ECE 50874: Advanced Software Engineering

Syllabus and Course Overview

Assistant Prof. James Davis
Assistant Prof. Santiago Torres-Arias

Slides outline

- Getting to know each other
- About the course
- Course structure
- Course logistics

Getting to know each other

Course staff

Instructor: Assistant Prof. James Davis
Indy section instructor: Assistant Prof. Santiago Torres-Arias

VIRGINIA TECH

PhD
2020

Research Intern (RiSE)
Microsoft
Jun 2019 – Aug 2019 · 3 mos
Redmond, Washington

Worked at MSR under Patrice Godefroid.

Research Intern (Storage)
IBM
May 2018 – Aug 2018 · 4 mos
Almaden, CA

Worked on a data provenance system wit

Software Test Engineer (GPFS)
IBM
2012 – Aug 2017 · 5 yrs

Now it’s your turn

- Name
- Describe yourself:
  - Give three adjectives and one noun
  - Career goals
  - What do you hope to gain from this class?
- After class, complete the Brightspace Discussion assignment “*Getting to know you*”

About the course

(“Syllabus review”)

Pre-requisites

- *Undergraduates*: ECE 461 or commensurate experience with instructor approval.
- *Graduate students*: Graduate standing and an interest in software engineering. Students will be best equipped for success if they have prior experience developing non-trivial software projects.

What you will learn in this course

Outcome	Primary Instruments
1. Understand mathematical bases for software engineering (*e.g.*, formal methods in software design and software verification) [ABET 1].	• Homework
2. Formulate, conduct, and report on a team-based research-oriented project in software engineering [ABET 1,5,6,7]	• Team Project
3. Consider the ethical implications of software engineering failures and successes [ABET 4]	• Homework • Class participation • Individual project
4. Summarize and analyze scholarly findings, both verbally and in writing [ABET 3]	• Homework • Class participation • Team Project

Why this course is important

- Software is key to realizing computer-based systems
- Software failure can be catastrophic

Risk of Inaccurate Results with Thermo Fisher Scientific TaqPath COVID-19 Combo Kit - Letter to Clinical Laboratory Staff and Health Care Providers

2020 (FDA warning)

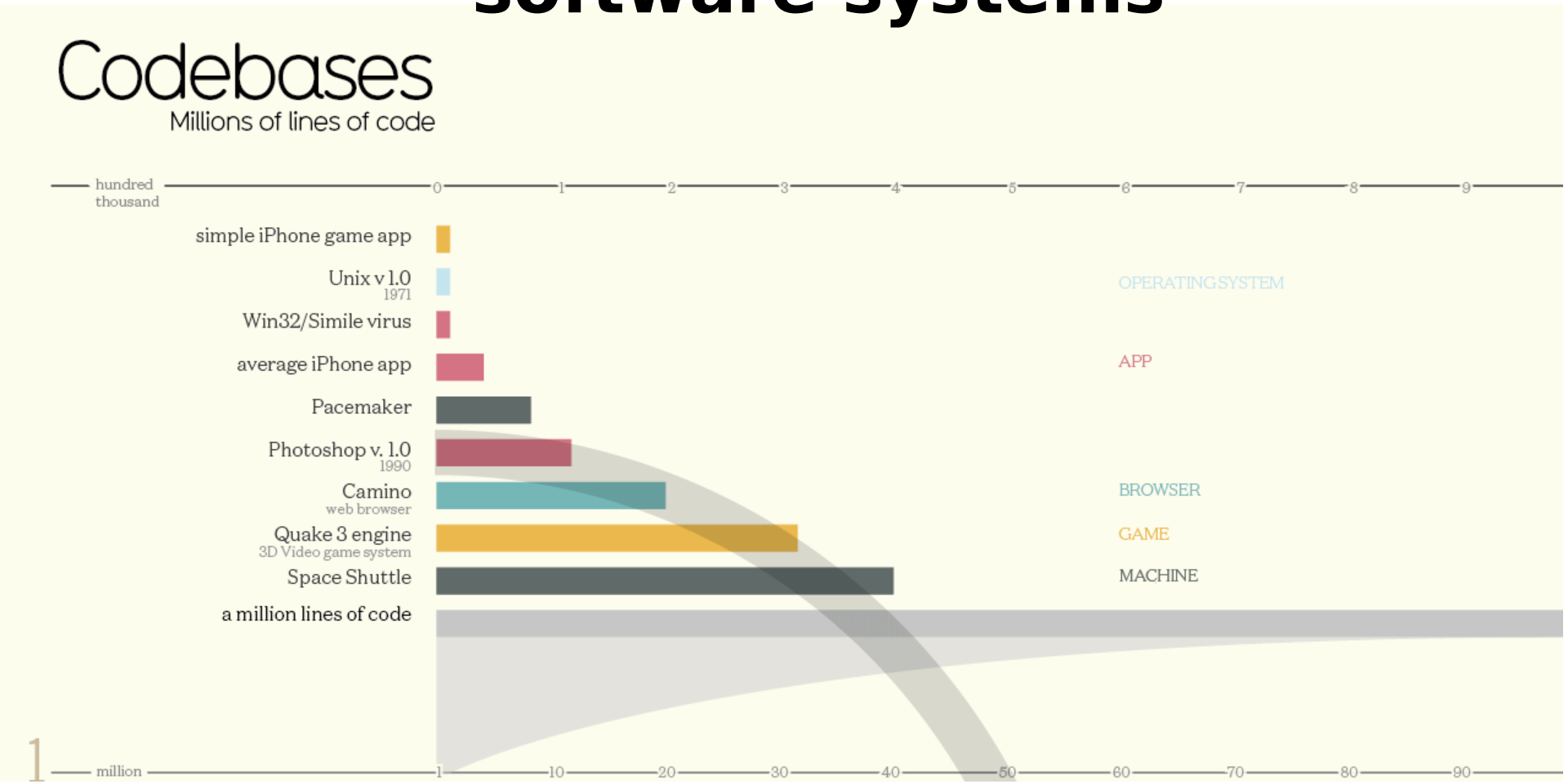
https://research.cs.queensu.ca/home/cisc323/2006w/slides/BillKelly.pdf

