CMSC 398z
Effective use of AI Coding Assistants and Agents

Bill Pugh and Derek Willis

Dec 12th, 2025

Plan for today

What Bill learned from a deep dive into Quuly

Class discussions about how to improve officehours.cs.umd.edu

Follow-ups on work on quuly and congressional record

What Bill & Derek learned from this class

The need to keep on honing your skills and knowledge at using AI coding tools

Submit form with thoughts from today

Deep dive into quuly

I had never used any of the frameworks used by quuly before

- I familiar with many of the concepts

I never would have attempted to try to understand or modify quuly without AI tools

- maybe if I was being paid enough

It needs a lot of work. I haven't and don't plan to type a single line of code

- I've closely reviewed some of the existing code, and
- I will review critical parts of new code

Datarace bugs are nasty - exceptionally hard to write test cases for, rarely happens in practice

I was unfamiliar with quuly

As I've gotten deeper into quuly, and seen how badly some features seem to be implemented, I've investigated, and found that some of those features were not being used, or not exposed in the UI

Chicken and egg - which caused which?

Looking at conversations in quuly

Understand the relationship between a conversation and an interaction

Conversation is created when a student initiates a request for assistance

Each step in the process is an interaction (e.g., enqueue -> dequeue -> resolve)

a sequence, ordered by time

A Conversation

```
CREATE TABLE conversations (
    id TEXT PRIMARY KEY DEFAULT generate_uid(),
    course_id TEXT NOT NULL REFERENCES courses(id),
    initiator_id TEXT NOT NULL REFERENCES users(id),
    created TIMESTAMPTZ NOT NULL DEFAULT NOW(),
    state conversation_state NOT NULL DEFAULT 'in_queue',
    room_id TEXT NOT NULL REFERENCES rooms(id),
    topic TEXT NOT NULL,
    description TEXT NOT NULL,
    location VARCHAR(200)
);
```

An interaction

```
CREATE TABLE interactions (
    id TEXT PRIMARY KEY DEFAULT generate_uid(),
    conversation_id TEXT NOT NULL REFERENCES conversations(id),
    typ interaction_typ NOT NULL,
    created TIMESTAMPTZ NOT NULL DEFAULT NOW(),
    responder_id TEXT NOT NULL REFERENCES users(id),
    comment_data TEXT
);
```

Notes, and normalization

The state of the conversation is the only mutable part of a conversation

An action on a conversation does two things:

- adds a new interaction
- updates the state of the conversation
  - uniquely determined by the type of the most recent interaction
  - An example of *denormalized* data - a magic word we are stuck with
  - In a normalized database, there is no redundant, duplicated data, no opportunity for inconsistency
- With normalized data, getting information you need often requires performing lookups that join multiple tables

A Conversation

```
CREATE TABLE conversations (
    id TEXT PRIMARY KEY DEFAULT generate_uid(),
    course_id TEXT NOT NULL REFERENCES courses(id),
    initiator_id TEXT NOT NULL REFERENCES users(id),
    created TIMESTAMPTZ NOT NULL DEFAULT NOW(),
    state conversation_state NOT NULL DEFAULT 'in_queue',
    room_id TEXT NOT NULL REFERENCES rooms(id),
    topic TEXT NOT NULL,
    description TEXT NOT NULL,
    location VARCHAR(200)
);
```

Mutable - but derivable from most recent interaction

Immutable, but maybe it shouldn't be

The most recent interaction

Sometimes you want to find all the interactions associated with a conversation

But most of the time, if you want an interaction, you just want the most recent one

doing a join with all of the interactions for a conversation, and then filtering to find the most recent one, is somewhat inefficient

Probably the reason state is stored in conversations, rather than being derived from the most recent interaction

Problematical concurrency pattern - Check then act

Problem is act assumes nothing has changed between the check and taking the action

- Has to be done with a lock that prevents change

Way more efficient to do it with an atomic check and act operation

- only succeeds if check is OK
- compare and swap is a common primitive for in memory synchronization
- Similar thing can be done with an update operation in SQL

