CS 490: Guided Design in Software Engineering

Martin Kellogg

Welcome to CS 490!

Welcome to CS 490!

Today’s agenda:

- What is 490 + course policies and expectations
- About the instructor (aka why you should listen to me)
- In-class activity: background survey
- Survey of the project + other assignments (syllabus day!)
- Start “Code-level design” lecture (if time permits)

Welcome to CS 490!

Today’s agenda:

- **What is 490 + course policies and expectations**
- About the instructor (aka why you should listen to me)
- In-class activity: background survey
- Survey of the project + other assignments (syllabus day!)
- Start “Code-level design” lecture (if time permits)

Course policies

Course policies

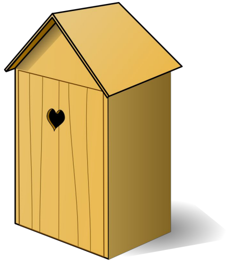
- **Most important**: the first time each class you ask or answer a question, I throw candy at you (sorry for poor aim)

Course policies

- **Most important**: the first time each class you ask or answer a question, I throw candy at you (sorry for poor aim)
  - Let's try it now! **Suggested questions**:
    - Why would you do that?
    - Are you just bribing us to pay attention?
    - Does that actually work?
    - Do even silly questions count?

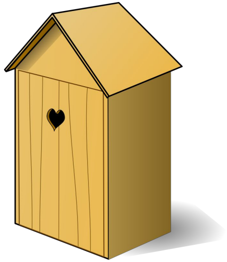
What is CS 490?

What is CS 490? An analogy



= CS 113/114

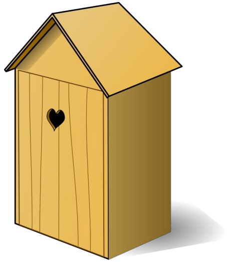
What is CS 490? An analogy



= CS 113/114

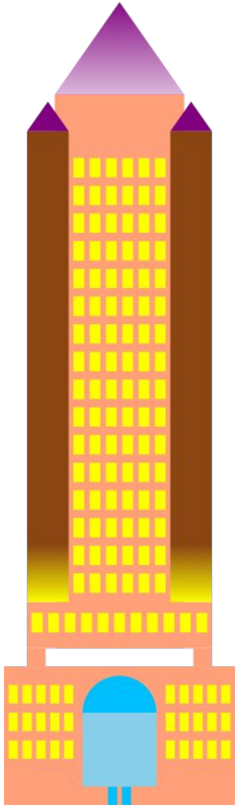
= CS 280/288

What is CS 490? An analogy



= CS 113/114

= CS 280/288



= CS 490

What is CS 490?

- Previous courses were about ***programming***
- A course about ***engineering*** software

What is CS 490?

- Previous courses were about ***programming***
- A course about ***engineering*** software
  - safety and reliability
  - working in a team, including with people with different skillsets
  - non-functional properties and trade-offs
  - architecture and design
  - using your mathematical skills to achieve a practical result
  - building something the right way
  - etc

What is CS 490?

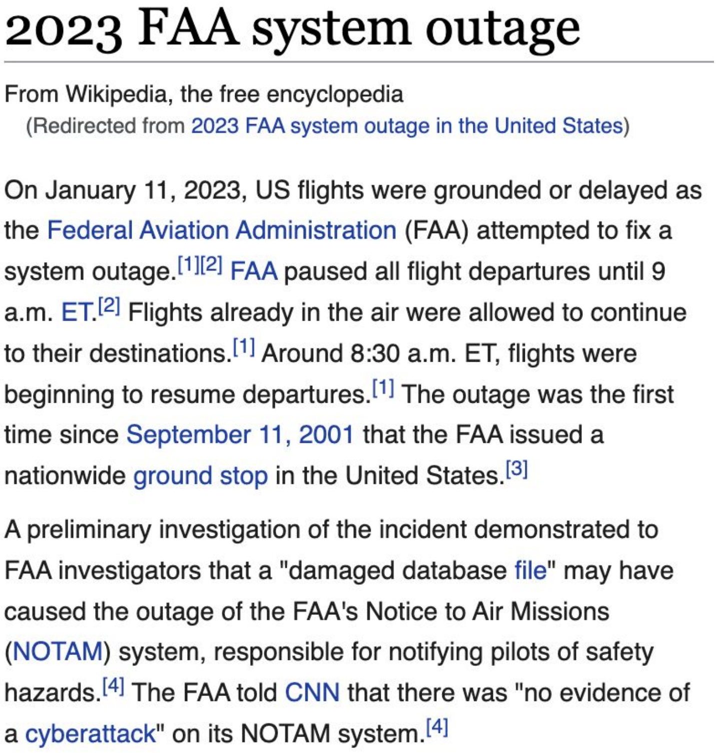
- Previous courses were about ***programming***
- A course about ***engineering*** software

  - safety and reliability
  - working in a team, including with people with different skillsets
  - non-functional properties and trade-offs
  - architecture and design
  - using your mathematical skills to achieve a practical result
  - building something the right way
  - etc

How do these principles apply to programming?