May 2021

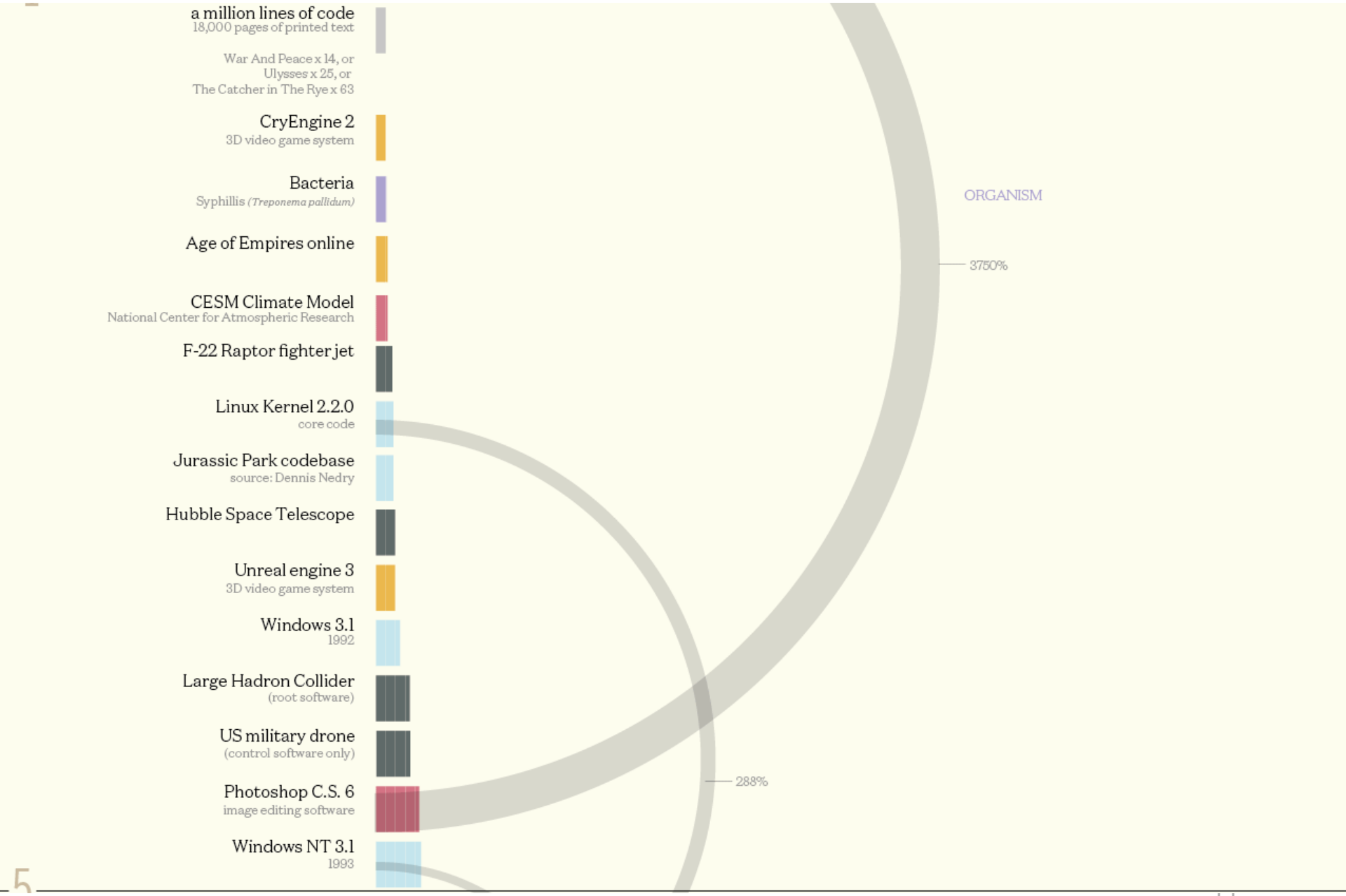
1985-

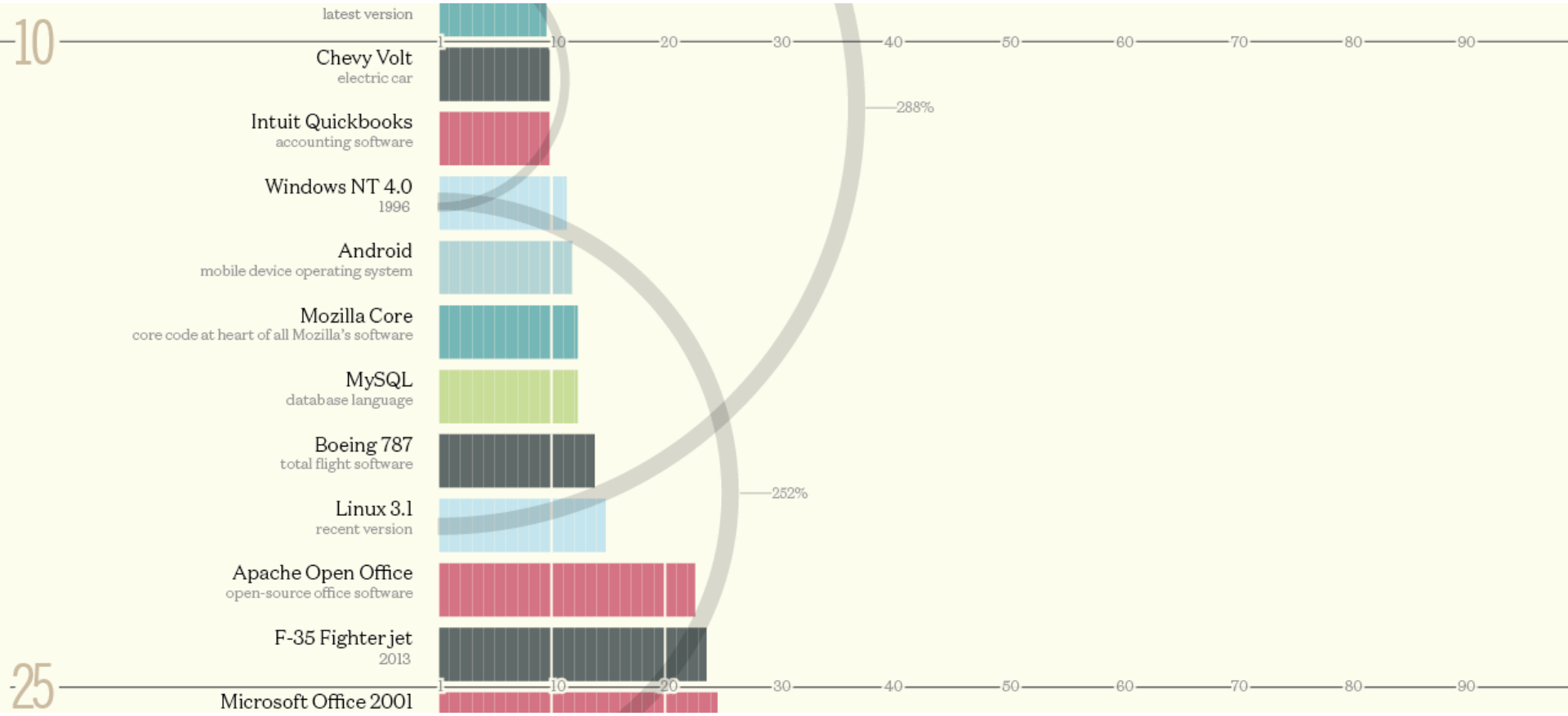
