Software Design, Development and Engineering

Administrative Matters First

- Syllabus
- Reverse roll call.

This course's target audience

- This course
  - intended as a final semester undergraduate course or a first semester grad course for those with limited industry experience.
    - builds on the undergraduate software engineering course
  - I assume you know
    - The basic ideas behind automated tests
    - Design patterns
    - Agile and Waterfall design
    - The basic idea of continuous integration
    - How to use version control software like git
    - How to read and understand code in 1-2 high level languages for a reasonably large (for academia) project of a couple thousand lines.
  - I am aware that some of you have done some of the first few at a more theoretical than practical level.

What I’m going to try to accomplish with you

- For undergrads:
  - Take those ‘legos’ you learned to carve in the other classes and put them together
  - Polish off the degree and fill any holes
- For everyone
  - Work you though building two projects in a more realistic manner than most of academia.
    - First one you will have to ‘eat your own dog food’ building on your own earlier decisions
  - Examine the current state of the industry and what it takes to get there from here
  - Practice using tools on problems large enough to make your work as much about you as about your ability to use the tools.

Toolchain for the class

- The class website
  - https://webhost.bridgew.edu/jsantore/Spring2026/SoftwareDev/
- Slack, because in tech, slack is most widely used
  - https://sqmagazine.co.uk/slack-vs-microsoft-teams-statistics/
- Git
  - Because (nearly) all new projects use git for version control
- Github with github actions
  - Because it is ‘free’ since Microsoft datamines everything you do on the free accounts
- jetbrains/google antigravity.

Constituent groups

- So where did a lot of this course come from?
  - Outside industry advisory board
  - Graduating seniors and MS students
    - Asking sophomores vs asking seniors/4th semester MS students who are applying for jobs
  - Alumni
  - Graduate advisory board
    - Which consists of graduates from our program along with a few other Master's degree holders.
  - Departmental self-reflection

Group Work

- At one time we didn’t do enough group work
- Theoretically today we do it in comp152, comp350, comp390 and here.
  - And in several electives
  - And for the grads in several classes.
- Group work is tricky enough at the university level
  - But real software dev work is done in groups – exclusively
  - Not always fun at first for all students
- But how well did it actually go – especially earlier?

• Not enough work on large projects

- Large Projects in academia are/were quite scarce
- I first heard this critique in our industry advisory board
- Then from many (many) talks by industry professionals as the biggest problem with academic computer science programs
  - At one point Nowhere in our required courses were students given a large code base, some documentation, and a pointer to a problem and told “fix it”
  - This is a glaring hole
  - Have we done it already elsewhere in the program yet?

Living with your own work

- Another thing that we need to see more of is students who have to live and work with the results of their own earlier decisions
  - Something that industry practitioners argue for a lot
  - Give the first (of many?) rants about industry code vs academic code
- I believe we are now doing this in comp390 software engineering?
- How about at the graduate level?
-
- You will have the opportunity to do this in the first project