Why does software need to be engineered?

Why does software need to be engineered?



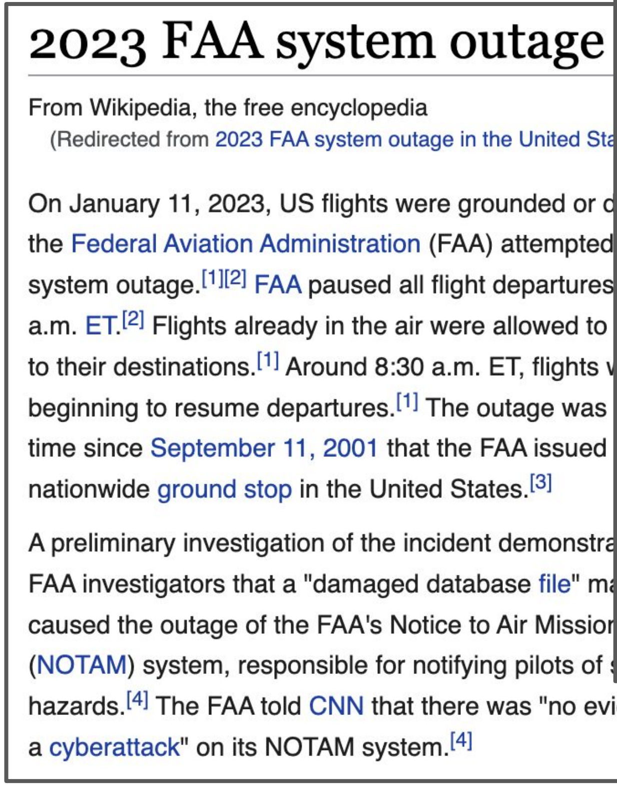
2023 FAA system outage

From Wikipedia, the free encyclopedia
(Redirected from 2023 FAA system outage in the United States)

On January 11, 2023, US flights were grounded or delayed as the Federal Aviation Administration (FAA) attempted to fix a system outage.[1][2] FAA paused all flight departures until 9 a.m. ET.[2] Flights already in the air were allowed to continue to their destinations.[1] Around 8:30 a.m. ET, flights were beginning to resume departures.[1] The outage was the first time since September 11, 2001 that the FAA issued a nationwide ground stop in the United States.[3]

A preliminary investigation of the incident demonstrated to FAA investigators that a "damaged database file" may have caused the outage of the FAA's Notice to Air Missions (NOTAM) system, responsible for notifying pilots of safety hazards.[4] The FAA told CNN that there was "no evidence of a cyberattack" on its NOTAM system.[4]

Why does software need to be engineered?