1987-1990

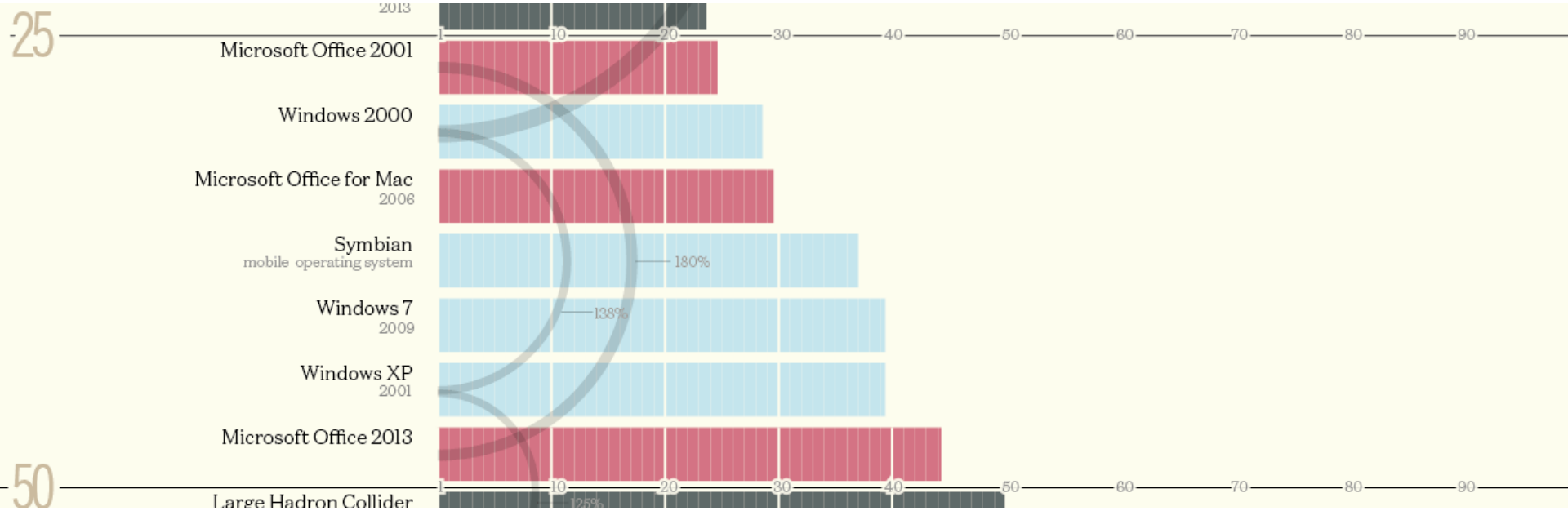
Causes of complexity in software systems

