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?
  - Chandan Kumar Mishra

# Introduction

- Who am I?
  - Nick Sumner (wsumner@sfu.ca)
  - Research Faculty
- Who is your TA?
  - Chandan Kumar Mishra
- 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?
  - Chandan Kumar Mishra
- 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 (https://courses.cs.sfu.ca/2015fa-cmpt-373-d1/discussion/)

# What is this course?

- What have you heard?

# What is this course?

- What have you heard?
- My perspective… hands on experience
  - workflows
  - tools
  - project management
  - dealing with a (possibly troublesome) customer
  - dealing with (and avoiding) problems

# What is this course?

- What have you heard?
- My perspective… hands on experience
  - workflows
  - tools
  - project management
  - 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 & using the right skills

# Why take this course?

- Most software projects fail(!)
  - Up to 85% depending on definition of "*failure*"

# Why take this course?

- Most software projects fail(!)
  - Up to 85% depending on 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?

- Most software projects fail(!)
  - Up to 85% depending on 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 (From 2 books)
  - Exercises with tools

# What will we be doing?

- On your own
  - Reading (From 2 books)
  - Exercises with tools
- In groups / tutorials
  - One development project with unclear requirements

# What will we be doing?

- On your own
  - Reading (From 2 books)
  - Exercises with tools
- In groups / tutorials
  - One development project with unclear requirements
- In class
  - Introduction to tools and techniques
  - Discussions about the reading
  - Discussions about the tools
  - Discussions about code

# Grading

- Subject to change as necessary
- Breakdown:
  - (15%) Responses to reading
  - (15%) Participation in class discussions
  - (50%) Useful contribution to semester project
  - (20%) Exercises

# Reading

- Assigned chunks of reading
  - Often ~200 pages per 1-2 weeks
  - Both books are available as e-books in library

# Reading

- Assigned chunks of reading
  - Often ~200 pages per 1-2 weeks
  - Both books are available as e-books in library
- Responses
  - A 2-3 page critical reaction to the reading
  - Single space
  - 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 ~200 pages per 1-2 weeks
  - Both books are available as e-books in library
- Responses
  - A 2-3 page critical reaction to the reading
  - Single space
  - 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
- First assignment posted after class

# 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
- All requirements gathering & customer interaction happens in the tutorials

# 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
- All requirements gathering & customer interaction happens in the tutorials
- Different teams (of 8) may receive different requirements

# Project code policy

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

# Project teams

- Assigned teams of 8
- Following a scrum-like process
  - Each tutorial meeting will involve:
    - Discussion 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 during tutorials

# Discussions

- In class discussions of both code & readings focus thematically on one issue:

Complexity