CMPT 373
Software Development Methods

Introduction

Nick Sumner
wsumner@sfu.ca

Introduction

- Who am I?
  - Nick Sumner (wsumner@sfu.ca)
  - Research Faculty

Introduction

- Who am I?
  - Nick Sumner (wsumner@sfu.ca)
  - Research Faculty
- Who is your TA?
  - Sami Ma, Bhavya Sankhla

Introduction

- Who am I?
  - Nick Sumner (wsumner@sfu.ca)
  - Research Faculty
- Who is your TA?
  - Sami Ma, Bhavya Sankhla
- What is the course website?
  - http://www.cs.sfu.ca/~wsumner/teaching/373/
  - OR: just search for "CMPT 373 sumner"

Introduction

- Who am I?
  - Nick Sumner (wsumner@sfu.ca)
  - Research Faculty
- Who is your TA?
  - Sami Ma, Bhavya Sankhla
- What is the course website?
  - http://www.cs.sfu.ca/~wsumner/teaching/373/
  - OR: just search for “CMPT 373 sumner”
- Where can you discuss course issues?
  - CourSys

First Day Homework

- Exercise 0
  - Basic competence in C++ and programming
  - This should be trivial if you meet the prerequisites for this class

First Day Homework

- Exercise 0
  - Basic competence in C++ and programming
  - This should be trivial if you meet the prerequisites for this class
  - Get in the habit of doing exercises when they are released.
    If you do not, you will struggle.

First Day Homework

- Exercise 0
  - Basic competence in C++ and programming
  - This should be trivial if you meet the prerequisites for this class
  - Get in the habit of doing exercises when they are released.
    If you do not, you will struggle.
  - This can also serve as a litmus test.
    If it isn't trivial, increase your workload expectations

First Day Homework

- Exercise 0
  - Basic competence in C++ and programming
  - This should be trivial if you meet the prerequisites for this class
  - Get in the habit of doing exercises when they are released.
    If you do not, you will struggle.
  - This can also serve as a litmus test.
    If it isn't trivial, increase your workload expectations
- Background Survey
  - Anonymous google forms survey on your competences from other classes.

First Day Homework

- Exercise 0
  - Basic competence in C++ and programming
  - This should be trivial if you meet the prerequisites for this class
  - Get in the habit of doing exercises when they are released.
    If you do not, you will struggle.
  - This can also serve as a litmus test.
    If it isn’t trivial, increase your workload expectations
- Background Survey
  - Anonymous google forms survey on your competences from other classes.
- Videos!
  - The course schedule will have links to excellent videos + external resources

What is this course?

- My perspective… hands on experience

What is this course?

- My perspective… hands on experience
  - **workflows**
  - tools
  - project management
  - writing better code
  - dealing with a (possibly troublesome) customer
  - dealing with (and avoiding) problems

What is this course?

- My perspective… hands on experience
  - workflows
  - **tools**
  - project management
  - writing better code
  - dealing with a (possibly troublesome) customer
  - dealing with (and avoiding) problems

What is this course?

- My perspective… hands on experience
  - workflows
  - tools
  - **project management**
  - writing better code
  - dealing with a (possibly troublesome) customer
  - dealing with (and avoiding) problems

What is this course?

- My perspective… hands on experience
  - workflows
  - tools
  - project management
  - **writing better code**
  - dealing with a (possibly troublesome) customer
  - dealing with (and avoiding) problems

What is this course?

- My perspective… hands on experience
  - workflows
  - tools
  - project management
  - writing better code
  - **dealing with a (possibly troublesome) customer**
  - dealing with (and avoiding) problems

What is this course?

- My perspective… hands on experience
  - workflows
  - tools
  - project management
  - writing better code
  - dealing with a (possibly troublesome) customer
  - **dealing with (and avoiding) problems**

What is this course?

- My perspective… hands on experience
  - workflows
  - tools
  - project management
  - writing better code
  - dealing with a (possibly troublesome) customer
  - dealing with (and avoiding) problems
- Slightly different than many courses
  - Less emphasis on “getting the right answer”
  - More emphasis on being aware of trade offs & using the right skills

Why take this course?