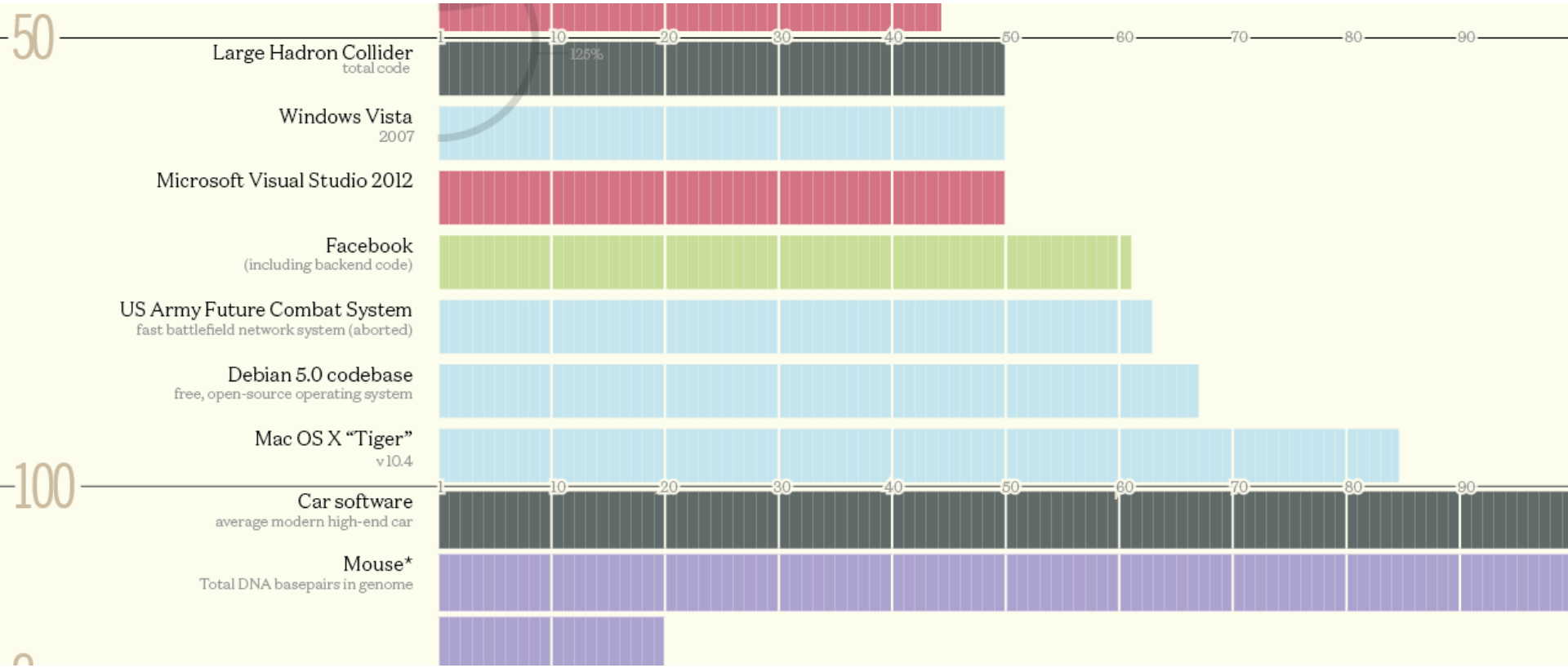


Source: https://www.visualcapitalist.com/millions-lines-of-code/


a million lines of code
18,000 pages of printed text
War And Peace x 14, or
Ulysses x 25, or
The Catcher in The Rye x 63
CryEngine 2
3D video game system
Bacteria
Syphillis (Treponema pallidum)
Age of Empires online
CESM Climate Model
National Center for Atmospheric Research
F-22 Raptor fighter jet
Linux Kernel 2.2.0
core code
Jurassic Park codebase
source: Dennis Nedry
Hubble Space Telescope
Unreal engine 3
3D video game system
Windows 3.1
1992
Large Hadron Collider
(root software)
US military drone
(control software only)
Photoshop C.S. 6
image editing software
Windows NT 3.1
1993
ORGANISM
3750%
288%


latest version
10
1
10
20
30
40
50
60
70
80
90
Chevy Volt
electric car
Intuit Quickbooks
accounting software
288%
Windows NT 4.0
1996
Android
mobile device operating system
Mozilla Core
core code at heart of all Mozilla's software
MySQL
database language
Boeing 787
total flight software
252%
Linux 3.1
recent version
Apache Open Office
open-source office software
F-35 Fighter jet
2013
25
1
10
20
30
40
50
60
70
80
90
Microsoft Office 2001


2013
25
1
10
20
30
40
50
60
70
80
90
Microsoft Office 2001
Windows 2000
Microsoft Office for Mac
2006
Symbian
mobile operating system
180%
Windows 7
2009
138%
Windows XP
2001
Microsoft Office 2013
50
Large Hadron Collider
125%


50
Large Hadron Collider
total code
125%
Windows Vista
2007
Microsoft Visual Studio 2012
Facebook
(including backend code)
US Army Future Combat System
fast battlefield network system (aborted)
Debian 5.0 codebase
free, open-source operating system
Mac OS X "Tiger"
v10.4
100
Car software
average modern high-end car
Mouse*
Total DNA basepairs in genome
1
10
20
30
40
50
60
70
80
90

Why might each of these be…?

