COIS 2240

Software Design & Modelling

Lecture 1

Introduction to Software Engineering and Version Control System

Taher Ghaleb

TRENT UNIVERSITY

Parts of the content on the slides were sourced from / inspired by the textbook (**Chapter 1**) slides

Course Instruction

Instructor: Dr. Taher Ghaleb

Email: taherghaleb@trentu.ca

Phone Number: 705-748-1011 x6244

Office: OC 120

Office Hours: Tuesday 6:00PM - 7:00PM **&** Thursday 2:00PM - 3:00PM

Lab Demonstrator: Jamie Mitchell

Email: jamiemitchell@trentu.ca

Phone Number: 705-748-1011 x7559

Office: SC 113

Office Hours: Check Blackboard

GTA: Robi (Md Robiuddin)

Email: robirobiuddin@trentu.ca

rrobiuddin@trentu.ca

Office: OC 102.11

Office Hours: Thursday 1:00PM - 2:00PM

About the instructor

Ph.D.: Queen's University (Sep 2017 - Sep 2021)

Postdoc: University of Ottawa (Oct 2021 - Sep 2023)

Research Scientist: University of Toronto (Oct 2023 - Apr 2024)

Assistant Professor: Trent University (Jul 2024 - Present)

Research Interests:

- Software Engineering
- Continuous Integration
- Data-Driven Software Analytics
- AI4SE

What is this course about?

We will learn about:

- Software modelling using Unified Modelling Language (UML)
- Object-Oriented Programming (OOP) in Java
- Software design patterns
- Software design principles
- Software Testing
- Git/GitHub

Course Schedule (Room: ENW 114)

Week	Lecture Topic	Lab
1	Introduction to Software Engineering and VCS	No Lab
2	Basics of Java Programming	Lab 1
3	Concepts of Object Orientation in Java (I)	Lab 2
4	Concepts of Object Orientation in Java (II)	Lab 3
5	UML Modelling (I)	Lab 4 / Assignment#1 Due
6	Software Design Patterns	No Lab
Reading Week		
7	Midterm	Lab 5
8	UML Modelling (II)	Lab 6
9	Software Testing	Lab 7 / Assignment#2 Due
10	Software Design Principles	Lab 8
11	Selected Topic (I)	Lab 9
12	Selected Topic (II) & Course Review	Lab 10 / Assignment#3 Due

Grading Scheme

Type	Weighting	Due Date
Assignment 1	5%	Feb 9th
Assignment 2	7%	Mar 16th
Assignment 3	8%	Apr 4th
Lab (attendance, completeness, and correctness)	10%	End of each week
Quizzes (3-4 in-class quizzes)	10%	TBD
Midterm Examination (in class)	25%	Feb 25th
Final Examination	35%	TBD

Assignments

- Instructions for each assignment will be detailed on the assignment document
- To be done individually
- You will need to use Git for some assignments
- Will prepare you for exams
- Originality and academic integrity is important

Textbook for 2240

Main: *Practical Software Development Using UML and Java, 2nd Edition, Timothy Lethbridge and Robert Laganiere, McGraw Hill.*

Recommended: *Java: The Complete Reference, 13th Edition, Herbert Schildt and Danny Coward, McGraw Hill.*

What is Software Engineering?

The process of solving customers' problems by the systematic development and evolution of large, high-quality software systems within cost, time and other constraints

