**CS 460 Exam 1 (in-class)**

**Name\_\_\_\_\_\_\_YeiSol Woo\_\_\_\_\_\_\_Date\_\_4/6/2012\_\_\_\_**

**Total: 100 pts**

**Part I: Software Engineering [Subtotal 80 pts]**

**(1) SWEBOK (Only Chapter 1: Intro to the Guide) [35 pts]**

**(2) PMBOK (Only Chapter 1: Intro to the Guide) [18 pts]**

**(3) RUP (Only pages 1-15) [12 pts]**

**(4) Team Roles/Tools/Standards [15 pts]**

**Part II: Essay [Subtotal 20 pts]**

**Provide an overview of your Project contribution:**

**(1)  Completed up to Spring Break (1/2 page + diagram) [10 pts]**

**(2)  Planned after Spring Break (1/2 page + diagram) [10 pts]**

**(3) Bonus [10 pts]**

**Part I: Software Engineering (SWEBOK, PMBOK, RUP, and Team Roles /Tool)**

**(1a) [T/F] [10 pts] SWEBOK Essentials**

|  |  |  |
| --- | --- | --- |
| **#** | **T/F** | **Statement**  |
| 1. | **F** | There is no difference between software engineering and programming code. |
| 2. | **T** | SWEBOK promotes a consistent view of software engineering worldwide and characterize the contents of the software engineering discipline. |
| 3. | **T** | At least 500 reviewers from 42 countries contributed to SWEBOK. |
| 4 | **T** | www.swebok.org is where more information can be found on SWEBOK. |
| 5. | **T** | Computer science is another discipline related to Software Engineering. |
| 6. | **F** | SWEBOK knowledge areas are hierarchically decomposed but not defined. |
| 7. | **T** | Each SWEBOK KA description also includes a matrix relating the reference material to the listed topics. |
| 8. | **T** | A requirement is defined as a property that must be exhibited in order to solve some real-world problem. |
| 9. | **T** | According to the IEEE definition [IEEE610.12-90], design is both “the process of defining the architecture, components, interfaces, and other characteristics of a system or component” and “the result of [that] process.” |
| 10. | **T** | The 2004 Guide is simply the current edition of a guide which will continue evolving to meet the needs of the software engineering community. |

**(1b) [3 pts] Define Software Engineering according to SWEBOK and describe how it relates to coding.**

The first thing to note is that many people will group software engineers and software developers in the same category. And while many of their skills and jobs will overlap, there is definitely a difference between the two jobs, and this difference can help explain the definition of a software engineer.

A software developer will write the code to create a software program.

A software engineer will focus on creating systems that may use this software program, but may also uses a range of other programs that need to integrate smoothly.

So the work of the software engineer is far broader in scale.

And the reason the term engineering was used initially was to try and differentiate between these two jobs, and also to explain the systematic processes a software engineer will use, which are similar in nature to the systematic way that a civil, electrical or mechanical engineer will go about their work.

 **(1c) [22 pts] SWEBOK Knowledge Areas**

|  |  |
| --- | --- |
| **Identify official Knowledge Area** | **Brief Definition** |
| 1. Software requirements | Software requirements Knowledge Area is concerned with the elicitation, analysis, specification, and validation of software requirements. Software requirements express the needs and constraints placed on a software product that contribute to the solution of some real-world problem. |
| 2. SoftwareDesign | Design is both ‘the process of defining the architecture, components, interfaces, and other characteristics of a system or component’ and ‘the result of the aforementioned process.’ Software Design plays an important role in developing software: it allows software engineers to produce various models that form a kind of the solution to be implemented. |
| 3. SoftwareConstruction | Software Construction is the detailed creation of working, meaningful software through a combination of coding, verification, unit testing, integration testing, and debugging |
| 4. Software testing | Testing is an activity performed for evaluating product quality, and for improving it, by identifying defects and problems. Software testing consists of the dynamic verification of the behavior of a program on a finite set of test cases, suitably selected from the usually initiate executions domain, against the expected behavior. |
| 5. Software maintenance | Maintenance is the phase of software lifecycle that starts upon delivery whose activities may start much earlier. In this life cycle phase, the code can be altered to fit changing needs of the client as well as fix any bugs that may arise. It is defined as the totality of activities required to provide cost-effective support to software. |
| 6. Software Configuration Management | The configuration management (CM) is the discipline of identifying the configuration of a system at distinct points in time for the purpose of systematically controlling changes to the configuration, and maintaining the integrity and traceability of the configuration throughout the system life cycle. |
| 7. Software Engineering management | Software Engineering Management can be defined as the application of management activities - planning, coordinating, measuring, monitoring, controlling, and reporting - to ensure that the development and maintenance of software is systematic, disciplined, and qualified. Engineering management addresses the management and measurement of software engineering. |
| 8. Software Engineering process | The engineering process deals with the definition, implementation, assessment, measurement, management, change, and improvement of the software engineering process itself. |
| 9. Software engineering tools and methods | The engineering tools in KA covers both software engineering tools and software engineering methods. Engineering tools are tools that aid in the engineering process in ways such as making collaboration easier, helping manage quality checks, etc. Software engineering methods include areas such as heuristic methods dealing with informal approaches, formal methods dealing with mathematically based approaches, and prototyping methods dealing with software development approaches based on various forms of prototyping. |
| 10. Software Quality | Quality deals with software quality considerations, which transcend the software life cycle processes, and includes areas such as software quality fundamentals, software quality management, and practical considerations. |
| 11. Related Disciplined of Software Engineering | In order to circumscribe software engineering, it is necessary to identify the disciplines with which software engineering shares a common boundary.  |