Relatively simple

- Pacemaker (80 KLoC)
- UNIX v1.0 (10 KLoC)

Relatively complex

- Linux 3.1 (15 MLoC)
- Windows 7 (40 MLoC)
- Facebook (60 MLoC)
- Boeing 787 (15 MLoC)
- Intuit Quickbooks (10 MLoC)

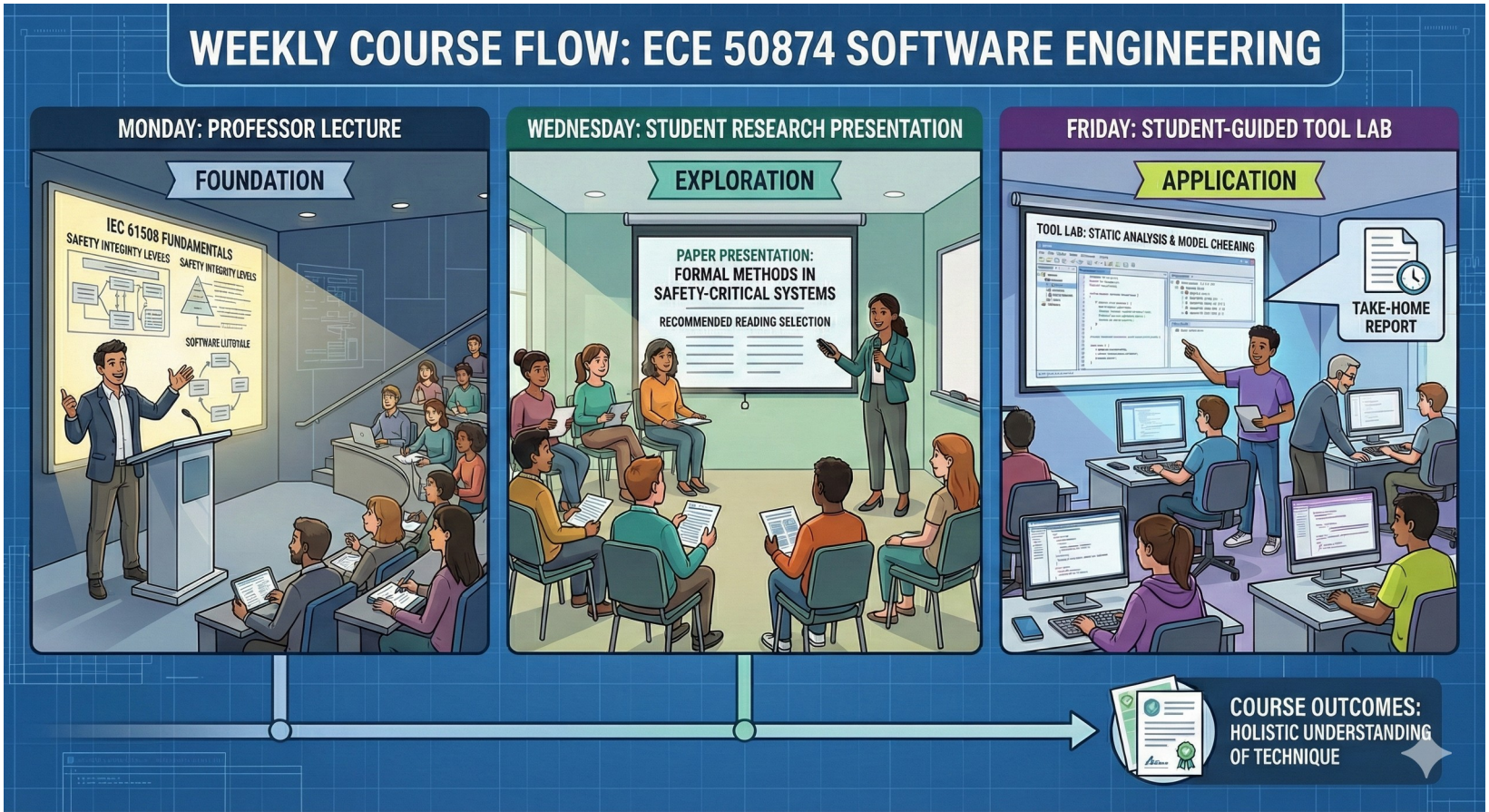
What I expect from you in this course

- Think
- Engage
- Grow
- Do

Course structure, assessments, & policies

In-class time: Two parts

- Part 1: Overview of IEC 61508
- Part 2: Structured guidance on techniques
- Part 3: IEC 61508 at speed – Agentic SE



Assignments

Type of assessment	Planned activities and time estimates	Total points
Homework (50 hours)	1. Lab report: Report on in-class lab activities (8-10x, 2-3 hours each) 2. Reading responses: Assigned for four research papers (4x, 4-6 hours each) 3. Project peer critique: Read and give feedback on another team's work (2x, 3-4 hours each)	20
Class participation (10-20 hours)	1. Class discussions: Weekly posts. Posing good questions, formulating good answers (tracked via discussion board) 2. Research paper presentation (asynchronous students may record this for viewing in class, or join synchronously to present) 3. Attendance (enforced as needed)	15
Team Project (80 hours)	1. Proposal sketch (written) 2. Proposal (oral and written) 3. Response to proposal critiques (written) 4. Peer assessment 1 5. Project updates (biweekly) 6. Final report rough draft 7. Final report 8. Peer assessment 2	50
Individual Project (20 hours)	1. Three project ideas and feasibility checks (Week 3) 2. Initial analysis (Week 7) 3. Refined analysis (Week 12) 4. Final analysis (Finals Week)	15
150-165 hours (16 week course, 9-11 hours/week, consistent with Purdue's guidance)		***Total: 100***

