

CIST 2931 Fall 2016

Service Learning Project Description

Introduction

CIST 2931 Advanced Systems Project is the capstone course for the CIST Computer Programming Diploma and Associates of Applied Science Degree. The course is designed to provide the student with a “real world” experience in designing and building a software system using all the skills learned in the program. In other words, this course is an opportunity to “pull it all together”.

To achieve this goal the bulk of this course is a team project. Teams are used for two reasons. First, almost all software development projects of any real size are team efforts in the industry today. Learning to work with a team of people that you may or may not get along with personally is a skill you will need. Secondly, not every student will have a full range of skills at the same strength. By having at least three people on the team allows you to assign work to the team member most capable. That being said it is expected that all team members will contribute in a meaningful way.

The Project

This term we are fortunate to have a real customer. This will be a service learning project we will do for The DeRiche Agency (called "the Customer" from this point forward). You will be automating a manual process for them. You will do this using an Agile Software Development methodology (Scrum). A representative from our client agency will be present to participate in the software development process in the role of “customer” or “user”. You will be building real software that is intended to be used in the real world.

Our Customer is a Center for Developmentally Disabled adults. They provide services for their clients and must document the time spent with their clients using a paper document called a Progress Note. This document must meet some detailed standards or our customer may lose funding associated with that client. A more detailed problem description has been provided by our customer and a representative from that firm will be visiting the class to participate in the development of the software project. In short, the software the customer wants will automate the creation of these progress notes in order to minimize errors.

In most programming assignments in the real world, a programmer is given code written by someone else and asked to change it (fix bugs and/or add new features). You will be doing the same thing here. Last semester a team started on version of the DeRiche project originally started by another team one year ago. The software is still not complete. That project as your starting point this semester.

Your team’s job will be to come up to speed on what the software you have been given. This means getting it working in the IDE in which it was developed. And understanding how it is structured, so you can begin working on adding the new features the customer wants.

Things to Think About

The point of this course is to pull it all together and complete all the steps of the Software Development Life Cycle. To accomplish this each team will use an iterative, agile process where unit testing is an important part. As a result your team will have to answer many questions (some may have already been answered by the previous team):

- How to figure out what the software you have been given works, and if you are not familiar with the language, how to come up to speed in a reasonable time frame?
- How will you figure out what the client's needs are?
- How will you create repeatable tests of the software you add to this project?
- How will the software be deployed on the user's machines?
- How will the user back up their data?
- How would the user recover a crashed machine?
- How secure does the data/application need to be?
- It is uncertain at this point if your team will be able to develop all the additional features that the customer may want in only 15 weeks. How will you document your project so another team can pick it up and continue development if needed?
- How will you provide an appropriate level of "Help" to the user?

This is not by any means a complete list of questions. You no doubt will come up with more. In other courses you were focused primarily on only part of the SDLC. In this course you'll have to actually do all of it. So try imagining the user, happily using your application. What do you have to do to get there? How will you control the details? For example do you really need a full blown installer with all the bells and whistles or can something simpler but more labor intensive be used in the version you deliver? I keep mentioning installation because it is not something normally covered in our programming classes and you need to think about it. Does the software you decide to start with install properly on the customers machine at this point? Or is that one of the fixed they want?

You can get answers to questions in two ways. For technology that was not covered in our curriculum so far, e.g. software to install your application, lectures will be given to introduce you to the concepts. You will be encouraged to research the details of new concepts yourselves. Technology is constantly changing so this is what you would do on a project in the real world after all. Lectures will also be given to briefly review concepts covered in other courses. But you are expected to know the content of courses take previously.

In terms of questions involving customer needs, those will be answered by our customer, not you or me. It is important to realize, that a nice neat new bell or whistle you think up, will take time to implement, and may not be what the customer needs. You do not want to waste time working all weekend on this cool idea that the customer later tells you they don't want. ALWAYS get approval for feature ideas that originate in the team, rather than the customer. Speaking of our customer, please be aware of the concept of "feature creep". It should be perfectly OK for the customer to change his or her mind about a feature, or add new features not originally thought of. That is what agile processes encourage. The

important thing is to control how and when the features are added or changed. And be sure the customer understands the consequences in terms of delivery at the end of the term.

How the Project Will Be Graded

This may all seem overwhelming. It does not have to be. **The important thing for your grade is quality not quantity.** The project will be graded using a rubric available on our Blackboard web site. Your team's application does not have to have all features running. The quality (quality means "it is professional") of the deliverables from the SDLC will determine the team's grade, not the completeness of the application (though some minimal set of new features is expected). Please see the rubric for details.

Individual project grades will be computed by using the team project grade and the results from a peer review that will be done at the end of the term. The peer review allows you to rank your fellow team-mates (but not yourself) and this ranking is used to weight the team grade up or down to determine each team members individual grade.

Your team will also be graded on a Project Proposal and weekly reports. But a peer review for these is not used. Rubrics for the proposal and weekly report are also on the Blackboard web site.