Introduction
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?
Introduction

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

• Who is your TA?
  –

• 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?
  -

- What is the course website?
  - OR: just search for “CMPT 373 sumner”

- Where can you discuss course issues?
  - CourSys
    (Link in description)
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**
  - Video lectures
  - Reading
  - Exercises (tools, programming techniques, workflows)
What will we be doing?

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

- **In groups**
  - One development project with unclear requirements
What will we be doing?

- **On your own**
  - Video lectures
  - 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
  - (10%) Class discussions & code reviews
  - (25%) Exam
  - (25%) Useful contribution to semester project
  - (30%) Programming exercises
Grading

- **Breakdown**
  - (10%) Responses to reading
  - (10%) Class discussions & code reviews
  - (25%) Exam
  - (25%) Useful contribution to semester project
  - (30%) Programming exercises

- **Late Policy**
  - 3 late days to use throughout the semester (on exercises & reading responses)
Reading

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

• Assigned chunks of reading
  – Often ~200 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
Assignments of reading
- Often ~200 pages per 1-2 weeks
- Books are available as e-books in the 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

First assignment posted after class
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 Thursday.
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 Thursday.

• 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
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 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
Project teams

- Assigned teams of up to 8
- Following an informal scrum-like process
  - Each team 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
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.
• And that’s it for now.

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