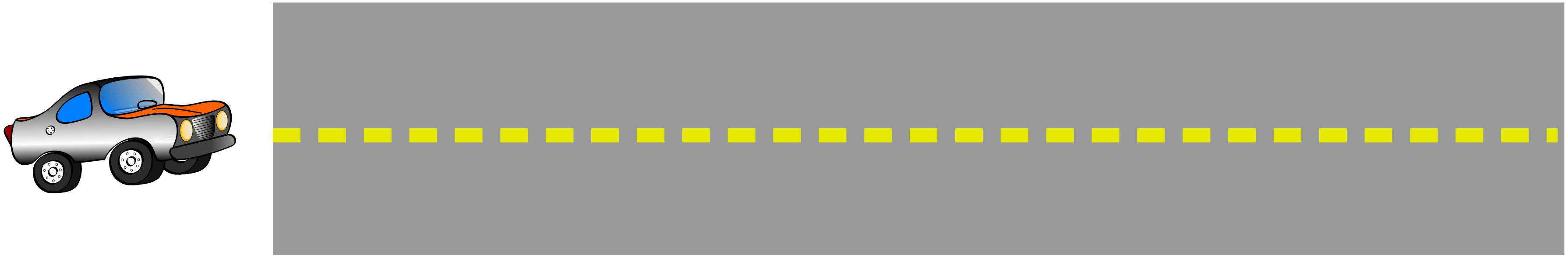
- Software project failure is common(!)

Why take this course?

- Software project failure is common(!)
  - 1/3 – 2/3 of projects (depending on source and definition of failure)

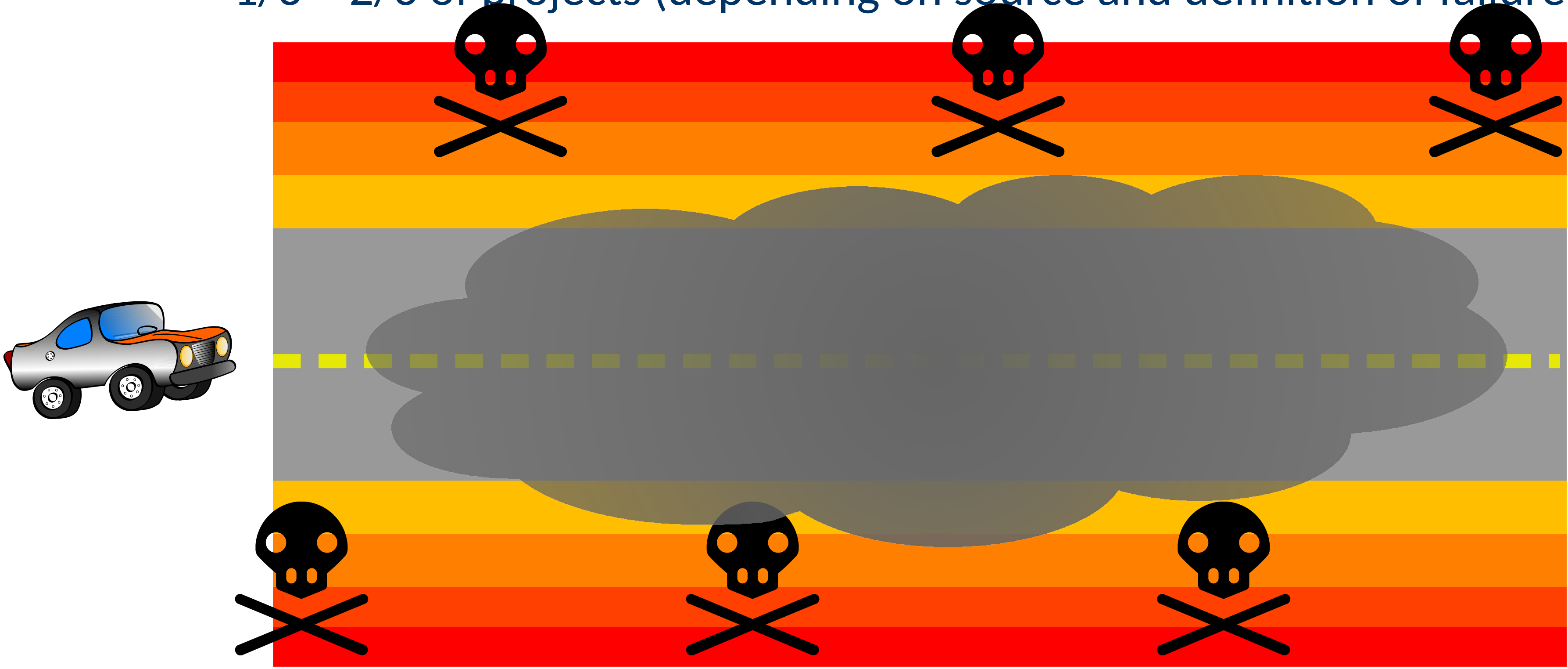
Why take this course?