Homework

- 20% of your course final grade
- Lab reports
- Project peer critique
- Research paper critiques

Course participation

- 15% of your course final grade
- In-class/Brightspace discussion
- Paper and Tool Lab presentations

Course project

- 50% of your course final grade
- Teams of 3
- Semi-open-ended *topic*
- *S*tructured *format*
- Two types of projects: Research & Practice
- Some description now
- More details on Wednesday

Important dates

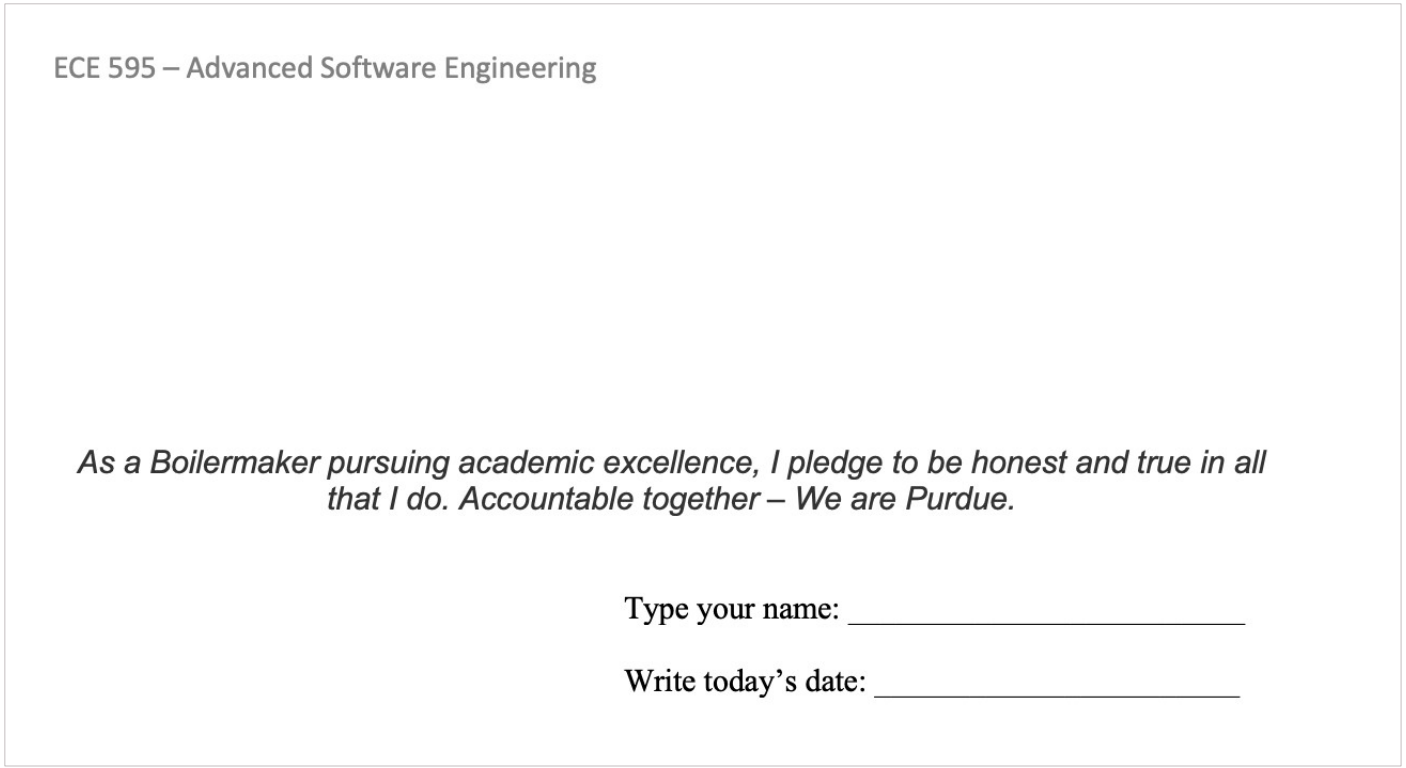
1. Interest survey (1/14)
2. Team composition (1/19)
3. Proposal sketch
4. Proposal
5. Several milestones
6. Final report rough draft (pre-Quiet week)
7. Final presentations

Individual project

- 15% of your course grade
- Task: Complete an independent failure analysis and identify process or product techniques that might have prevented the failure
- Timeline
  - Three project ideas and feasibility checks (Week 3)
  - Initial analysis (Week 7)
  - Refined analysis (Week 12)
  - Final analysis (Finals Week)

Academic honesty

As a Boilermaker pursuing academic excellence, I pledge to be honest and true in all that I do. Accountable together – We are Purdue.