**(2) [18 pts] PMBOK Knowledge Areas**

|  |  |
| --- | --- |
| **Identify official Knowledge Area** | **Brief Definition** |
| 1. Project Integration Management | Project Integration Management includes the processes required to ensure that the various elements of the project are properly coordinated.  |
| 2. Project Scope Management | Project Scope Management includes the processes required to ensure that the project includes all the work required to complete the project successfully. Major project scope management processes are: Initiation, scope planning, scope definition, scope verification, and scope change control.  |
| 3. Project Time Management | Project Time Management includes the processes require to ensure timely completion of the project. The major processes are activity definition, activity sequencing, activity duration estimating, schedule development, and schedule control. |
| 4. Project Cost Management | Project Cost Management includes the processes required to ensure that the project is completed within the approved budget. Major processes are resource planning, cost estimating, cost budgeting, and cost control.  |
| 5. Project Quality Management | The quality of a product is measured by its ability to satisfy its requirements. A project’s quality is managed through planning, assurance, and control measures to ensure its requirements are met.  |
| 6. Project Human Resource Management | Project Human Resource Management includes the processes required to make the most effective use of the people involved with the project. The major processes are Organizational Planning, Staff Acquisition, and Team Development. |
| 7. Project Communications Management | Project Communications Management includes the processes required to ensure timely and appropriate generation, collection, dissemination, storage, and ultimate disposition of project information. It provides the critical links among people, ideas, and information that are necessary for success. |
| 8. Project Risk Management | Project Risk Management includes the processes concerned with identifying, analyzing, and responding to project risk. It maximizes the results of positive events and minimizes the consequences of adverse events. The major processes are Risk Identification, Risk Quantification, Risk Response Development, and Risk Response Control. |
| 9. Project Procurement Management | Project Procurement Management includes the processes required to acquire goods and services from outside the performing organization. Goods and services, whether one or many, will generally be referred to as a “product.” Procurement planning, Solicitation planning, Solicitation, Source selection, Contact administration, and Contact close out are the major processes. |

**(3) [12 pts] RUP Phases**

|  |  |  |
| --- | --- | --- |
| **Identify Phase** | **Identify at least 1 project deliverable** | **Briefly describe project deliverable** |
| 1. Inception | Vision document, SRS, FMEA (Risk Analysis), initial use cases | Identify project scope; define requirements; estimate risks. |
| 2. Elaboration | Architectures, use cases, PMP, QFD, DM, DSM, SDD | Creation of architecture, decision to implement or cancel project. |
| 3. Construction | Code, executable, test cases, user manuals | Architecture is filled in with code, unit-testing, verification. |
| 4. Transition | Marketing, user-oriented documentation, user feedback, software updates | Software is delivered to end-users and integrated into the production system; validation (beta-testing); training users and maintainers. |