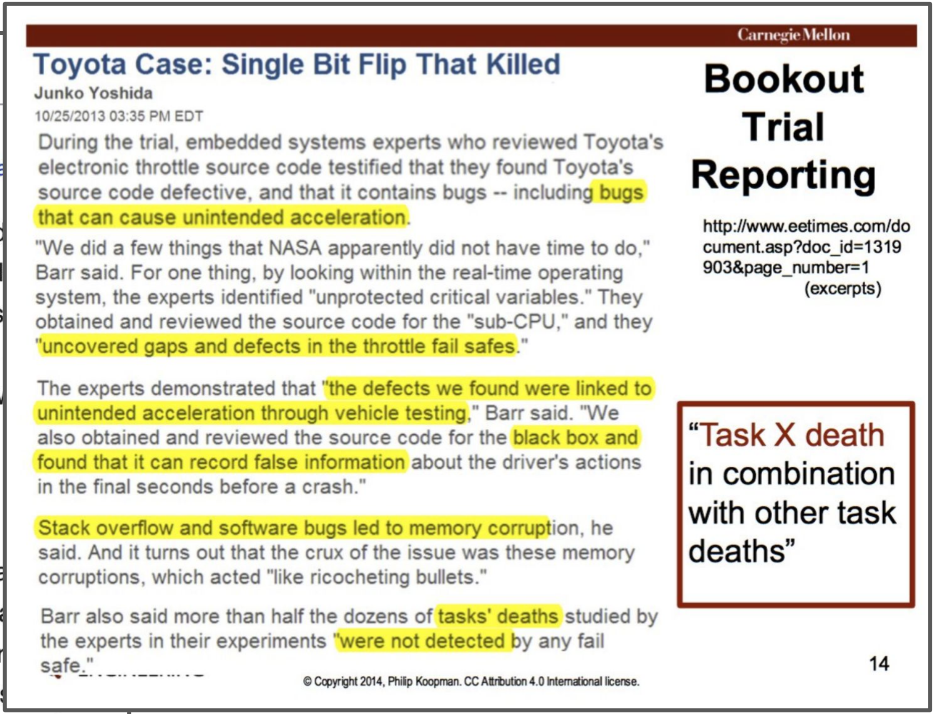
2023 FAA system outage

From Wikipedia, the free encyclopedia

(Redirected from 2023 FAA system outage in the United Sta

On January 11, 2023, US flights were grounded or d the Federal Aviation Administration (FAA) attempted system outage.[1][2] FAA paused all flight departures a.m. ET.[2] Flights already in the air were allowed to to their destinations.[1] Around 8:30 a.m. ET, flights w beginning to resume departures.[1] The outage was time since September 11, 2001 that the FAA issued nationwide ground stop in the United States.[3]

A preliminary investigation of the incident demonstra FAA investigators that a "damaged database file" ma caused the outage of the FAA's Notice to Air Mission (NOTAM) system, responsible for notifying pilots of hazards.[4] The FAA told CNN that there was "no evidence of a cyberattack" on its NOTAM system.[4]



Carnegie Mellon

Toyota Case: Single Bit Flip That Killed

Junko Yoshida

10/25/2013 03:35 PM EDT

During the trial, embedded systems experts who reviewed Toyota's electronic throttle source code testified that they found Toyota's source code defective, and that it contains bugs -- including bugs that can cause unintended acceleration.

"We did a few things that NASA apparently did not have time to do," Barr said. For one thing, by looking within the real-time operating system, the experts identified "unprotected critical variables." They obtained and reviewed the source code for the "sub-CPU," and they "uncovered gaps and defects in the throttle fail safes."

The experts demonstrated that "the defects we found were linked to unintended acceleration through vehicle testing," Barr said. "We also obtained and reviewed the source code for the black box and found that it can record false information about the driver's actions in the final seconds before a crash."

Stack overflow and software bugs led to memory corruption, he said. And it turns out that the crux of the issue was these memory corruptions, which acted "like ricocheting bullets."

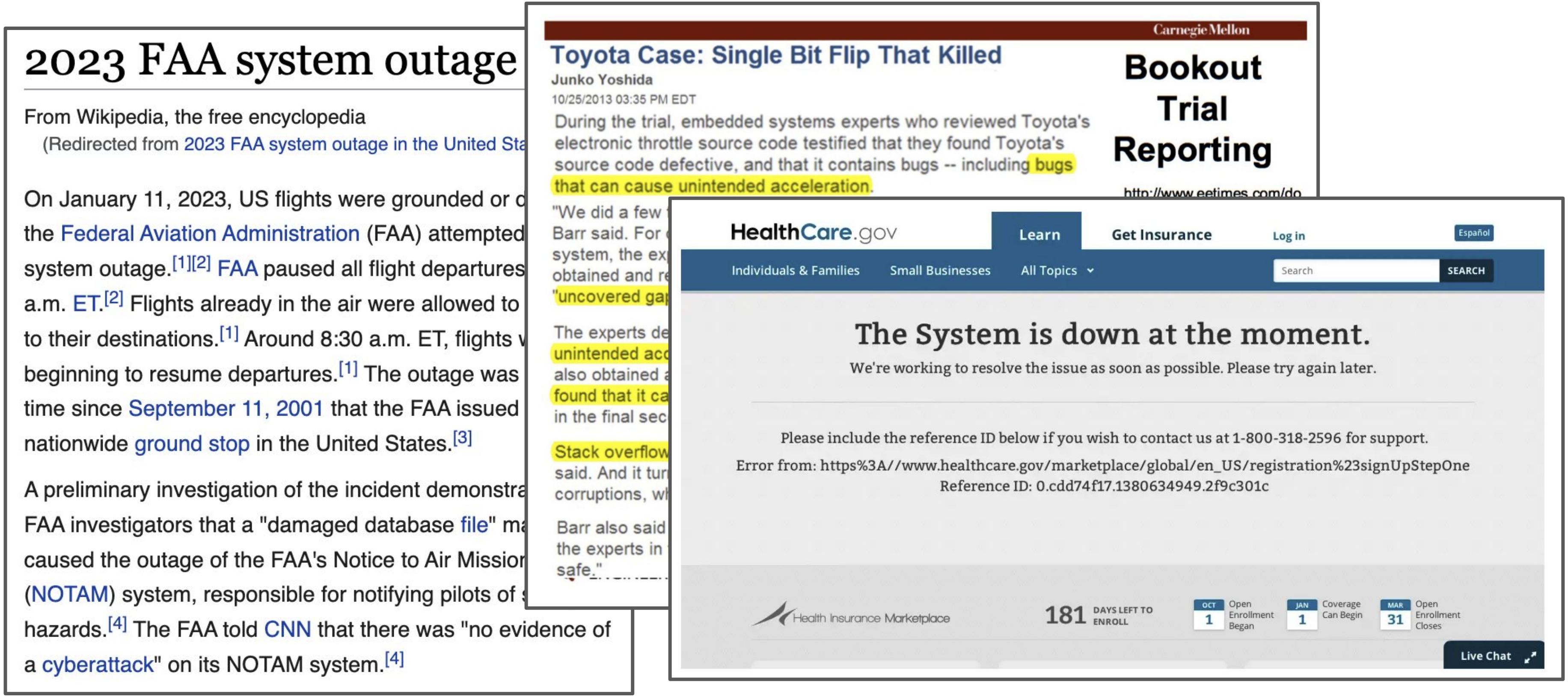
Barr also said more than half the dozens of tasks' deaths studied by the experts in their experiments "were not detected by any fail safe."

