

CS3733 Classes 1-3

Class 1: Introduction

Course overview

Intro to the project

The project for this course consists of your team producing a software system that satisfies your stakeholders. The system has already been conceptually divided into components (subsystems). Your job is to manage the construction of the subsystem that is assigned to you. You will be required to interact with the stakeholders to prioritize requirements, manage the scope, and ensure that you are building the right things.

Teams

There will be a certain number of large teams working on the project. Each of these large teams will form a group of smaller teams, each of which will be responsible for producing one or two of the subsystems.

The large team will be responsible for the overall project coordination and organization. Each small team will be responsible for building their individual subsystems and helping with the final integration.

There will be a forum on myWPI for you to use to seek out teams and team members and report the makeup of your teams.

You are encouraged to create a name for your team. This is your "company" name.

- Choose wisely

One of the surest ways to have difficulty with this class is to end up on an ineffective, or dysfunctional, team.

Choose your teammates wisely. Find those who have the same work ethic and set of values that you do. Then keep each other involved and responsible to the rest of the team.

Firing a member

You can fire a team member who is not pulling his or her weight. All team members must agree that the firing is warranted and be willing to document the reasons for the dismissal.

Team members who are fired will be required to find another team who is willing to take them on.

What is Software Engineering?

This course is part of your computer science curriculum. However, software engineering is not computer science. What then is it?

Our definition

Software engineering is a combination of many different fields. The main contributors to software engineering are described in the following subsections.

Computer science

Software engineering is intimately coupled with computer science. You cannot be a successful software engineer unless you have some in-depth knowledge of algorithms, programming methods, logic, discrete mathematics, and other basic computer science topics.

Engineering

Engineering is:

The profession in which a knowledge of the mathematical and natural sciences gained by study, experience, and practice is applied with judgement to develop ways to utilize, economically, the materials and forces of nature for the benefit of mankind.

(Accreditation Board for Engineering and Technology, 1996)

Engineering is different from a basic science in that it is concerned with the application of science to solve problems, rather than discovering new scientific principles.

Business

Software engineering is done in a business, or enterprise context. We engineer solutions to satisfy the needs of the organization.

Software engineering is applied in many commercial settings. It is much more than providing Information Technology (IT) systems to an organization. Software has become an ubiquitous entity in every aspect of life today; whether it is manifested in the form of embedded software in appliances and communication devices, large data processing and data mining applications, scientific and engineering systems, and so on.

As the textbook observes, the software system is less than the enterprise's information system. Software engineering takes into consideration the whole enterprise.

Key concepts

There are two key concepts that you need to understand and consider early and often throughout a project.

Change

One constant in software development is that *things change*. Change is inevitable. An effective process allows you to address changes in requirements, business conditions, new technologies, ...

Risk

Successful projects are those that constantly pay attention to potential risks and work at reducing them. One of the things you will need to do is maintain a risk list and keep it up-to-date.

You should prioritize your risks on a project according to the likelihood that they will happen and the possible damage that can happen if the risk materializes.

Iterative development

We will be discussing lifecycles later in the course. For now, it's enough to know that most modern software processes are based upon an iterative development model. For this class, iterations will be one week long, ending on Thursday. This means that every Friday, beginning with the second Friday of the course, your team must have working software that has more functionality in it than the previous week. We will talk about the

Class 2: Tools

Eclipse demo

SourceForge

We will use SourceForge for the course projects. In order to do this, you need to take the actions described in the next sections. The URL for the WPI SourceForge installation is at:

<http://sourceforge.wpi.edu>.

Create an account

Go to the SourceForge home page and do the following:

- click on Create an Account
- Fill out the account creation screen.
 - You MUST use your WPI user name (e.g., gpollice)
 - Use your WPI e-mail address
- Submit the form. Once it is approved, you will receive an e-mail message that will give you a link to a page for you to activate your account. You must do this in order for your account to become active.

Create a team project

Once you have selected your team, you should have one person request a project. Log into SourceForge and do the following:

- Go to My Workspace and then select the Projects tab.
- Click Create Project
- Name your Project "CS3733-D05 <team name>."
- Select "Project Template," not "Public Project Template."
- Submit the form and wait for an e-mail notification indicating that your project has been created.

Once you have a project, get the user names of all members of your team and add them to the project.

Create a repository in the project

Ask the instructor for a repository number. Once you have the project created, go to the project page and click on the Source Code tab. Then create a repository by doing the following:

- Click Create Repository.
- In the request form, make the directory cs3733d04-n, where 'n' is the number given to you by the instructor.
- Submit the form and wait for e-mail indicating the repository is set up.

Repositories are not just for source code. All artifacts for your project should be under version control.

Monitor the discussions, etc.

Each member of the team should monitor all discussions and artifacts they care about in the project. See the SourceForge on-line help for information on how to do this and set up mailing lists.

CVS

CVS is a version control system that is bundled with SourceForge. That is the repository you created. You should keep all of your source code and documents in version control.

Version control systems like CVS are not a luxury, they are a necessity for software development. For any significant software project, this is a critical action to take. You are always faced with the possibility of someone deleting a file or changing a file so bad that you simply need to back up to a previous version.

You will be working with CVS through your Eclipse IDE. This will be shown in the Eclipse demo. You should read the documentation in the SourceForge on-line help about working with CVS.

We strongly recommend that your team establish guidelines for file naming, and commenting checkins and checkouts in CVS.

Homework

There is an assignment on the Course Web Site schedule page. It is due next Monday.

Class 3: Project teams

What the heck are you talking about?

Any time you don't understand something, it is **your responsibility** to speak up. It's perfectly reasonable at any time during a class (except when taking tests, etc.) to ask "**What the heck are you talking about?**" At that point the speaker has the responsibility to try and clarify the issues for you.