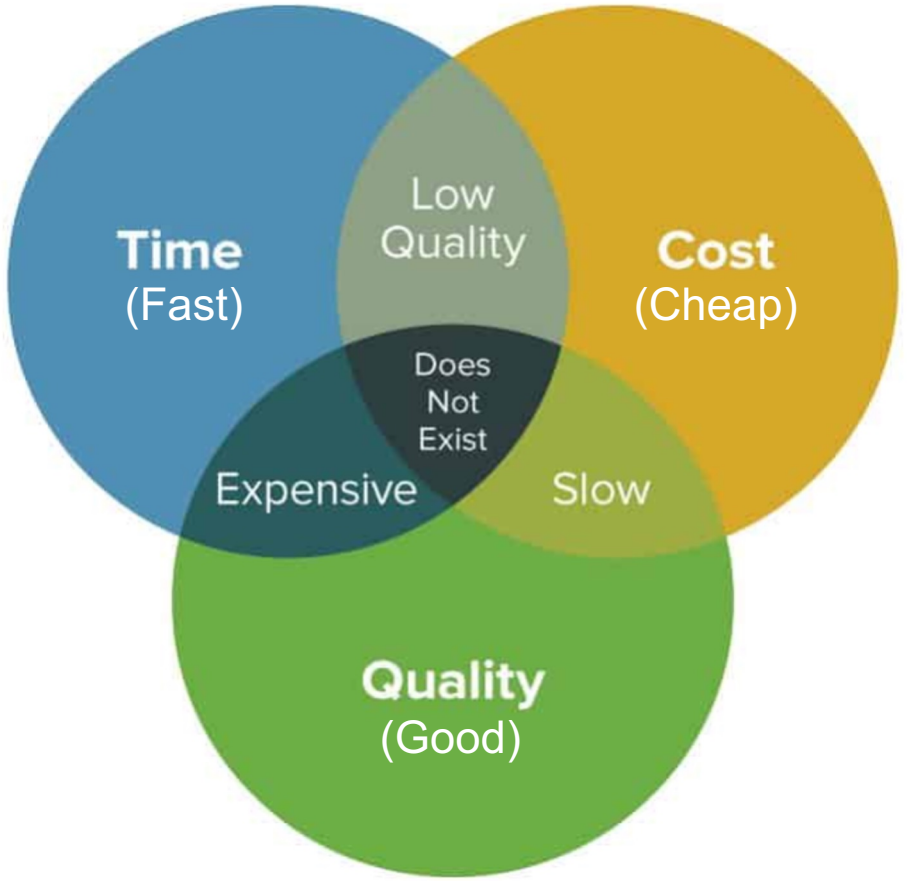
- The goal is to solve customers' problems
- Sometimes the solution is to buy, not to build
- Software engineers must communicate effectively to identify and understand the problem

Systematic development and evolution

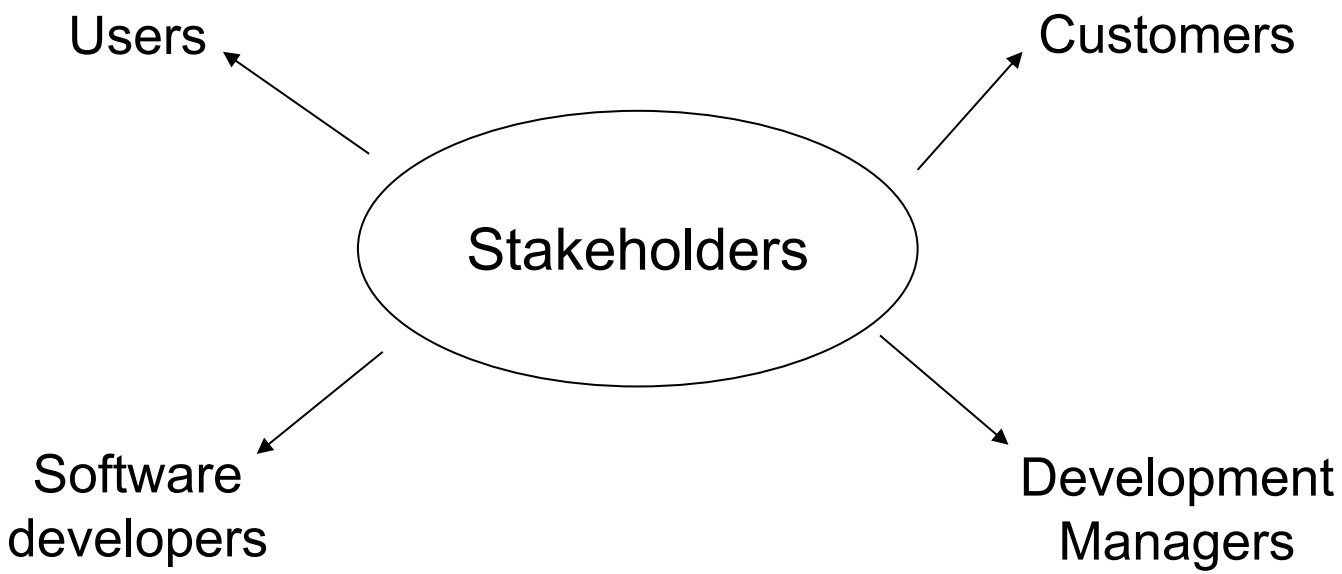
- An engineering process involves applying well understood techniques in a organized and disciplined way
- Many well-accepted practices have been formally standardized
  - e.g. by the IEEE or ISO
- Most development work is evolution
- Teamwork and co-ordination are required
  - *Key challenge:* Dividing up the work and ensuring that the parts of the system work properly together
- The end-product must be of sufficient quality

