COMP 4050 – Project Management

Course Description

Calendar entry

Introduction to the issues involved in managing large, complex software projects. Prerequisite: COMP 3350 and (COMP 3040 or a course that fulfills the Written English requirement).

General Course Description

You have *plenty* of coding experience. You've built complex software by yourself, you've learned about the theory behind Computer Science, you've learned about how machines work from first principles, you've even made a complex piece of software as part of a *team*.

While all these experiences have been formative, you've been guided through the software development process by a course instructor or a team leader in a workplace. In this course we're going build on your experiences and take an in-depth look at the responsibilities inherent in guiding a project from start to completion.

Detailed Prerequisites

Before entering this course, a student should be able to:

- Identify issues related to working with a team of people on a software project.
- Create basic software planning artifacts (vision, user stories, estimates).
- Apply a specific agile process to the develop a software project with a team of people.

Course Goals

By the end of this course students will:

- Have a better overall idea of the roles and responsibilities required to build large software projects with a team.
- Assess the initiation and planning phases of a software project.
- Evaluate the management of a software development team.
- Discuss strategies for interacting with humans involved in a software project.
- Develop a release plan and project hand-off.

Learning Outcomes

Initiating software projects

Students should be able to:

1. Assemble and report properties and characteristics of a specific software development project.
2. Identify a variety of costs, values, and risks of a software development project for different stakeholders.
3. Differentiate a "project" from something that is not a project.
4. Compare and contrast different software development methodologies and processes (e.g., waterfall, agile).
5. Justify the application of a specific software development methodology to a particular project.

Planning software projects

Students should be able to:

1. Evaluate software planning artifacts for specific software projects.
2. Identify risks in software development projects.
3. Use specific techniques for software development planning and estimation and defend their use.
4. Apply planning techniques specific to large complex software projects.
5. Critique an existing project plan.

Building a team

Students should be able to:

1. Justify the need to hire a new employee based on the cost of hiring compared to the value a new employee could add to a software project.
2. Evaluate the quality of a job advertisement for a software development position.
3. Compare and contrast approaches for assessing candidates in interviews for software development positions.
4. Assess onboarding plans for a new software developer.

Guiding a team

Students should be able to:

1. Evaluate the practices of high-functioning software development teams.
2. Evaluate communication strategies between software developers on a team.
3. Evaluate strategies for measuring progress.
4. Justify changes to plans based on progress measurements, newfound information, or changes to resources.

Interacting with humans

Students should be able to:

1. Identify the need for having a difficult conversation with members of your team or with your client.
2. Discuss strategies for dealing with interpersonal issues between team members.
3. Analyze strategies for managing requests from clients.
4. Evaluate the outcome of a difficult conversation with members of your team or with your client.

Release planning and hand-off

Students should be able to:

1. Prepare a release plan for software product delivery.
2. Assemble and report information required to hand-off a software development project.
3. Evaluate the outcomes of a software development project and create plans for improving future software development projects.