Software engineering

Teams code: ajvk6ef

Looking forward

What is the deal?

- I will attempt to condense the relevant parts of my 10 years of industry experience to get you hireable
- Taking this course seriously should give you a massive boost to getting a job or you might even decide to start a company with your teammates
- I will try minimize the academic and industry nonsense
- Attendance is not mandatory for anything
- Your grade is 100% from lab + seminar assignments, retake is a regular exam


YOU CAN'T GIVE HER THAT! IT'S NOT SAFE!
IT'S A SWORD,
THEY'RE NOT MEANT TO BE SAFE.
SHE'S A CHILD!
IT'S EDUCATIONAL.
WHAT IF SHE CUTS HERSELF?
THAT WILL BE AN IMPORTANT LESSON.

What to expect at the labs

- The lab is there to simulate somewhat realistic requirements, scenarios and client nonsense.
- You will be working in a team
- You will decide how to best approach the expanding requirements (so, have fun but expect to rewrite your code occasionally)
- Your team will start from half a semigroup (6-7 students) and will slowly expand to encompass the entire group by the end of the semester
- Expect the greatest challenge to be other people not technology
- You will use C# + .NET
- Teaching assistants will take the roll of clients, to whom you demo your work
- Forming teams is codified in the lab rules

What to expect at the courses

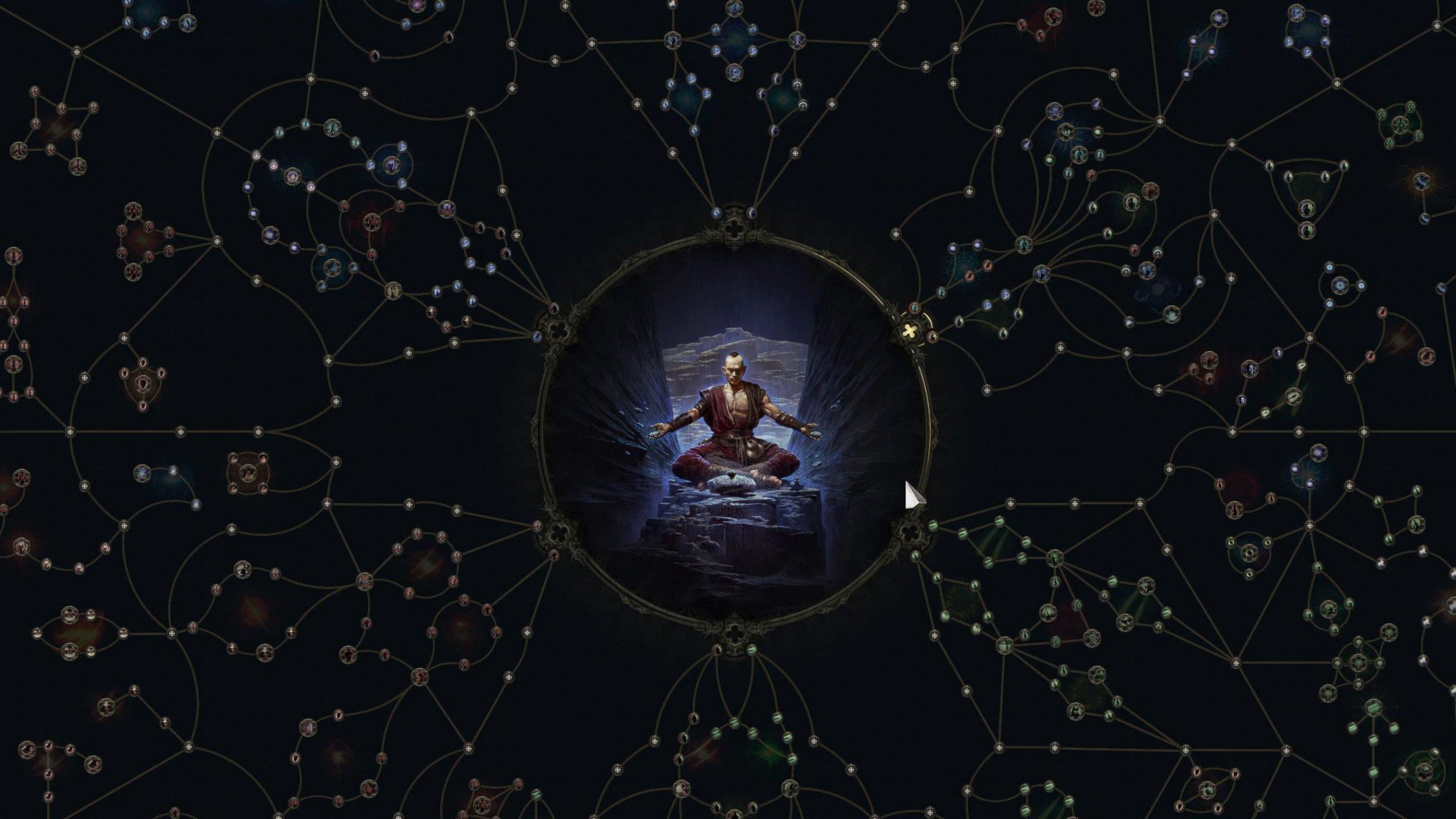
- The course is there to show you the best practices, that people have come up with, to deal with the nonsense of real-life software development
- A.K.A. how NOT to make their working life a living hell
- I will take you through the different phases of software development from: requirements, design, implementation, testing, deployment, review, to maintenance
- We will also talk architectures, CV, interview strategies and more
- I expect you to ask questions and take notes