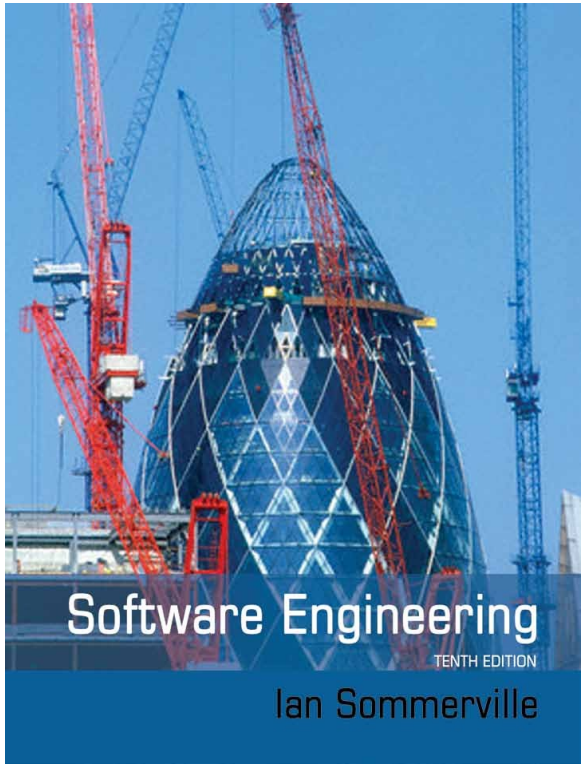
Course logistics

Logistics overview

- One required standard (we will provide access)
- Two required texts
- Brightspace → Discussions, Slides, Lecture recordings, Readings, Q&A
- TBD on submission – Course staff is still discussing Brightspace vs. Gradescope

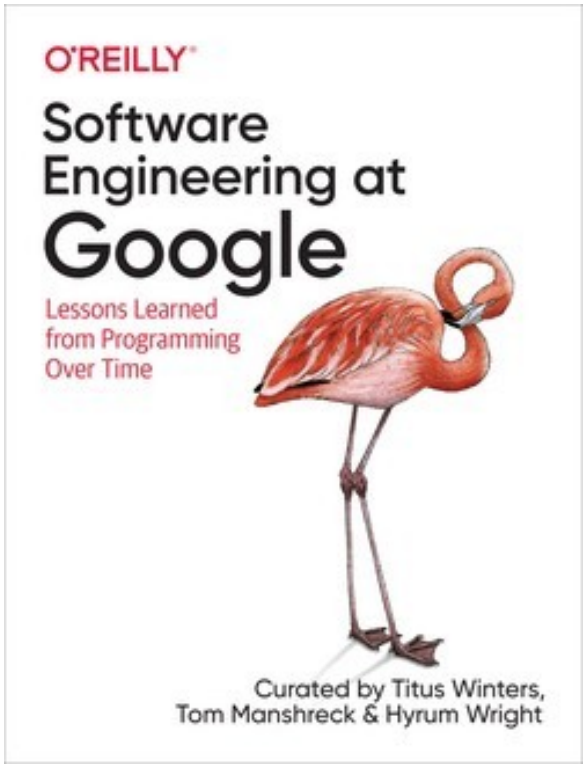
Texts

https://iansommerville.com/software-engineering-book/



I could not find the 9th or 10th editions through Purdue's library. There were major changes between the 8th and 9th editions.

https://www.oreilly.com/library/view/software-engineer
-at/9781492082781/

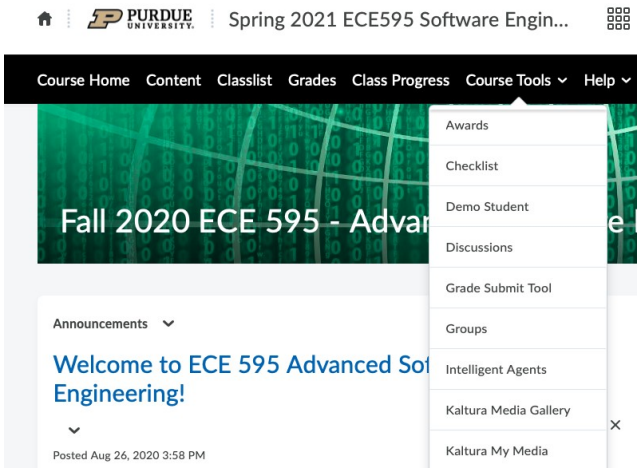


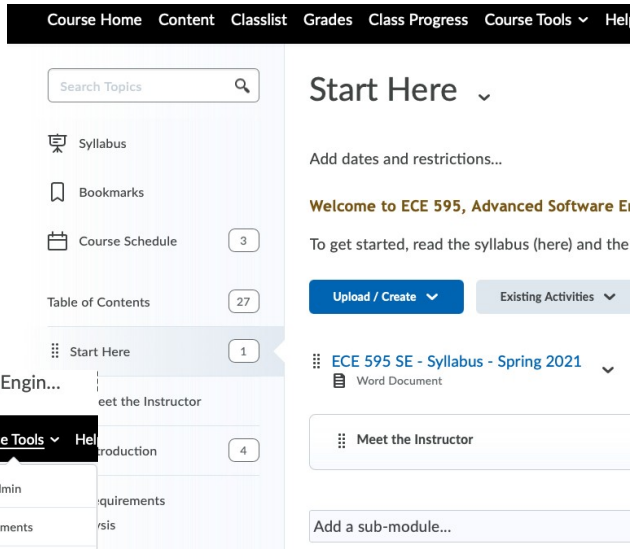
Like all O'Reilly books, an e-editi
is available through Purdue's libr