Bookout Trial Reporting

http://www.eetimes.com/document.asp?doc_id=1319903&page_number=1 (excerpts)

"Task X death in combination with other task deaths"

Why does software need to be engineered?



2023 FAA system outage

From Wikipedia, the free encyclopedia

(Redirected from 2023 FAA system outage in the United Sta

On January 11, 2023, US flights were grounded or d
the Federal Aviation Administration (FAA) attempted
system outage.[1][2] FAA paused all flight departures
a.m. ET.[2] Flights already in the air were allowed to
to their destinations.[1] Around 8:30 a.m. ET, flights w
beginning to resume departures.[1] The outage was
time since September 11, 2001 that the FAA issued
nationwide ground stop in the United States.[3]

A preliminary investigation of the incident demonstra
FAA investigators that a "damaged database file" ma
caused the outage of the FAA's Notice to Air Mission
(NOTAM) system, responsible for notifying pilots of
hazards.[4] The FAA told CNN that there was "no evidence of
a cyberattack" on its NOTAM system.[4]

Carnegie Mellon

Toyota Case: Single Bit Flip That Killed

Junko Yoshida

10/25/2013 03:35 PM EDT

During the trial, embedded systems experts who reviewed Toyota's electronic throttle source code testified that they found Toyota's source code defective, and that it contains bugs -- including bugs that can cause unintended acceleration.

"We did a few t
Barr said. For
system, the ex
obtained and re
"uncovered gap

The experts de
unintended acc
also obtained a
found that it ca
in the final sec

Stack overflow
said. And it tur
corruptions, wh

Barr also said
the experts in
safe."

Bookout Trial Reporting

http://www.eetimes.com/do

HealthCare.gov

Learn | Get Insurance | Log in | Español

Individuals & Families | Small Businesses | All Topics | Search | SEARCH

The System is down at the moment.

We're working to resolve the issue as soon as possible. Please try again later.

Please include the reference ID below if you wish to contact us at 1-800-318-2596 for support.

Error from: https%3A//www.healthcare.gov/marketplace/global/en_US/registration%23signUpStepOne
Reference ID: 0.cdd74f17.1380634949.2f9c301c