What to expect at the seminar

- The seminar is there because I could not get it transformed into labs
- They will be different from regular seminars in that:
  - We will not be repeating what you heard at the course
  - We will not be solving your assignments for you
- At the first one you will break down the problem that you will pick in a few minutes, into roughly equal parts for each team
- You will be solving something during the seminar, and get a grade for it at the end, that is related to your assignment but not something you would be presenting at the demo. E.g. team git use at the seminar and implementation at the demo

If you are not convinced

- At the end of the second course if you are not convinced that you want to participate, then seek me out to see what other options there are for you to pass.
- This in not available to you if this course is optional for you or you are retaking this course



Requirements gathering

Requirements

Functional

- Describe what the system should do, independent from the implementation. E.g.:
  - User must be able to log in with their username and password
  - User can view previous purchases

Non-functional

- Constraints, such as performance, usability, reliability, and security. E.g.:
  - The system must respond to user actions under 1 second
  - All transactions must be encrypted with industry standard encryption

Elicitation Techniques

- **Interviews**: Directly talking to stakeholders to understand their needs and expectations.
- **Surveys/Questionnaires**: Collecting a large volume of information from many stakeholders, useful when direct interviews are not feasible.
- **User Observation**: Watching end-users interact with current systems or perform their jobs to identify unarticulated needs.
- **Workshops**: Bringing together multiple stakeholders to discuss requirements, which is useful for reconciling conflicting requirements.
- **Prototyping**: Developing a quick and rough version of the system to help stakeholders visualize and refine their requirements.
- **Document Analysis**: Reviewing existing documentation and system artifacts to identify requirements implicit in current workflows or systems.

Clear / Unambiguous

- "The application should have a nice login screen.“
- "The application shall display a login form on the home page with two clearly labelled fields: 'Username' and 'Password'."

Valid / Correct

- "The application should help users recover their credentials."

- "The application shall allow users to reset their password by entering their registered email address, after which a password reset link will be sent to that email."

Complete

- "The application shall allow users to register by providing a username, password, and email."

- "The application shall allow users to register by providing:
  - A unique username
  - A password with a minimum of 8 characters, including at least one uppercase letter, one lowercase letter, one numeral, and one special character
  - A valid email address
    If any input fails validation, the system shall display an error message specifying the field and the issue."

Consistent

- Company standard requires all user interfaces to have a logo in upper right corner of the screen.
- A user interface requirement specifies that the logo must be at the bottom center of the screen.

Verifiable

- “The application shall be fast.”
- “The application shall load the main page within 2 seconds on a 4G mobile connection for 95% of requests.”

Feasible

- “The application shall support an unlimited number of concurrent users.”

- “The application shall support up to 1,000 concurrent users during peak hours on the current hardware configuration.”

All together

- The system shall allow registered users to update their profile information, specifically first name, last name, email address, and phone number, via the account settings page. The update operation must complete within 2 seconds on a broadband connection (minimum 25 Mbps). The system shall validate that the email address is in a proper format and that the phone number consists solely of digits. If any field is invalid, a specific error message indicating the problematic field and the nature of the error shall be displayed. This functionality will be integrated with the existing user management module, which has been reviewed and confirmed to support these operations within current resource constraints.

- GIVEN: the program runs without errors AND
- GIVEN: the input file is at c:\input\log.txt
- THEN: I can view the errors grouped by severity
- Acceptance criteria:
  - It’s done when…