- Software project failure is common(!)
  - 1/3 – 2/3 of projects (depending on source and definition of failure)



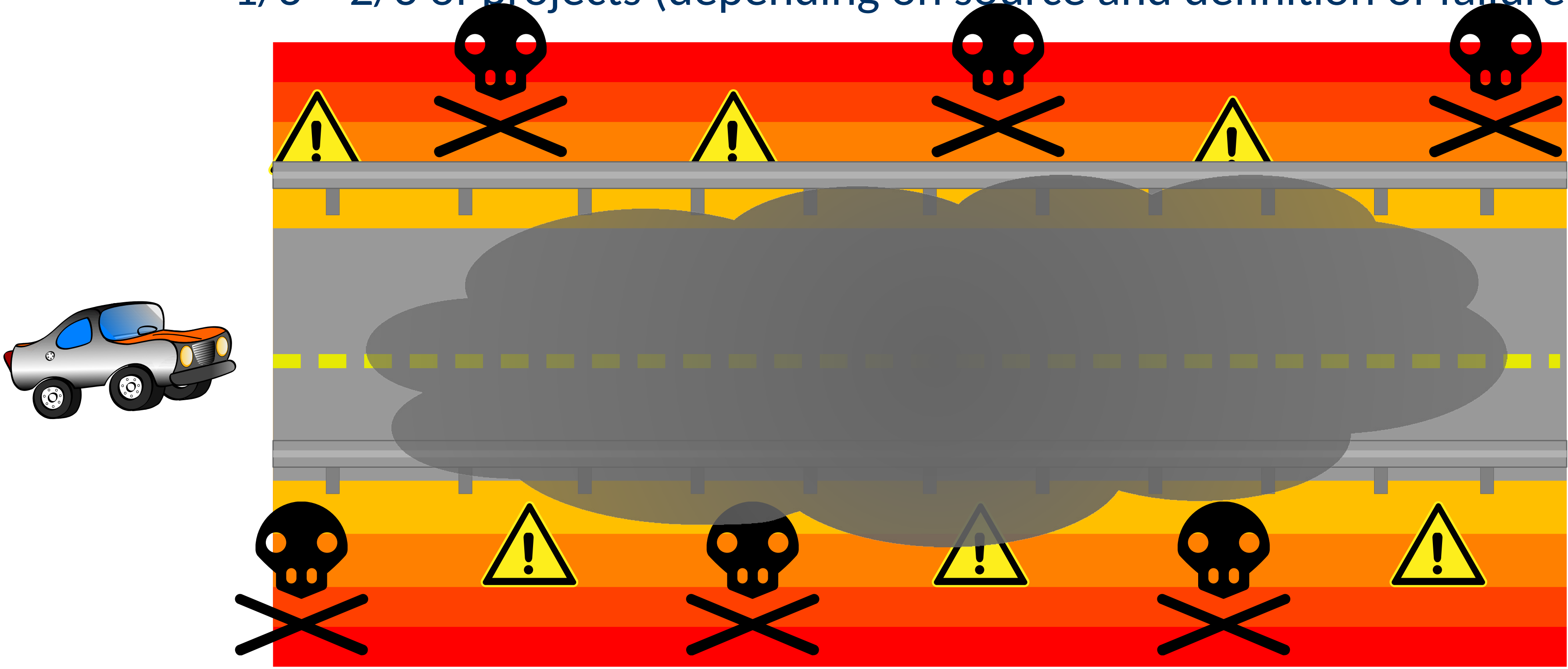
Why take this course?

- Software project failure is common(!)
  - 1/3 – 2/3 of projects (depending on source and definition of failure)



Why take this course?

- Software project failure is common(!)
  - 1/3 – 2/3 of projects (depending on source and definition of failure)



Why take this course?

- Software project failure is common(!)
  - 1/3 – 2/3 of projects (depending on source and definition of failure)
- Most graduates with a CS degree are not ready
  - Software engineering is about *process* and *awareness*
  - Software development is a *craft* that requires practice

Why take this course?

- Software project failure is common(!)
  - 1/3 – 2/3 of projects (depending on source and definition of failure)
- Most graduates with a CS degree are not ready
  - Software engineering is about *process* and *awareness*
  - Software development is a *craft* that requires practice
- Hands on experience yields an advantage
  - You can better understand how to create a product that has value both now and in the future.

What will we be doing?

- On your own
  - Reading
  - Exercises (tools, programming techniques, workflows)

What will we be doing?