Health Insurance Marketplace

181 DAYS LEFT TO ENROLL

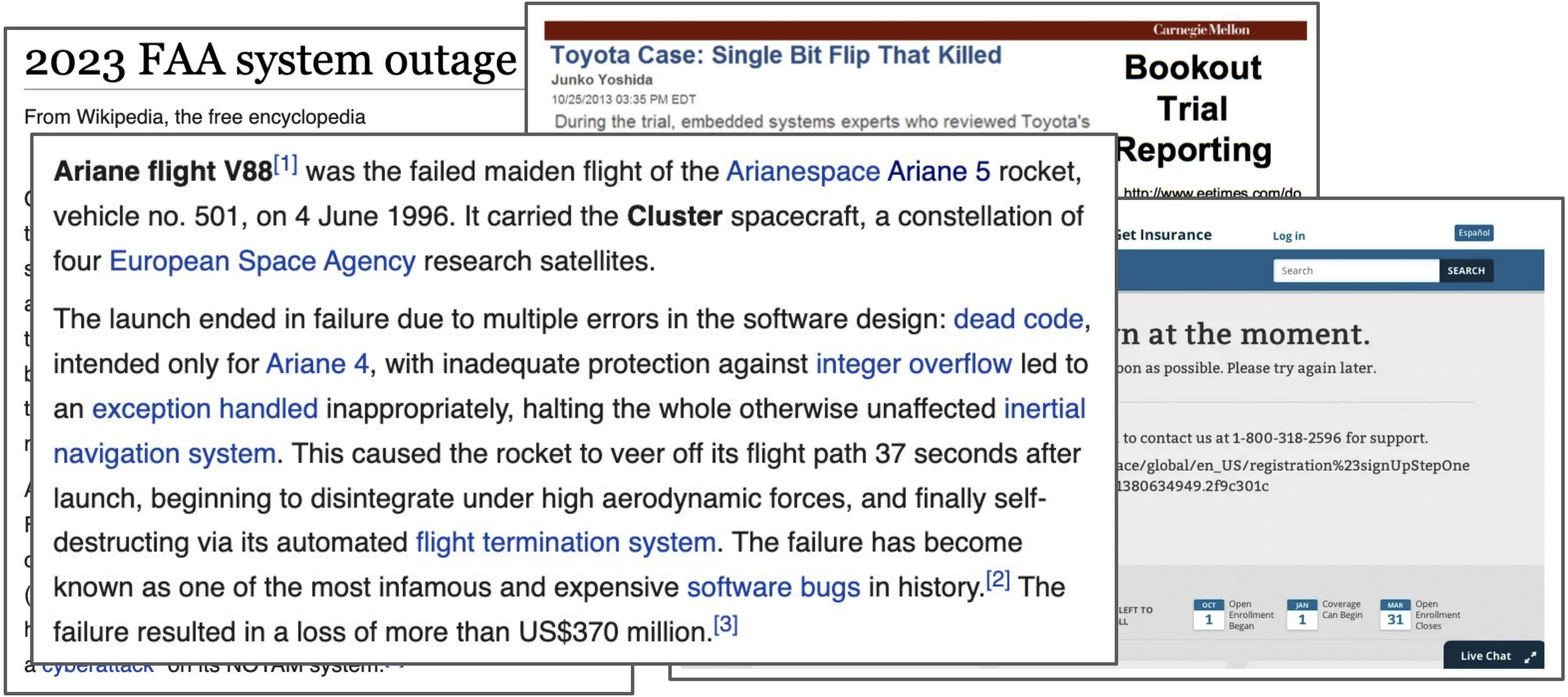
OCT 1 Open Enrollment Began

JAN 1 Coverage Can Begin

MAR 31 Open Enrollment Closes

Live Chat

Why does software need to be engineered?



2023 FAA system outage

From Wikipedia, the free encyclopedia

a cyberattack on its NOTAM system.

Toyota Case: Single Bit Flip That Killed

Junko Yoshida

10/25/2013 03:35 PM EDT

During the trial, embedded systems experts who reviewed Toyota's

Bookout Trial Reporting

http://www.eetimes.com/do

Ariane flight V88[1] was the failed maiden flight of the Arianespace Ariane 5 rocket, vehicle no. 501, on 4 June 1996. It carried the **Cluster** spacecraft, a constellation of four European Space Agency research satellites.

The launch ended in failure due to multiple errors in the software design: dead code, intended only for Ariane 4, with inadequate protection against integer overflow led to an exception handled inappropriately, halting the whole otherwise unaffected inertial navigation system. This caused the rocket to veer off its flight path 37 seconds after launch, beginning to disintegrate under high aerodynamic forces, and finally self-destructing via its automated flight termination system. The failure has become known as one of the most infamous and expensive software bugs in history.[2] The failure resulted in a loss of more than US$370 million.[3]

n at the moment.

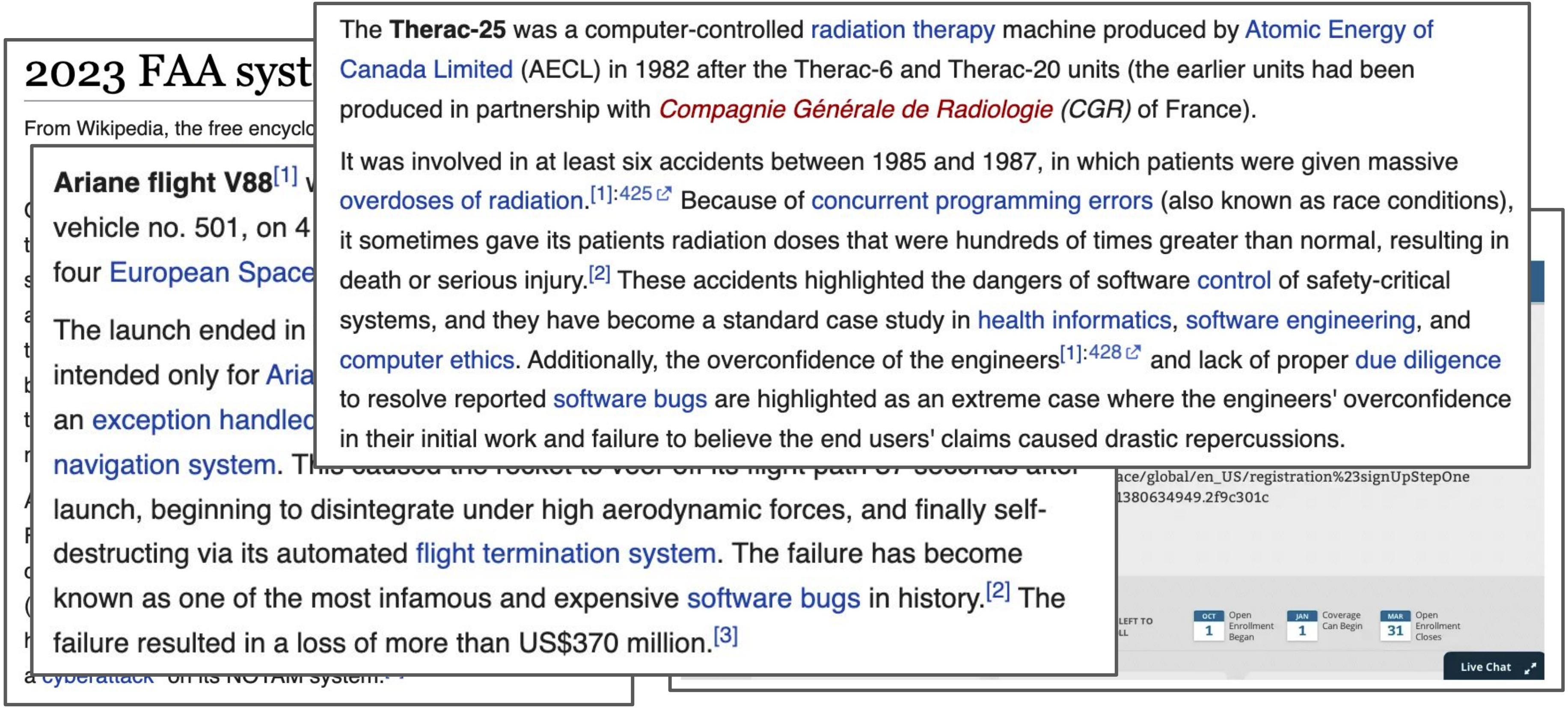
oon as possible. Please try again later.