**(4) [15 pts]** **List your team members, state their roles, and match the tool/standard they use.**

|  |  |  |
| --- | --- | --- |
| **Team Member Name** | **At least 1 role**  | **At least 1 tool/standard used** |
| 1. Marat Kurbanov
 | Project Leader, Application Architect, Web Developer
 | Ms Visio, MS project, Eclipse, MS office, basecamp, GIT |
| 2. Trent Forkert
 | Asst. Project Leader, System Architect, System Dev.
 | Ms Visio, MS project, Eclipse, MS office, basecamp, GIT |
| 3. Alek Bouillon
 | Network Architect, Web Developer
 | Ms Visio, MS project, Eclipse, MS office, basecamp, GIT |
| 4. YeiSol Woo
 | Database Manager, Business Logic Dev.
 | Ms Visio, MS project, Eclipse, MS office, basecamp, GIT |
| 5. Connor Becker | Documentation Manager, Business Logic Dev.
 | Ms Visio, MS project, Eclipse, MS office, basecamp, GIT |

**II. Project Essay**

**Provide an overview of your Project contribution:**

**(1)  Completed up to Spring Break (1/2 page + diagram)**

**(2)  Planned after Spring Break (1/2 page + diagram)**

**< next page>**

1. **[10 pts] Completed up to Spring Break (1/2 page + diagram)**



 

 

I Create a Usecase model and Usecase specification for user authentication by using RUP template.

I create a simple webpage to verify the user's credentials. If a user provides right information, LDAP bind is successful. Otherwise, the LDAP bind is unsuccessful then the credentials are considered invalid and no session is created for the user.

An LDAP server is a server that organizes users into groups and stores details for each user including their real name, their employee or student ID number, their email address, etc. It also stores the user’s password. For example, if a user wants to access a web page then the web server would prompt the user for their user name and password and then check if they are valid. The web server could also check if the user was a member of a particular group. This way access to the web page could be given to all the users in a particular group (i.e. all CS professors can access a web page showing the grades for students in the CS department). So basically an LDAP server can be used to (1) lookup information about users and (2) validate user name and password.

1. **[10 pts] Planned after Spring Break (1/2 page + diagram)**



Task that I did after spring break

1. Create a bingo table in HTML to display “B.S. CS Bingo Sheet” (Without Tab Design)

2. Updated the previous bingo table by adding tab design to display “B.S. CS Bingo Sheet” efficiently

3. Change the HTML bingo table to JSP style by using the data model

4. Create a general website template for myIPFW Advisor

5. Added a servlet filter to handle LDAP login for security

6. Create bingo sheet servlet to generate the sheet form the data model

7. Change color scheme and style to better match IPFW’s style and create some tabs on top to switch between different areas

8. Added a calendar to display the iCal calendars from the scheduler

9. When advisor login, bingo table could be editable and the admin tab will show up. However, when student login, bingo table could not be editable and the admin tab will not show up.

Task I did recently:



When the user opens the web application a servlet filter checks if the user has logged in already. If they have already logged in, then it redirects them to the home page (index.jsp). If they have not already logged in, then it redirects them to the login page.

The user types in their IPFW username and password and that is validated against LDAP. After the user's DN is found, it does an authenticated bind to the LDAP server to allow the program to get more details from the LDAP server. The LDAP server provides the user's IPFW ID number, their email, and sometimes it even lists their advisor's name. All of this information is then stored in a Java object.

The IPFW ID number is used to check if the user's data is already in Connor's database. If it isn't then it creates a new header entry to store the information extracted from the LDAP server. It also uses the ODS data model to pull a list of all of the student's course history (course taken, grade, semester) along with any transfer credits.

The list of course history is compared against the requirements in Connor's database and a list of satisfying courses (see satisfiers.xml) for each requirement. For each satisfier, an entry is created in Connor's database.

It basically log into the application and all of the data was imported automatically. The same process could be done with an advisor manually typing in the student's IPFW ID number rather than the student logging in.

**Bonus [10 pts total ]**

**B-1 [5 pts] List up to 5 topics you learned this semester from class.**

PMBOK/ SWEBOK

IEEE 12207

RUP

UML design

Real time design using COMET

**B-2 [5 pts] List up to 5 favorite topics from class.**

SWEBOK

PMBOK

UML design

RUP

IEEE 12207

**B-3 [10 pts] Identify and describe your implementation duties in CS 460.**

This is my first time to do a real longtime project and from this class I learned teamwork and lots of things during the project. On the project, my work was an important part in the project. Since I need to gather all the information and display properly. I didn’t design all of the functionality in the project since I started to join in this semester. However, while we have a project meeting on Wednesday, I always discuss lots of function with professor Sedlmyer. And while I’m constructing the design and real application I learned the difference between small project between our group project and I learned a lot from this class. ☺