- On your own
  - Reading
  - Exercises (tools, programming techniques, workflows)
- In groups
  - One development project with unclear requirements

What will we be doing?

- On your own
  - Reading
  - Exercises (tools, programming techniques, workflows)
- In groups
  - One development project with unclear requirements
- In class
  - Practice the techniques learned
  - Q&A about lecture material
  - Discussions about the reading, tools, programming, term project
  - Meeting with your adversarial customer

Grading

- Breakdown:
  - (10%) Responses to reading
  - (5%) Class discussions & code reviews
  - (10%) Midterm
  - (25%) Exam
  - (25%) Useful contribution to semester project
  - (25%) Programming exercises

Grading

- Breakdown
  - (10%) Responses to reading
  - (5%) Class discussions & code reviews
  - (10%) Midterm
  - (25%) Exam
  - (25%) Useful contribution to semester project
  - (25%) Programming exercises
- Late Policy
  - 3 late days to use throughout the semester
    (on exercises & reading responses)

Reading

- Assigned chunks of reading
  - Often ~100 pages per 1-2 weeks
  - Books are available as e-books in library

Reading

- Assigned chunks of reading
  - Often ~100 pages per 1-2 weeks
  - Books are available as e-books in library
- Responses
  - A 2 page critical reaction to the reading
  - Single spaced
  - Must include 3 units of:
    - A quote, with citation
    - 1-2 paragraphs discussing the quote
  - Relate the material to your own experiences
  - Form an opinion about it, and *justify* it

Reading

- Assigned chunks of reading
  - Often ~100 pages per 1-2 weeks
  - Books are available as e-books in library
- Responses
  - A 2 page critical reaction to the reading
  - Single spaced
  - Must include 3 units of:
    - A quote, with citation
    - 1-2 paragraphs discussing the quote
  - Relate the material to your own experiences
  - Form an opinion about it, and *justify* it
- The reading schedule is already posted

Programming Exercises

- Small programming exercises will provide
  - Focused experience with specific techniques
  - A chance to identify missing skills

Programming Exercises

- Small programming exercises will provide
  - Focused experience with specific techniques
  - A chance to identify missing skills
- Your first exercise on prerequisite skills is available now!

Discussions

- Code Review:

Discussions

- Code Review:
  - Each group will submit 100-200 lines of code each week by Friday, 10pm

Discussions

- Code Review:
  - Each group will submit 100-200 lines of code each week by Friday, 10pm
  - I'll assign members from other groups to review the code
    (I may choose some other code entirely)

Discussions

- Code Review:
  - Each group will submit 100-200 lines of code each week by Friday, 10pm
  - I'll assign members from other groups to review the code
    (I may choose some other code entirely)
  - Individual reviews due by 10pm Tuesdays

Discussions

- Code Review:
  - Each group will submit 100-200 lines of code each week by Friday, 10pm
  - I'll assign members from other groups to review the code
    (I may choose some other code entirely)
  - Individual reviews due by 10pm Tuesdays
  - Submitters & reviewers for ~2 submissions will present on Wednesday.

Discussions

- Code Review:
  - Each group will submit 100-200 lines of code each week by Friday, 10pm
  - I'll assign members from other groups to review the code
    (I may choose some other code entirely)
  - Individual reviews due by 10pm Tuesdays
  - Submitters & reviewers for ~2 submissions will present on Wednesday.
- In class discussions of both code & readings focus thematically on one core issue:

Complexity

Semester project

- You will interact with me as a customer in tutorials

Semester project

- You will interact with me as a customer in tutorials
- The requirements of the project ***will change***

Semester project

- You will interact with me as a customer in tutorials
- The requirements of the project ***will change***
- You will use (and be evaluated in part on) skills from the exercises in the project

Semester project

- You will interact with me as a customer in tutorials
- The requirements of the project ***will change***
- You will use (and be evaluated in part on) skills from the exercises in the project
- Different teams may receive different requirements

Semester project

- You will interact with me as a customer in tutorials
- The requirements of the project ***will change***
- You will use (and be evaluated in part on) skills from the exercises in the project
- Different teams may receive different requirements
- You should expect to *personally* contribute >= 1K quality SLOC in order to pass

Project code policy

All code pushed to a project repository may be viewed, analyzed, and critiqued by all students ***in class***
(even in future semesters).