to contact us at 1-800-318-2596 for support.

ace/global/en_US/registration%23signUpStepOne

1380634949.2f9c301c

Why does software need to be engineered?



The **Therac-25** was a computer-controlled radiation therapy machine produced by Atomic Energy of Canada Limited (AECL) in 1982 after the Therac-6 and Therac-20 units (the earlier units had been produced in partnership with *Compagnie Générale de Radiologie (CGR)* of France).

It was involved in at least six accidents between 1985 and 1987, in which patients were given massive overdoses of radiation.[1]:425 Because of concurrent programming errors (also known as race conditions), it sometimes gave its patients radiation doses that were hundreds of times greater than normal, resulting in death or serious injury.[2] These accidents highlighted the dangers of software control of safety-critical systems, and they have become a standard case study in health informatics, software engineering, and computer ethics. Additionally, the overconfidence of the engineers[1]:428 and lack of proper due diligence to resolve reported software bugs are highlighted as an extreme case where the engineers' overconfidence in their initial work and failure to believe the end users' claims caused drastic repercussions.

Ariane flight V88[1] w

vehicle no. 501, on 4

four European Space

The launch ended in

intended only for Aria

an exception handle

navigation system. This caused the rocket to veer off its flight path 37 seconds after launch, beginning to disintegrate under high aerodynamic forces, and finally self-destructing via its automated flight termination system. The failure has become known as one of the most infamous and expensive software bugs in history.[2] The failure resulted in a loss of more than US$370 million.[3]

