

Software Configuration Management Project, Spring 2015

Create an application which allows a professor to create an exam from a set of questions in XML format.

Project Management

You will need to submit the following forms of documentation:

- Risk Mitigation Plan (Create a new version every 4 weeks with updated risks, probabilities, and impact values.)
- Schedule (Create your schedule using Microsoft Project. Update the schedule weekly with percentages complete.)
- Software Estimates (Estimate LOC count at the beginning of the project and describe your methodology.)

Requirements

You need to come up with the requirements for the program but they must follow these guidelines:

- There will be two separate input XML files.
 - One for the set of questions.
 - One to guide the application as to how many questions of particular types and categories should be selected for the resulting output file.
- Each question is an element in the XML file.
- Each question has the following attributes:
 - Type = Short Answer/ Matching / True-False / Multiple Choice
 - Question
 - Answer
 - Category
- The resulting output file should be a plain text file that lists the questions grouped together by type.

Design

You must create the following diagrams for your software:

Class Diagrams

Sequence Diagrams

Each one of the diagrams must be complete and correct; however, you don't need to put everything on one diagram. You may create these diagrams using any modeling tool and place them into the design document. Other diagrams can be added if necessary.

Coding and Unit Test

You may code your program in any one of the following languages:

C++, C#, Java

The program does not need to be a GUI. You must also use revision control software to keep track of the changes to your source code.

After each major section of your code is done, you must perform a unit test. After the unit test is complete, you must hold a code and unit test review for that section of code. Instructions for the code and unit test review will be provided.

Code and Unit Test Reviews

You will need to perform at least three code and unit test reviews. For each code review, copy the code into a Microsoft Word document and turn "Track Changes" on in the Review tab so any changes made to the code in the Word document will be captured. Also, use the comment features in Word to provide to and respond to comments during the peer reviews. For each review, you must also fill out a unit test report before the review to describe what steps you have taken to test the code being reviewed. At the end of each code review, you must fill out an inspection report and input any defects found into ClearQuest for tracking. You must update the ClearQuest records as you complete the fixes for all the defects discovered.

System Testing

You must prepare a software test document and document the results during system testing in order to verify the system meets the requirements. You must also submit sample XML files that you have used to test the system.

Final Report and Presentation

You will need to prepare a 20-page minimum final report that details what you have worked on over the semester. You will also need to deliver a minimum 30-minute presentation that covers the material from your report.

Some questions to think about while writing the report:

What tools do you use to keep track of scheduling and project milestones?

How did you identify the risks and how did you develop plans to mitigate those risks?

What actions did you take to elicit requirements from the customer?

What tools or methodologies did you use to design your software?

How did you capture defects then track and resolve them by using Clear Quest?

What kind of revision control software did you use? How did you use it?

How did you gather status from the various team members?
How did you start to code and unit test the project?
How did you conduct requirements reviews, design reviews, and code reviews?
What tools and methodologies did you use to code your project?
How did you conduct code and unit test reviews?
Did you create a version of your software after every successful code and unit test review?
How did you unit test your software?
How did you come up with the formal test procedures to put in the software test plan?
How did you construct your test cases for the software testing?
How did you make sure every requirement was being tested in the unit test and the software test plan?
Where there any unresolved issues after formally testing the software? If so, how did you capture these?
What are the ethical and social impacts of your application?

Documentation

You will turn in several pieces of documentation in one zip file. Only one team member needs to submit the documentation for the entire team project. Templates will be provided for some of the documentation to submit. The documentation to be included is:

1. Project Management Documentation
2. Requirements Documentation
3. Design Documentation
4. Source Code
5. Code and Unit Test Review Documentation
6. Test Documentation with Results and Input Data
7. Final Report and Presentation
8. Revision Control Software History Log

Assessments

Each team member will perform a self-assessment of how they did on the project and a peer-assessment of how each of their team members performed. Items to include in the assessments are:

- Responsibilities
- Strengths
- Weaknesses
- Recommended Score (0-100)

Each person will turn in their assessments separately from the documentation listed previously. Give specific examples and write at least 2 paragraphs for each item (responsibilities, strengths, weaknesses).