - kloc?, "what about comments?", ok, fine, executable statements when compiled?
  - number of bytes?
  - person-hours, or some other effort metric?
  - number of developers?
  - number of features?
  - number of processors running the code?
  - number of users of the software?
  - number of bugs?

Computer Science
UNIVERSITY OF TORONTO

what is large? (3)

– size of the box? number of floppy disks, and hours it takes to install it? ☺

what is large? (4)

- answer (well at least my answer):
  - something that is not a *"spike"* and is intended for end users – released
  - can benefit from, and is not hindered by, proper practices (modeling, source control, automated unit testing, continuous integration…)
  - standard development tools are not too cumbersome (ex. making an eclipse project is overkill => not "large" for our purposes)
  - so, basically anything non-trivial

Computer Science
UNIVERSITY OF TORONTO

summary

- this course deals with the challenges of major software projects
  - working with legacy code/systems
  - analyzing problems
  - deciding what is feasible, with the given team, and the amount of time available
  - delivering quality software in a professional software environment

summary (2)

- this course is different from most
  - you will work on a much larger project with code written by (many) people you've never met
  - you will use your own judgment in deciding what is feasible
  - you will manage your own risks and plan
- your mileage will vary
  - there are no right or wrong answers
  - credit will be given for good judgment

summary (3)

Looking forward to a fun and rewarding term!

CUSEC 2014

canadian university software engineering conference

january 16 – 18, montreal
http://2014.cusec.net

next lecture…

- modeling
- uml review
- form groups