2023 FAA syst

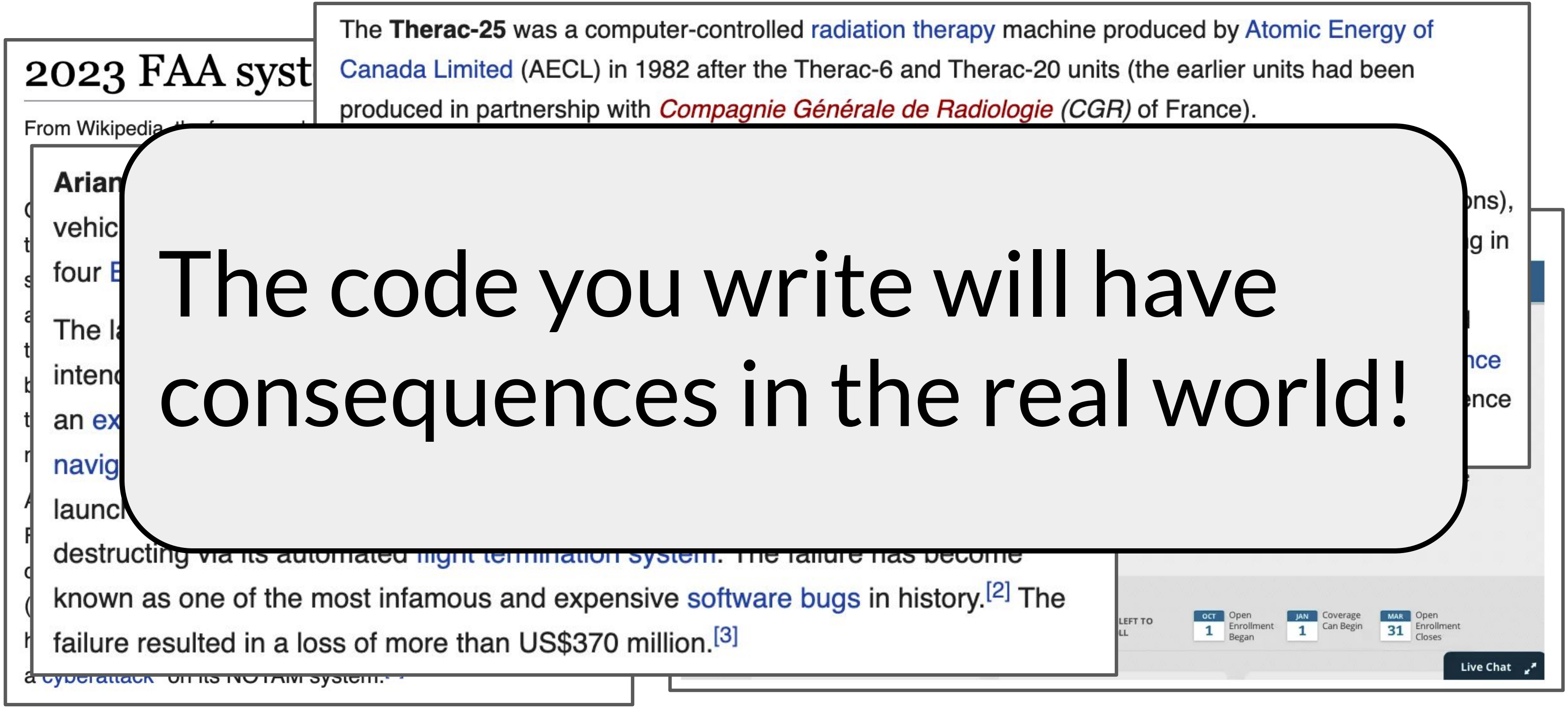
From Wikipedia, the free encyclo

a cyberattack on its NOTAM system.

ace/global/en_US/registration%23signUpStepOne

1380634949.2f9c301c

Why does software need to be engineered?



2023 FAA syst

From Wikipedia, th

The **Therac-25** was a computer-controlled radiation therapy machine produced by Atomic Energy of Canada Limited (AECL) in 1982 after the Therac-6 and Therac-20 units (the earlier units had been produced in partnership with *Compagnie Générale de Radiologie (CGR)* of France).

Arian

vehic

four E

The la

intend

an ex

navig

launc

destructing via its automated flight termination system. The failure has become known as one of the most infamous and expensive software bugs in history.[2] The failure resulted in a loss of more than US$370 million.[3]

a cyberattack on its NOTAM system.

ons),

g in

nce

ence

The code you write will have consequences in the real world!

My expectations

- You know how to ***program***

My expectations

- You know how to ***program***
  - you can write code
  - you can program against an English specification
  - you can read code and figure out what it does
  - you can teach yourself a new programming language
  - you can debug code that's not behaving like you expect
  - you can install software yourself + do basic troubleshooting
  - when you get stuck, you know how to google for answers

My expectations

- You know how to ***program***
  - you can write code
  - you can program against an English specification
  - you can read code and figure out what it does
  - you can teach yourself a new programming language
  - you can debug code that's not behaving like you expect
  - you can install software yourself + do basic troubleshooting
  - when you get stuck, you know how to **google** for answers

The ability to solve problems yourself with just a search engine is a ***critical*** skill for a software engineer!

My expectations

- You know how to ***program***
- Professionalism

My expectations

- You know how to ***program***
- Professionalism
- Participation

CS 490 goals

Officially the following:

- Students will be able to explain the major theories and methods applicable to professional software engineering.
- Students will be able to design, implement and evaluate a computer based system to meet desired needs.
- Students will be able to function effectively on a team to accomplish a goal.
- Students will be able to use current techniques, skills and tools necessary for computing practice.

CS 490 goals

Officially the following:

super vague!

- **Students will be able to explain the major theories and methods applicable to professional software engineering.**
- Students will be able to design, implement and evaluate a computer based system to meet desired needs.
- Students will be able to function effectively on a team to accomplish a goal.
- **Students will be able to use current techniques, skills and tools necessary for computing practice.**

CS 490 goals

Officially the following:

- Students will be able to explain the major theories and methods applicable to professional software engineering.
- **Students will be able to design, implement and evaluate a computer based system to meet desired needs.**
- **Students will be able to function effectively on a team to accomplish a goal.**
- Students will be able to use current techniques, skills and tools necessary for computing practice.

course project!