Time to pick a project

Famous disasters

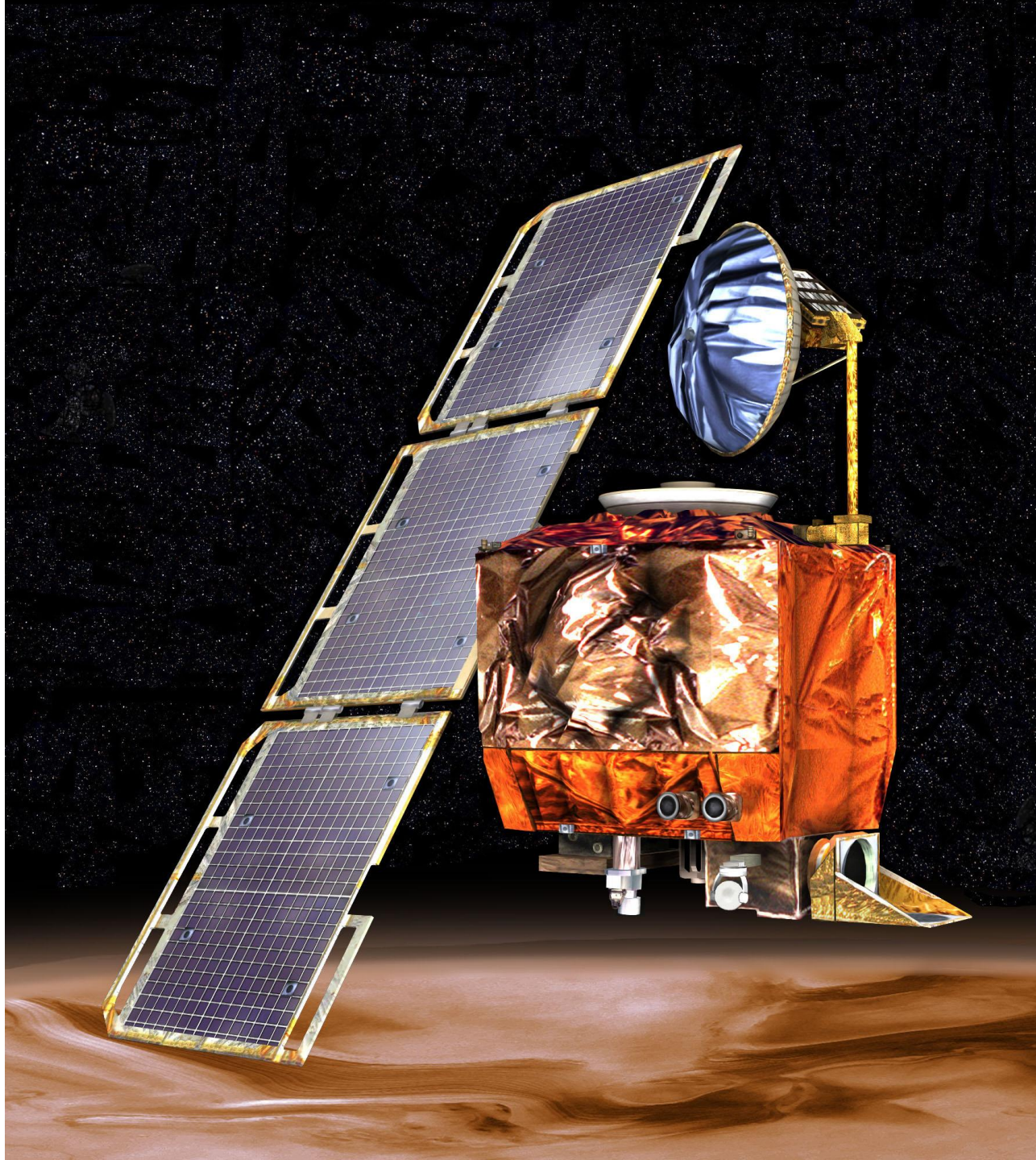
Therac-25 (1985-1987)

Canada’s Therac-25 radiation therapy machine malfunctioned and delivered lethal radiation doses to patients.

Because of a race condition, a technician could accidentally configure Therac-25 so the electron beam would fire in high-power mode without the proper patient shielding.

NASA’s Mars Climate Orbiter (1998)

The Mars Climate Orbiter burned up after getting too close to the surface of Mars. According to the investigation report, the ground control software produced by Lockheed Martin used imperial measurements, while the software onboard, produced by NASA, was programmed with SI metric units.



Knight’s $440M in bad trades (2012)

Knight Capital Group had invested in new trading software that was supposed to help them make a killing on the stock markets. Several software errors combined to send Knight on a crazy buying spree, spending more than $7 billion on 150 different stocks in under 30 minutes. This wiped 75% off the value of one the biggest capital groups in the world.

ESA Ariane 5 Flight V88 (1996)

Just 36 seconds after its maiden launch, the rocket engines failed due to the engineers reusing incompatible code from Ariane 4 and a conversion error from 64-bit to 16-bit data. The failure resulted in a $370 million loss for the ESA.

World War III… Almost (1983)

The Soviet early warning system falsely indicated the United States had launched five ballistic missiles. Fortunately, the Soviet duty officer had a “funny feeling in my gut” and reasoned if the U.S. was really attacking, they would launch more than five missiles, so he reported the apparent attack as a false alarm.

Cause: A bug in the Soviet software failed to filter out false missile detections caused by sunlight reflecting off cloud-tops.