Harmless Check and Act code in queues.go

```
res := p.client.ZCard(getQueueKey(courseID, roomID))

count, err := res.Result()

…  error handling code …

convo := p.client.ZPopMin(getQueueKey(courseID, roomID), 1)

values, err := convo.Result()

// In this case, could simplify removing call to ZCard

// It won't return anything if the queue is empty
```

A bad datarace bug

Student Leaves While TA Dequeues (Race Condition)

When a student clicks "Leave Queue", the code path is:

FindByID — fetch conversation state from database

GetNextState — validate transition is legal (pure function, no side effects)

Remove — call ZRem to remove from Redis queue

Database transaction — update state to 'ongoing', insert 'leave' interaction

Window 1: TA dequeues before student's FindByID

Student reads state = 'being_helped' (TA already changed it)

GetNextState('being_helped', LEAVE) returns ok = false

Student gets INVALID_TRANSITION_ERROR

Database is consistent, but error message is confusing

Window 2: TA dequeues after student's FindByID but before student's tx.Commit()

Student reads state = 'in_queue'

GetNextState('in_queue', LEAVE) returns 'ongoing', ok = true

TA dequeues (Redis ZPopMin + DB update to 'being_helped')

Student's Remove calls ZRem, which returns 0 items removed but no error

Student's DB transaction overwrites state to 'ongoing' (no check that state is still in_queue)

DATABASE IS CORRUPTED — TA thinks they're helping, but state says 'ongoing'

Root cause:

The Remove function in zion/redis/queues.go doesn't check if ZRem actually removed anything

The database update didn’t check that the state is still in_queue

Another common bug pattern: ignoring return values

```
func (p *PushClient) Remove(courseID, roomID, conversationID string) error {
    res := p.client.ZRem(getQueueKey(courseID, roomID), conversationID)
    return res.Err()  // Only checks for errors, not whether anything was removed
}
```

Rethinking conversations

Let's allow conversations to move from room to room, and allow students to move from one location in the room to another.

Move potentially mutable fields from conversation to interaction

Add a references from a conversation to the most recent interaction

- not normalized - barely
- allows for atomic update:
  - Can do a database operation that updates the most recent conversation only if you correctly identify what had been the most recent conversation

Need a join to get most recent interaction

We've moved a lot of fields into the interaction, need to do join to get that from db

- in particular, the getting the state of a conversation requires a joi

But we can store a conversation record in redis that has all the relevant information for a conversation and most recent interaction

Database is the source of truth

Virtual rooms are a mess

When a new class is created, all virtual rooms are added to it

- new courses wind up with 23+ virtual rooms

Nobody actually uses virtual rooms

Instructors create 'physical rooms' with names like online office hours

A TA or instructor can't be on duty in both a virtual and physical room

Fixing virtual rooms

Rooms can be physical, virtual or hybrid, and are always associated with a class

- They are really a queue rather than a room

When a TA goes on duty, or a student requests help, in a hybrid room, they specify if they are available physically, virtually, or both

When a TA dequeues a student, the student gets told whether it is in person or virtually

- The TA can provide a message, such as a link to a zoom room

Tags

Most courses don't use tags

- Either the course doesn't define them, or students don't use them

Some use them heavily

What could be changed to make them more useful and used?

Other feature requests

Integrate office hour schedules

Prevent camping in queue

- students enqueuing many hours before office hours open

Allowing prioritization of students who haven't recently received help

Your turn

Most/all of you have used quuly more than I have

Discuss around the table, keep notes

- you will be asked to provide your thoughts on form submission for today

What have been your experiences with quuly in courses?

Are there other similar tools you've used in non-CS courses?

Thoughts on the ideas I discussed?

Your own ideas for how we could improve quuly?