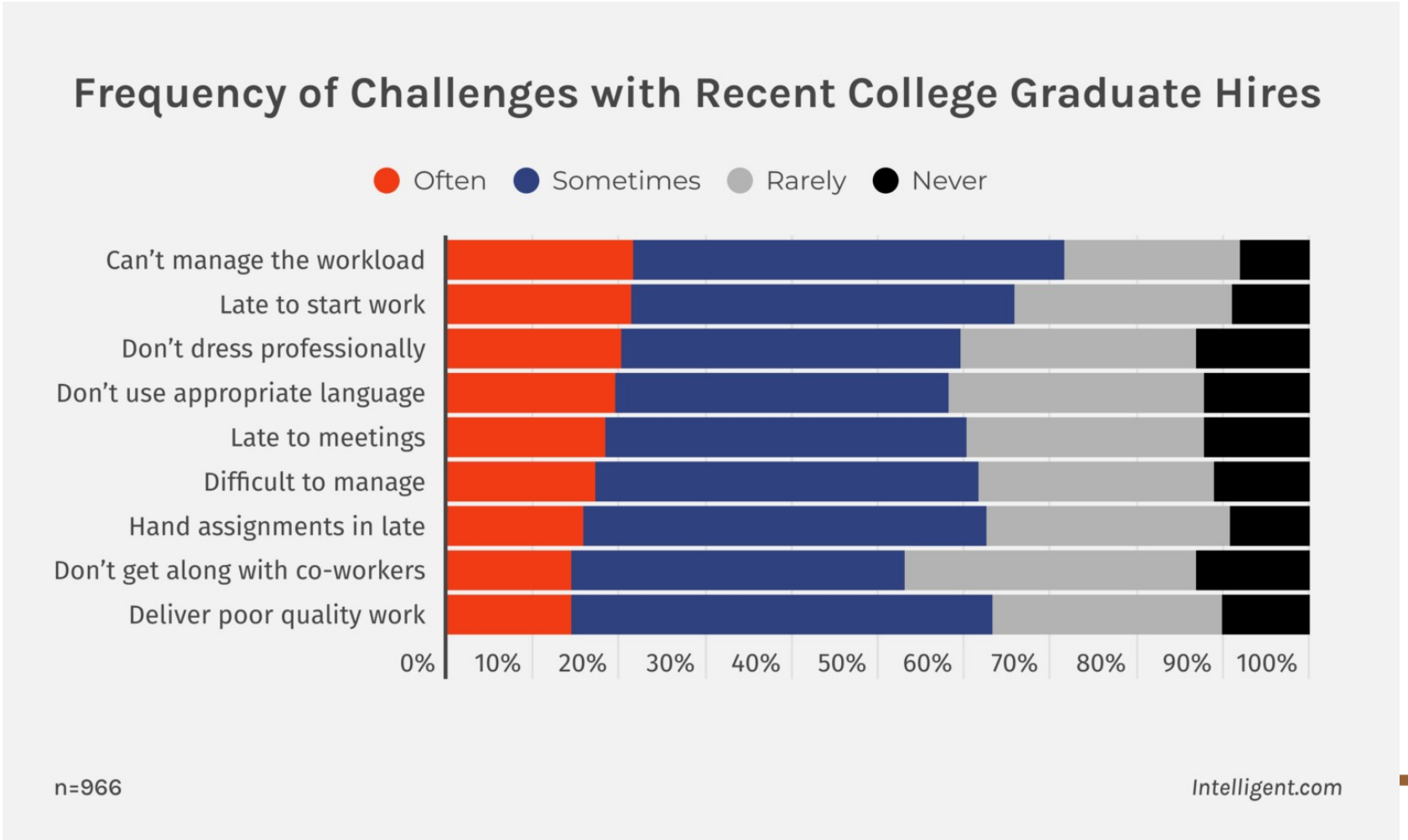
Project oriented goals for the course

- You sit down in front of a ‘big’ (40 files or so) project and
  - Realize its no big deal, nothing that 20-40 hours won’t be enough to get your head around.
  - You just start using documentation and testing to learn the new system.
- And on our way there
  - Practice agile development by building a larger project in smaller incremental steps.
- Have your <gasp> moment here – not at your first job
  - And “impostor syndrome?”

What do I do now?

- From a grad a couple of year ago:
  - “What do I do now? You all should help us figure out what to do next”
- One of the themes of the course:
  - Use Career Services
  - Start/Grow your network
    - Make meaningful connections with peers and people in industry
      - On or off line, but no ‘driveby comments’ or AI slop.
  - Practice for the interview
  - Practice being a professional.
    - Professional writing
    - Professional speaking
    - Perhaps occasionally follow a professional dress code

I want to see all of you in the black and gray.



- No red or blue for our grads please

The Job of a tech professional

- What is the job of a tech professional?
  - Lucky volunteer^{tm} ?

The Job of a tech professional

- What is the job of a tech professional?
  - As articulated by a member of our advisory board (and 2009 BSU CS grad)
    - “Their job is to solve problems that the business has using technology”
  - Once upon a time some companies
    - “bring your whole self to work”
    - Virtually nobody says that any more
  - Your main job is solving problems.
    - Which includes communicating ideas clearly and professionally in written and verbal form.
    - And of course using your deep critical thinking and problem solving to help build technical solutions to problems.

On the Topic of LLMs/AI

- LLM/Coding AI in the industry
  - What have you heard?

On the Topic of LLMs/AI

- LLM/Coding AI in the industry
  - It is a muddy area – copyright issues
  - Depending on area might be more or less allowed
  - Like?

On the Topic of LLMs/AI

- LLM/Coding AI in the industry
  - It is a muddy area – copyright issues
  - Depending on area might be more or less allowed
    - Defense contractors/classified or better security: not allowed at all
    - Compliance-required industries (like?)
      - Very limited
    - Some other industries allow more use
  - What is it good for?
  - What is it not?
- Most common Industry perspective?:

On the Topic of LLMs/AI

- LLM/Coding AI in the industry
  - It is a muddy area – copyright issues
  - Depending on area might be more or less allowed
    - Defense contractors/classified or better security: not allowed at all
    - Compliance-required industries (like?)
      - Very limited
    - Some other industries allow more use
  - What is it good for?
  - What is it not?
- Most common Industry perspective?:
  - “AI coding assistants are like pair programming with a really enthusiastic, fast typing junior developer who never learns from their mistakes”

AI in this class

- In a lot of larger companies, employees can only use company approved tools.
- To simulate that in this class, this following (and only the following) AI tools will be allowed for use to assist writing out of class code projects:
  - See links on the resources page of the class website.
- First project will have some sprints that I ask you to use agentic AI, you need to be able to read the output and understand and vouch for it.

The last major course goal

- Lifetime learning
  - One of the major goals of a college education
    - Teach you how to learn on your own.
  - When you begin (first two semesters) want lots of “hand holding”/help
  - By this course we want to take off the training wheels
    - I’ve heard from some colleagues that some reluctance
    - But as some of you who were in the recent CS club presentations
    - you’ll get “go look at this” in industry – and you have to go learn it and see if it will work.
    - Get some practice here

Soft Skills and Hard skills

- Soft Skills
  - I wanted a book on developing software in real life
  - I’ve tried books, but in recent semesters I’ve used some podcasts
  - Cowboy coding bad!
  - Take care of yourself
  - People with both coding and people skills will go far.
    - People with only one will need to work on the other

Tools

- You will graduate at the end of this semester or next
  - And will never use blackboard again
  - So lets use actual industry tools
    - Not every industry job, and many won't have this exact mix, but I'll pick some common ones
  - Slack
    - Heavily used in US tech industry – even if whatapp and office/teams is beating it world wide and in education
      - https://electroiq.com/stats/slack-statistics/
      - https://colorlib.com/wp/slack-statistics/
  - Git nearly everyone uses it
  - Github nearly all open source and even many closed source projects use it

Assignment

- Read the first chapter of the Pragmatic Programmer: “A Pragmatic Philosophy”
  - we’ll discuss this in a week
- Join the slack workspace for this class
  - Install slack on devices you will use and check it regularly
  - The join link is in the one and only blackboard message for this class.
- If you don’t have a github account for class use, make one
  - And send me your github ID. !!!!!!
    - This is the one that people often fail to remember. It is worth a quiz grade
    - Send it even if you’ve had me before and sent it to me once before.
- Get google antigravity or jetbrains if you can use the AI there.
  - Look at project? Depending on how long it takes to get here.