Special Topics Discussion

CS 485/698: AI-Assisted SE

Today’s Agenda

- A7 discussion groups
  - 6 discussions, ~10 minutes each
- Course wrapup and announcements (~10 minutes)

A7 discussion format

- Undergrads should all have a **letter of the alphabet**
  - Grads have a **big number**

A7 discussion format

- Undergrads should all have a **letter of the alphabet**
  - Grads have a **big number**
- Grad student instructions: spend no more than 5 minutes explaining your topic to the undergrads. Rest of time: answer questions from undergrads and/or undergrads discuss topic

A7 discussion format

- Undergrads should all have a **letter of the alphabet**
  - Grads have a **big number**
- Grad student instructions: spend no more than 5 minutes explaining your topic to the undergrads. Rest of time: answer questions from undergrads and/or undergrads discuss topic
- When I call time, undergrads move to next number (e.g., 1->2, 6->1)

A7 discussion format

- Undergrads should all have a **letter of the alphabet**
  - Grads have a **big number**
- Grad student instructions: spend no more than 5 minutes explaining your topic to the undergrads. Rest of time: answer questions from undergrads and/or undergrads discuss topic
- When I call time, undergrads move to next number (e.g., 1->2, 6->1)
- Congregate **near the number that corresponds to your letter**
  - As go to 1, Bs to 2, etc

A7 discussion format

- Undergrads should all have a **letter of the alphabet**
  - Grads have a **big number**
- Grad student instructions: spend no more than 5 minutes explaining your topic to the undergrads. Rest of time: answer questions from undergrads and/or undergrads discuss topic
- When I call time, undergrads move to next number (e.g., 1->2, 6->1)
- Congregate **near the number that corresponds to your letter**
  - As go to 1, Bs to 2, etc
- Michael and I will be rotating in reverse order and evaluating both **grad students' discussion leadership** and **undergrad participation**

A7 discussion format

- Undergrads should all have a **letter of the alphabet**
  - Grads have a **big number**
- Grad student instructions: spend no more than 5 minutes explaining your topic to the undergrads. Rest of time: answer questions from undergrads and/or undergrads discuss topic
- When I call time, undergrads
- Congregate **near the numbe**
  - As go to 1, Bs to 2, etc
- Michael and I will be rotating in reverse order and evaluating both **grad students' discussion leadership** and **undergrad participation**

For undergrads, this is just a participation grade. For grads, this is part of your A7 score.

A7 discussion format

- Undergrads should all have a **letter of the alphabet**
  - Grads have a **big number**
- Grad student instructions: spend no more than 5 minutes explaining your topic to the undergrads. Rest of time: answer questions from undergrads and/or undergrads discuss topic
- When I call time, undergrads move to next number (e.g., 1->2, 6->1)
- Congregate **near the number that corresponds to your letter**
  - As go to 1, Bs to 2, etc
- Michael and I will be rotating in reverse order and evaluating both **grad students' discussion leadership** and **undergrad participation**

Course Wrapup

- My claim from the first lecture:

  the same software engineering techniques that we've been using for decades to help improve the quality of code written by junior developers can be used to help improve the quality of AI-generated code

Course Wrapup

- My claim from the first lecture:

  the same software engineering techniques that we've been using for decades to help improve the quality of code written by junior developers can be used to help improve the quality of AI-generated code

- We've explored this in the context of requirements engineering, testing, CI/CD, static analysis, etc.

Course Wrapup

- My claim from the first lecture:

  the same software engineering techniques that we've been using for decades to help improve the quality of code written by junior developers can be used to help improve the quality of AI-generated code

- We've explored this in the context of requirements engineering, testing, CI/CD, static analysis, etc.
- So, **do you believe it**?

Course Wrapup: Predictions

- It's famously difficult to predict the future. But, my **educated guess** is that the future of software engineering will involve more:

Course Wrapup: Predictions

- It's famously difficult to predict the future. But, my **educated guess** is that the future of software engineering will involve more:
  - writing good tests yourself

Course Wrapup: Predictions

- It's famously difficult to predict the future. But, my **educated guess** is that the future of software engineering will involve more:
  - writing good tests yourself
  - reading and writing of natural language text
    - especially specifications and design documents!

Course Wrapup: Predictions

- It's famously difficult to predict the future. But, my **educated guess** is that the future of software engineering will involve more:
  - writing good tests yourself
  - reading and writing of natural language text
    - especially specifications and design documents!
  - code review

Course Wrapup: Predictions

- It's famously difficult to predict the future. But, my **educated guess** is that the future of software engineering will involve more:
  - writing good tests yourself
  - reading and writing of natural language text
    - especially specifications and design documents!
  - code review
  - debugging code you didn't write

Course Wrapup: Predictions

- It's famously difficult to predict the future. But, my **educated guess** is that the future of software engineering will involve more:
  - writing good tests yourself
  - reading and writing of natural language text
    - especially specifications and design documents!
  - code review
  - debugging code you didn't write
  - complex testing and analysis techniques to keep agentic coding tools honest
    - maybe even formal verification tools!

Course W… P…

My advice: focus on these skills to differentiate yourself in tomorrow's job market

- It's fam… …ed guess is that the future of software engineering will involve more:
  - writing good tests yourself
  - reading and writing of natural language text
    - especially specifications and design documents!
  - code review
  - debugging code you didn't write
  - complex testing and analysis techniques to keep agentic coding tools honest
    - maybe even formal verification tools!

Final Reminders

- P7 due Sunday

Final Reminders

- P7 due Sunday
- Project presentations on Monday and Wednesday next week

Final Reminders

- P7 due Sunday
- Project presentations on Monday and Wednesday next week
  - I sent out presentation order this morning
    - let us know if you need to be absent ASAP

Final Reminders

- P7 due Sunday
- Project presentations on Monday and Wednesday next week
  - I sent out presentation order this morning
    - let us know if you need to be absent ASAP
  - Note that all slides etc are due with P7 on Sunday
    - you may not change the slides before you actually present!
    - presentations will be **preloaded on my laptop** to reduce switching time, so include your demo link in the slides

Final Reminders

- P7 due Sunday
- Project presentations on Monday and Wednesday next week
  - I sent out presentation order this morning
    - let us know if you need to be absent ASAP
  - Note that all slides etc are due with P7 on Sunday
    - you may not change the slides before you actually present!
    - presentations will be **preloaded on my laptop** to reduce switching time, so include your demo link in the slides
- Course evals close on May 7 (Thursday)
  - please fill them out before then! Only 7 responses so far…