Cost, time and other constraints

- Finite resources
- The benefit must outweigh the cost
- Others are competing to do the job cheaper and faster
- Inaccurate estimates of cost and time ca cause many project failures



Stakeholders in Software Engineering



All four roles can be fulfilled by the same person

Software Quality

Usability

- Users can learn it and fast and get their job done easily

Efficiency

- It does not waste resources, such as CPU time and memory

Reliability

- It does what it is required to do without failing

Maintainability

- It can be easily modified

Reusability

- Its parts can be used in other projects

Software Quality: Conflicts and Objectives

The different qualities can conflict

- Increasing efficiency can reduce maintainability or reusability
- Increasing usability can reduce efficiency

Setting objectives for quality is a key engineering activity

- You design to meet the objectives
- Avoids ‘over-engineering’ which wastes money

Optimizing is also sometimes necessary

- E.g. obtain the highest possible reliability using a fixed budget

Software Engineering Projects

Most projects are evolutionary or maintenance projects, involving work on *legacy* systems

- Corrective projects: fixing defects
- Adaptive projects: changing the system in response to changes in:
  - Operating system
  - Database
  - Rules and regulations
- Enhancement projects: adding new features for users
- Reengineering or perfective projects: changing the system internally so it is more maintainable

Activities Common to Software Projects

STEP 1
Requirement Analysis
Domain and problem analysis, requirements gathering, analysis, and specification

STEP 2
Design
Design decisions and representations (use case, structural, and dynamic/behavioral)

STEP 3
Development
Implementation: Programming, forms, and reports

STEP 4
Testing
Quality assurance Reviews, inspections

STEP 5
Deployment
Installation and user training

Managing the process (maintenance)

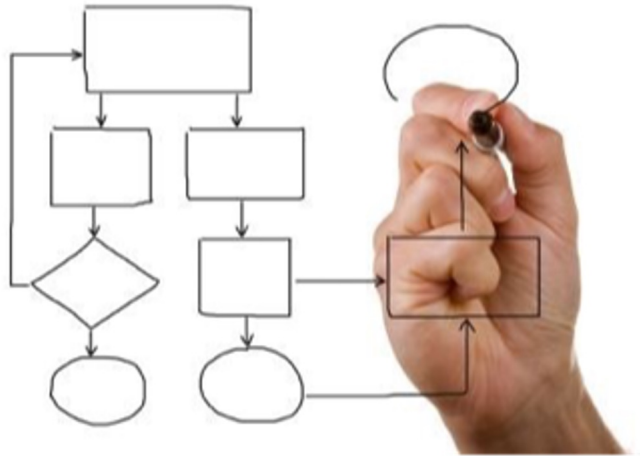
Software Design

- **Systems Engineering:** Hardware vs. software decisions
- **Architecture:** Define subsystems and interactions
- **Detailed Design:** Design classes, algorithms, etc.
- **UI Design:** Define user interaction and system look
- **Data Storage:** Database and file storage decisions