Followups on congressional record and quuly

congressional record

some of the issues we found were pretty small

- e.g., fixing a regex, handling two volumes on Jan 3rd

Some were much more substantial

- changing how it handles things that are not speeches
- Really solving those well involves significant changes that would require a major revision
  - involve significant buy-in from maintainer, users of that project
  - we will get those changes in

quuly

plan to have all the changes integrated into the version of quuly that will be used in the spring

Thinking about how to use AI tools

Learning the fundamental concepts

To analyze quuly, I put lots of fundamental concepts to work:

- understanding state and mutability
- good and bad patterns for dealing with concurrency
- good and bad patterns for dealing with security

I learned these concepts the old fashioned way, starting more than 50 years ago

- writing and debugging code a line at a time, without any AI assistance
- Is that important to building up the mental scaffolding necessary?
- A lot of uncertainty about introducing AI tools in CS1 and CS2 courses

No shortcut to becoming a good software developer

If AI tools allow students to complete CS1 projects in ⅓ the time, they need to be doing at least twice times as many projects

- and not just outsourcing all the technical issues to an AI tool
- maybe turning off the AI tools for a bit, and coding/debugging the old fashioned way


Luke, you switched off your targeting computer. What's wrong?
Nothing. I'm all right.

Learning by coding a lot more

You can use AI tools to handle the boilerplate work, and engage deeply on the technically interesting parts of the projects

- the fun parts
- getting into the flow
- Sometimes, just coding to learn, rather than to build something
- If I planned to be doing a lot of work in React or Go, I wouldn't delegate so much of the code writing to Claude, and would carefully review the code written by Claude to learn from it

Be deliberate about what you outsource

There are some coding/programming problems you should be able to do with paper and pencil

- and 10 years from now, good software developers will still need to be able to do that
- for a start, key algorithms and data structures
  - insertion in sorted linked list or binary trees
  - sorting algorithms
  - using queues, stacks, recursion
  - bitwise operations
- Even if you will never have to write that code in practice

Worrying about mistakes in generated code

AI coding tools make mistakes

a lot of them, and most are not subtle.

The sooner you catch the mistake, the better

Many ways to catch mistakes. What are going to be most effective?

- careful code review (by you, or by another tool)
- Really good and comprehensive testing
- Having a good conceptual understanding of what the code should be doing
- adding automated checks for consistency, preconditions, etc
- logging of errors/warnings/issues, ways to surface those

For me and quuly

There are only a limited number of things I can check in React and go code

- e.g., is this redis/db transaction going to be atomic

quuly had almost no tests

- Added some unit tests, also added e2e testing using playwright

Not much documented in terms of invariants, properties, etc.

    e.g., who has permission to update a conversation

Feeling better about quality of code after changes, even without reviewing most changes

Continuing your journey with AI coding tools

much of what you've learned in this class will be out of date in 6-12 months

maybe 3 months

You will need to keep using and exploring these tools

Many of the things taught in CS courses are fundamental, and won't change significantly in years, or decades

Effective use of AI tools isn't

Goal of this course has been to get you to the point where you are comfortable enough that you can continue to explore this.

- And also appropriately cautious

But you need projects to work on

- Don't know how soon AI coding tools will be allowed in CMSC 400 courses

Getting Project ideas

Solve problems for yourself, sometimes very small ideas

- Simon WIllison has been doing a lot of this,
  - see Useful patterns for building HTML tools
- What have people in this room done?

Work on open source projects

- Can be hard to get connected, really understand the project, have an impact

Internships

Research projects

Work with Derek and/or journalism students

Student organizations

AI Clubs/groups at UMD

Claude builder's club - join - discord

Gemini Hack night

Big Th!nkAI @ UMD

other software building Clubs/groups at UMD

Hack4Impact @ UMD

Startup Shell

Mokhtarzada Hatchery at the University of Maryland

Technica, Bitcamp

XR Club - demo day today, 5-7pm, AV Williams

partial list of CS student orgs

Get funding from Bill

Bill has providing funding to the Sandbox makerspace over the years at UMD

I don't think the department is going to move as fast as I would like in providing course offerings that would offer coverage of the topics in this course to students

Got an idea for how a student organization could help fill that gap?

- Would funding help make that happen? Pitch me