Project teams

- Assigned teams of up to 8

Project teams

- Assigned teams of up to 8
- Following an informal scrum-like process
  - Each team meeting should involve:
    - Mention of what you did since the last meeting
    - What the present obstacles are to meeting goals
    - A plan for the next meeting

Project teams

- Assigned teams of up to 8
- Following an informal scrum-like process
  - Each team meeting should involve:
    - Mention of what you did since the last meeting
    - What the present obstacles are to meeting goals
    - A plan for the next meeting
- I will act as both customer & coach

Goals

- Writing valuable code as a team
  - Some teammates will write well from the beginning.
  - Some will need help from teammates.

Goals

- Writing valuable code as a team
  - Some teammates will write well from the beginning.
  - Some will need help from teammates.
  - Working together is the only real way.

Goals

- Writing valuable code as a team
  - Some teammates will write well from the beginning.
  - Some will need help from teammates.
  - Working together is the only real way.
  - This is just as true in industry.

Goals

- Writing valuable code as a team
  - Some teammates will write well from the beginning.
  - Some will need help from teammates.
  - Working together is the only real way.
  - This is just as true in industry.
- Manage complexity & change
  - Requirements will change in practice.
  - I will try to change requirements that force design changes.

Goals

- Writing valuable code as a team
  - Some teammates will write well from the beginning.
  - Some will need help from teammates.
  - Working together is the only real way.
  - This is just as true in industry.
- Manage complexity & change
  - Requirements will change in practice.
  - I will try to change requirements that force design changes.
  - Better designs & process will make the transitions easier.

Bad Reasons to Take CMPT 373

- You struggled with earlier classes and want to “make up for it”

Bad Reasons to Take CMPT 373

- You struggled with earlier classes and want to “make up for it”
- Your friends told you it would be easy

Bad Reasons to Take CMPT 373

- You struggled with earlier classes and want to “make up for it”
- Your friends told you it would be easy
- You think that you can let your team do all of your work for you

Bad Reasons to Take CMPT 373

- You struggled with earlier classes and want to “make up for it”
- Your friends told you it would be easy
- You think that you can let your team do all of your work for you
- You want to focus on learning a particular framework, frontend, architecture, …

Bad Reasons to Take CMPT 373

- You struggled with earlier classes and want to “make up for it”
- Your friends told you it would be easy
- You think that you can let your team do all of your work for you
- You want to focus on learning a particular framework, frontend, architecture, …
- You want to focus on web app development

Bad Reasons to Take CMPT 373

- You struggled with earlier classes and want to “make up for it”
- Your friends told you it would be easy
- You think that you can let your team do all of your work for you
- You want to focus on learning a particular framework, frontend, architecture, ...
- You want to focus on web app development
- You want to focus on learning microservices
  - This last point at least makes sense in this course, but it requires understanding distributed systems and is outside out scope

A Good Reason to Take CMPT 373

- You understand the content from earlier courses, *but*

A Good Reason to Take CMPT 373

- You understand the content from earlier courses, *but* you want to make better decisions about
  how and why

A Good Reason to Take CMPT 373

- You understand the content from earlier courses, *but*
  you want to make better decisions about
  how and why
  to develop a solution one way versus another

Expectations

- You will be available to meet during class times and meet with your teammates outside of class

Expectations

- You will be available to meet during class times and meet with your teammates outside of class
- You understand and are comfortable with the basics of C++ (old or modern)

Honesty

- I expect all students to complete their own work

Honesty

- I expect all students to complete their own work
- All students must contribute substantially to their term projects to pass (you will review your teammates)

Honesty

- I expect all students to complete their own work
- All students must contribute substantially to their term projects to pass (you will review your teammates)
- You will not share code or solutions for anything other than the term project

Honesty

- I expect all students to complete their own work
- All students must contribute substantially to their term projects to pass (you will review your teammates)
- You will not share code or solutions for anything other than the term project
  - Even after the semester ends, this counts as academic dishonesty

Honesty

- I expect all students to complete their own work
- All students must contribute substantially to their term projects to pass (you will review your teammates)
- You will not share code or solutions for anything other than the term project
  - Even after the semester ends, this counts as academic dishonesty
  - Don’t be the idiot who fails a class *retroactively* for posting to github…. (seriously. Wow.)

Honesty

- I expect all students to complete their own work
- All students must contribute substantially to their term projects to pass (you will review your teammates)
- You will not share code or solutions for anything other than the term project
  - Even after the semester ends, this counts as academic dishonesty
  - Don’t be the idiot who fails a class *retroactively* for posting to github…. (seriously. Wow.)
  - Your team term project is the only exception.
    If you take it seriously, it can be worth sharing.

So let’s begin the first Q&A!

Q&A

And that’s it for now.

I hope you’re ready for an interesting and collaborative semester.