Software Modelling

A model is an abstraction of reality for a given purpose (cannot represent all aspects of reality)

- **Use Case Modeling:** Represents user actions
- **Structural Modeling:** Represents classes and objects
- **Dynamic/Behavioral Modeling:** Represents system states, activities, and interactions



- **Visual Modeling:** Use diagrams and semi-formal languages (e.g., UML)

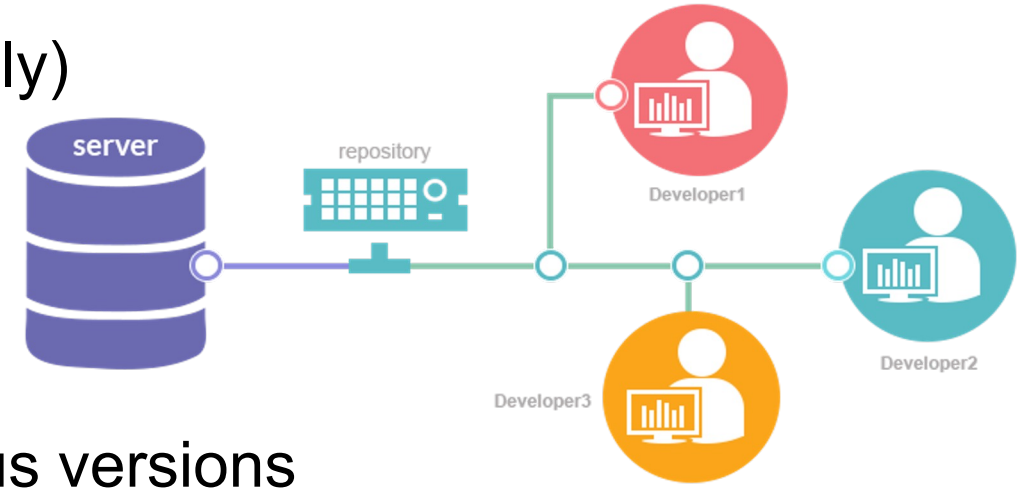
Collaborative Development

- In modern software development, you need to work with a group of other developers.
- Group members change; some leave, some get promoted, some change teams.
- How to track everyone's work?
  - Use VCS.

Version Control Systems (VCS)

A system (e.g., Git or SVN) that tracks changes to files over time

- Manage code changes and maintain version history
- Collaborate with others (internally or externally)



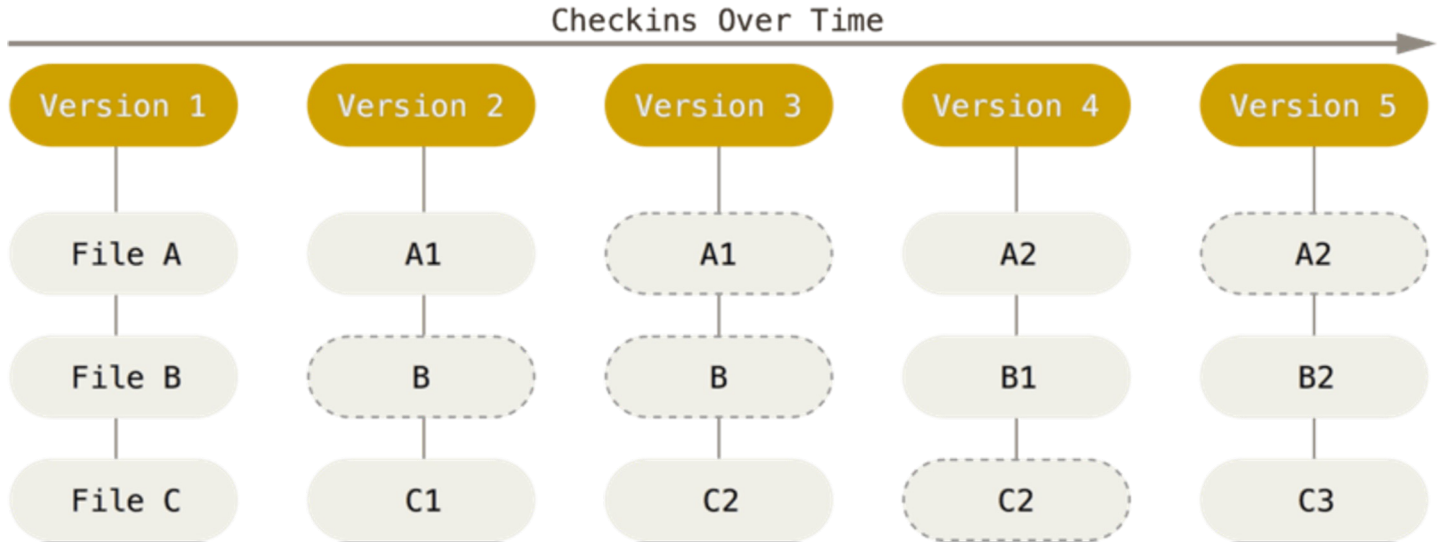
Key Features:

- Tracking: what, when, and who
- Reverting: Undo changes or revert to previous versions
- Branching: Create separate paths for new features or bug fixes

Git

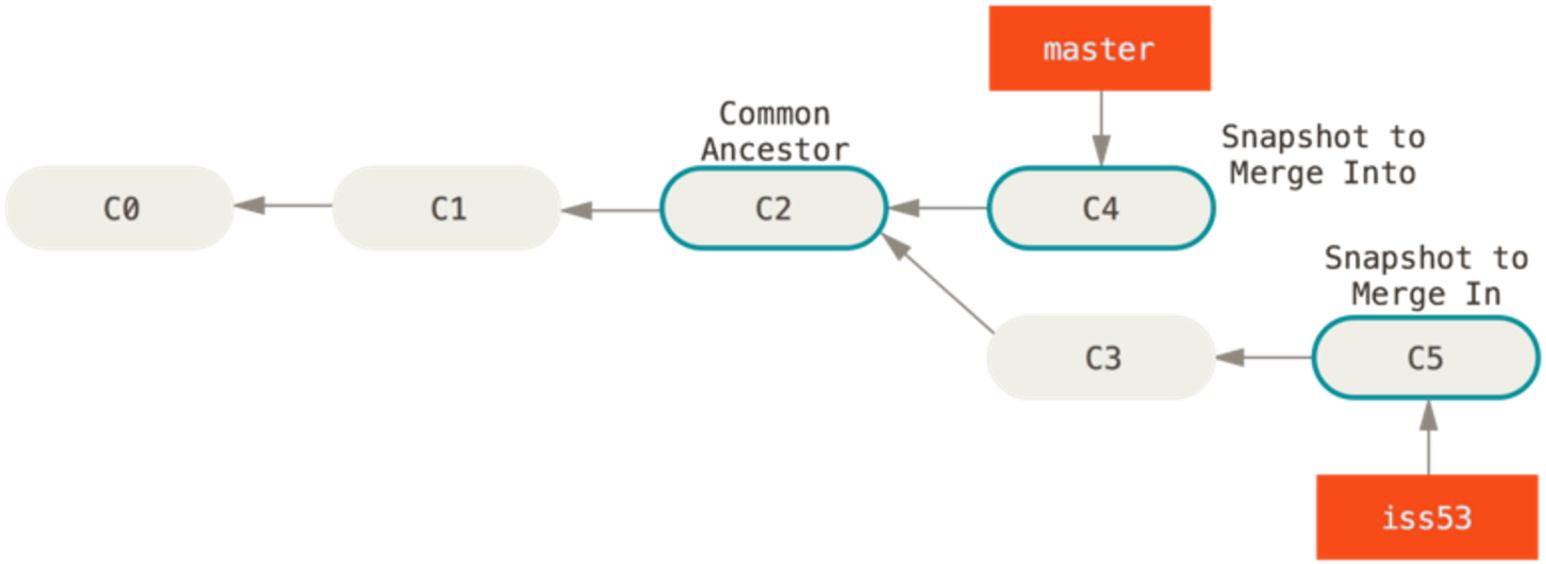
A ***distributed*** VCS, where each developer has a complete copy of the repository