CS 490 goals

Officially the following:

- Students will be able to explain the major theories and methods applicable to professional software engineering.
- Students will be able to design, implement and evaluate a computer based system to meet desired needs.
- Students will be able to function effectively on a team to accomplish a goal.
- Students will be able to use current techniques, skills and tools necessary for computing practice.

My goals for you:

- Students will be able to assess the **quality of software engineering** being done at some future workplace
- Students will be **competent software engineers** that I wouldn't be worried about hiring

Welcome to CS 490!

Today’s agenda:

- What is 490 + course policies and expectations
- **About the instructor (aka why you should listen to me)**
- In-class activity: background survey
- Survey of the project + other assignments (syllabus day!)
- Start “Code-level design” lecture (if time permits)

Who am I?

- NJIT assistant professor since 2022
- Previously:
  - PhD at University of Washington (Seattle) until June 2022
  - BS at University of Virginia (Charlottesville) in 2016

Who am I?

- NJIT assistant professor since 2022
- Previously:
  - PhD at University of Washington (Seattle) until June 2022
  - BS at University of Virginia (Charlottesville) in 2016

I'm an academic, not a professional software engineer

So what do I know about software engineering?

So what do I know about software engineering?

- My research area is in software engineering
  - more specifically, static analysis design (“compilers”)

So what do I know about software engineering?

- My research area is in software engineering
    - more specifically, static analysis design (“compilers”)
- ~25% of my PhD spent embedded at AWS
    - two co-authored publications
    - my analysis tools deployed on > 70M lines of AWS code

So what do I know about software engineering?

- My research area is in software engineering
    - more specifically, static analysis design (“compilers”)
- ~25% of my PhD spent embedded at AWS
    - two co-authored publications
    - my analysis tools deployed on > 70M lines of AWS code
- My lab is one of the few in the world to take SE seriously when writing research code
    - Inherited from my PhD advisor, who employed 3 SDEs concurrently while I was a student!

Our TAs

- Tiffany and Nathan took this course with me in Au24 (and did well)
    - they know covey.town well and have experienced this course “in your shoes”

Our TAs

- Tiffany and Nathan took this course with me in Au24 (and did well)
  - they know covey.town well and have experienced this course “in your shoes”
- Each TA will have two office hours each week
  - Times/locations TBD, will be announced on Discord

Our TAs

- Tiffany and Nathan took this course with me in Au24 (and did well)
  - they know covey.town well and have experienced this course “in your shoes”
- Each TA will have two office hours each week
  - Times/locations TBD, will be announced on Discord
- TA office hours are where you should go for help with covey.town
  - they know the system better than I do at this point

Office hours

- My office hours are Thursday, 1-2pm
    - if you have questions about lecture contents, my OH are best

Office hours

- My office hours are Thursday, 1-2pm
    - if you have questions about lecture contents, my OH are best
- You can also ask on Discord
    - both the TAs and I will be monitoring Discord, so you can ask any questions you’d like there

Office hours

- My office hours are Thursday, 1-2pm
    - if you have questions about lecture contents, my OH are best
- You can also ask on Discord
    - both the TAs and I will be monitoring Discord, so you can ask any questions you'd like there
- There is no YWCC tutoring for CS 490 this semester
    - it was underutilized in the past, anyway

Welcome to CS 490!

Today's agenda:

- What is 490 + course policies and expectations
- About the instructor (aka why you should listen to me)
- **In-class activity: background survey**
- Survey of the project + other assignments (syllabus day!)
- Start "Code-level design" lecture (if time permits)

Break: background survey

https://forms.gle/UkTudX3Pw8xgxzv17

Welcome to CS 490!

Today’s agenda:

- What is 490 + course policies and expectations
- About the instructor (aka why you should listen to me)
- In-class activity: background survey
- **Survey of the project + other assignments (syllabus day!)**
- Start “Code-level design” lecture (if time permits)

A brief tour through the course website

- https://web.njit.edu/~mjk76/teaching/cs490-au25/

A brief tour through the course website

- https://web.njit.edu/~mjk76/teaching/cs490-au25/
    - Mandatory readings + reading quizzes
    - “Your Choice” readings
    - Individual Project 0: due < 1 week from today
    - My grading: “tough but fair” + curve at the end
    - Collaboration policy (I expect you to use a search engine!)
    - Project structure
    - How to get help
    - Overview of topics

Welcome to CS 490!

Today’s agenda:

- What is 490 + course policies and expectations
- About the instructor (aka why you should listen to me)
- In-class activity: background survey
- Survey of the project + other assignments (syllabus day!)
- **Start “Code-level design” lecture (if time permits)**

Action items for next class

- Individual Project 0
- 4 (short!) mandatory readings: there will be a quiz!
  - Syllabus and IP0 specifications are also fair game
- Make sure you can access all course materials
  - Course website
  - Canvas
  - Discord