Brightspace 101

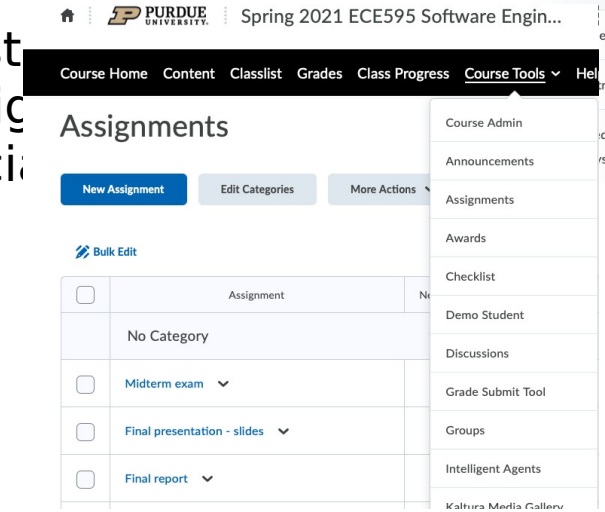
- Recordings: Kaltura Media Gallery

(incl. live stream)



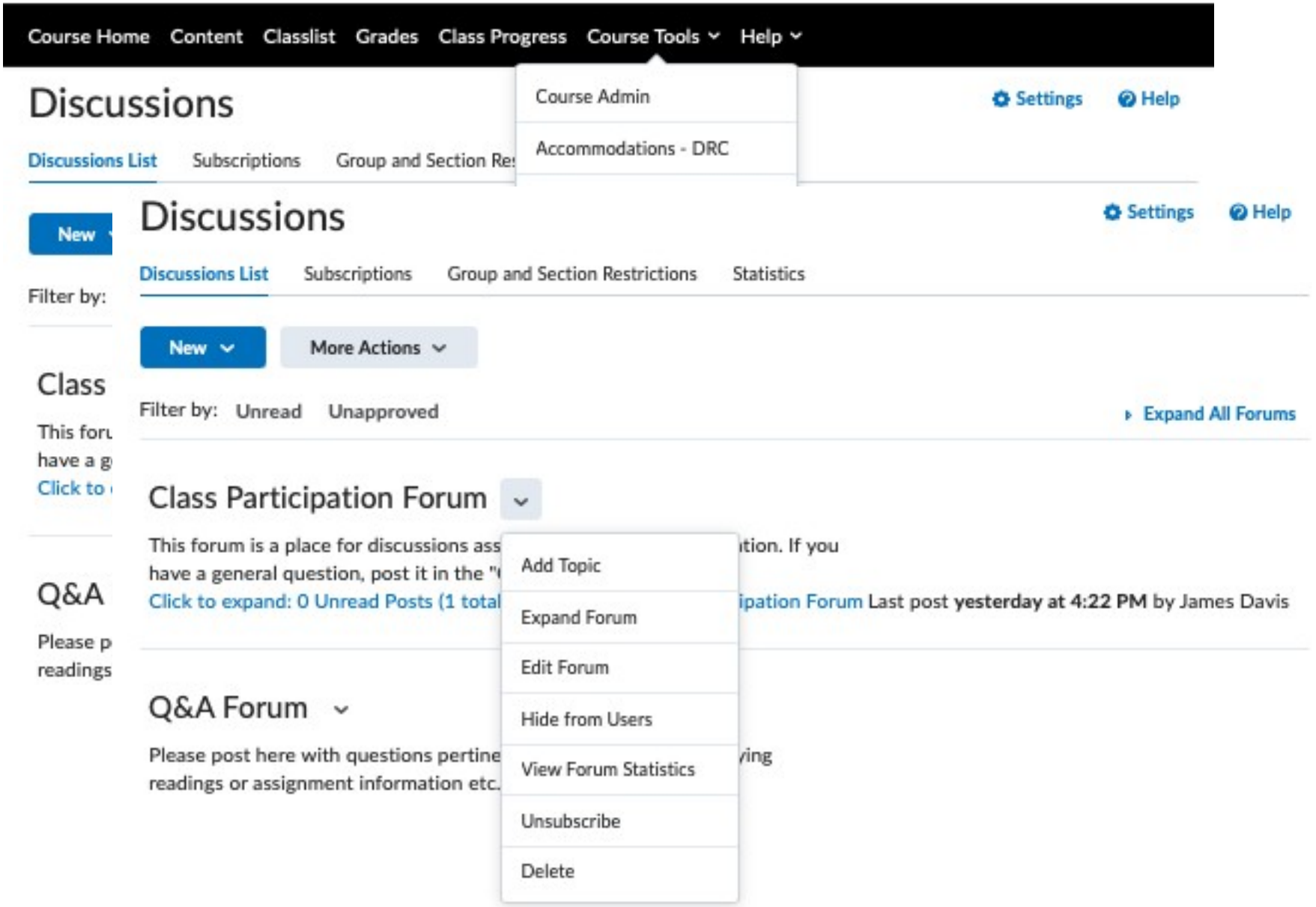


- Week-by-week: Click “Content” in the banner
  - Still Under Const
  - Non-Project assig
    under the associ



- Upcoming assignments

Accessing Brightspace Discussions



Office hours

- See Brightspace
- Zoom if needed
- By appointment if needed

Asking questions

- General question: Post it in the Brightspace thread titled “Course Q&A”
- Private question: Email the course staff
- But please Read The Fine Manuals

- All emails to course staff should begin with the subject “ECE 50874: …”

Asynchronous students

- Greetings!