- Git keeps snapshots of files over time
- Each snapshot is a picture of your entire project
- Every change is stored in your local machine
- Git adds data to the repository using snapshots (commits)



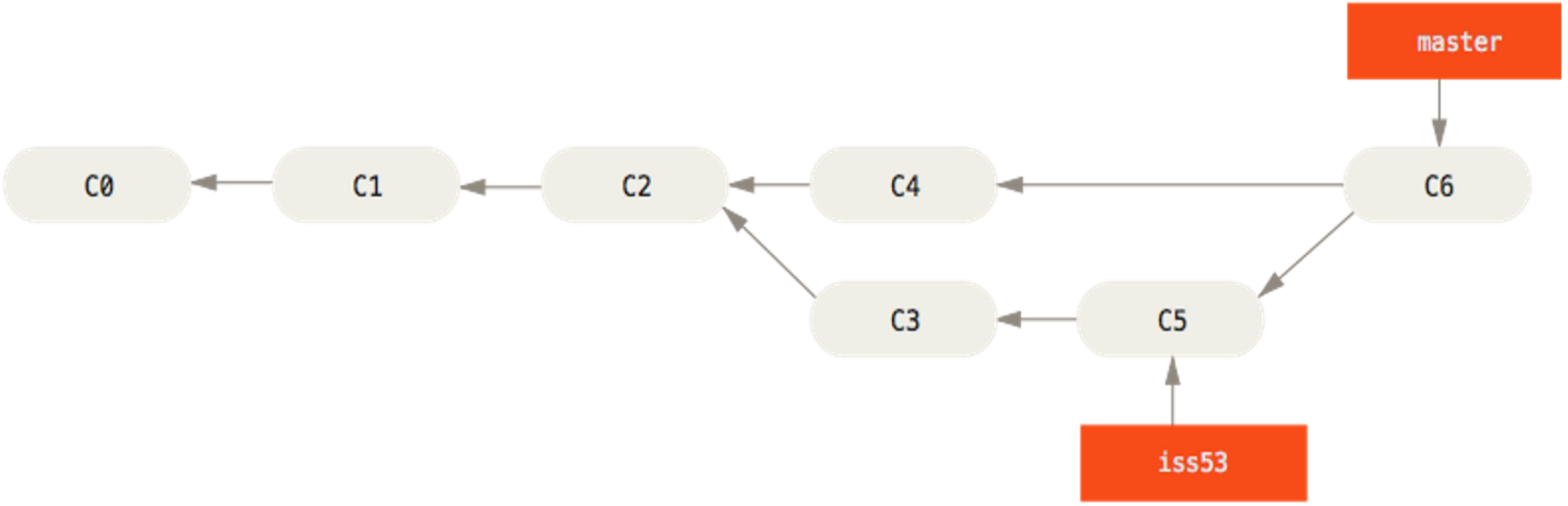
Branching in Git

- In Git, project development is seen as a list of commits
- When development track splits, a branch is created
- In Git, branches are pointers to commits
- The head pointer is the pointer to your current branch



Merging in Git

- When work is done on a certain branch, it can be merged back to the master branch
- During merging, you may have to resolve any arising conflicts
- You can learn more about git from https://git-scm.com



GitHub

A *web-based* platform for hosting Git repositories

- Centralized repository hosting.
- Collaboration: Work with others through pull requests and code reviews
- Issue Tracking: Manage bugs and feature requests
- CI/CD Integration: Automate testing and deployment
- Access control and project management

https://github.com

Summary

- Software Engineering involves requirements analysis, ***design/modelling***, implementation, testing, and deployment
  - We will focus on software design and modelling, but we will touch on some important concepts of software engineering
- Software quality is an important aspect for different stakeholders
- VCS (e.g., Git) is useful for evolving and group projects
- Utilize Labs and Assignments to apply concepts in practice