

COSC 310: Software Engineering

Dr. Bowen Hui

University of British Columbia
Okanagan

Real World Applications

- Undefined or changing scope
- Different type of jobs and responsibilities
 - Program managers
 - Project managers
 - Business analysts
 - System analysts
 - Software architects
 - Programmers (teams of 20+)
 - Testers
 - Quality Assurance

Major Challenges

- Scope, budget, timeline management
- Communicating in a common “language”
- Working with people

School vs. Real World Projects

	Class Project	Industry Project
Time:	2-4 weeks	Months, Years
Staff:	1-4	SMEs: 2-5 Large: 10-100s
LOC:	Hundreds	Millions
Capital at Stake:	\$0	\$\$\$ Millions

Why SE is Important

- Required for large scale, quality development
- A professional developer must be able to:
 - **Communicate with others** – learn design methods, communicate with non-technical users
 - **Work with others** – coordination, collaboration
 - **Work effectively** – select and use appropriate tools
 - **Strive for improvement** – monitor performance and improve process

When Things Go Wrong

- Class? Lose marks, fail projects, fail class
- Industry
 - Lose money
 - Stressed employees
 - Lose customers' trust
- Examples
 - An order purchasing system (UK, 2013) could not handle online orders: \$15M loss in the first 30 minutes
 - Hudson Bay Co. (Canada, 2005) problems with inventory system: \$33M loss
 - HP (US, 2004) problems with enterprise resource planning system: \$160M loss

Good SE Practices

- Successful projects
- Happy and repeated customers
- Business value

Course overview

- Course outline
- Evaluation criteria
 - 40% Team assignments
 - 30% Midterms (written, not programming)
 - 30% Individual project and presentation
- Late penalty
- Missed exam

Team Assignments

- To provide team work experience on a programming project
- To apply team work concepts
- To try out different roles in a team
- To assess individual's ability to work in a team
- To assess team presentation skills

Individual Project

- Extension of team assignments
- To assess individual's ability to:
 - Solve problems and work independently
 - Refine requirements and design
 - Program software features
 - Deliver faultless software
 - Present a project professionally

SE is both Quantitative + Qualitative

- Quantitative aspects
 - There is a right answer
- Example
 - What is the time complexity of this algorithm?
 - Which data structure is more efficient in this context?
 - What does this code output?
 - How many tests did the software pass?
 - What is the productivity level of this team?

SE is both Quantitative + Qualitative

- Qualitative aspects
 - There is no single right answer
 - Answers are generally: bad, okay, good
- Example
 - How good is this design diagram?
 - How appropriate is this elicitation question?
 - How would you deal with a team conflict?

Clarifying Misconceptions

- Proficiency in programming \neq excellent software engineer
- Professional software engineer requires:
 - Good communication skills (“soft” skills)
 - Knowledge of tools and processes
 - Ability to work well in teams
- Fun parts (problem solving, design) should NOT be done while programming!

Resources

- Textbook
- Instructor & TA
- Connect

Lab 1 & A1

- Labs start next week
- Assignment 1 Due: Jan 17th Thurs. 11:59pm
 - Group poster on an area of CS
 - Computational linguistics
 - Vision and image processing
 - Computational music and art
 - Machine learning
 - HCI (hardware)
 - HCI (software)
 - Graphics and animation
 - Networks