**Name \_Honore Hodary\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Date \_11/19/2011\_\_\_\_\_**

**(Type answers and include question)**

**CS 360 Midterm I: Software Architecture and OO Detailed Design (Fall 2011)**

1. **[10 pts] We have covered the architectural hierarchical decomposition process from your initial vision document. Describe:**
	1. **Application architecture**

**Application architecture is an architecture that describes the internal structure of an application. It encapsulates the principal characteristics of a class of systems**

* 1. **Information architecture**

**Information architecture is the FRs/DPs view there the system being designed is expressed as a set of functional requirements (outputs) and design parameters (inputs).**

* 1. **Systems architecture**

**Systems architecture is a flow chart that shows the hierarchical structure of the system that emerges from the decomposition of FRs/DPs from the topmost level (system) to the levels of increasing details (components).**

* 1. **UML architecture**

**UML architecture is an architecture that uses uml diagrams to capture a particular facet of the system being modeled. Uml can capture the system using logical, process, physical, development, or use case view.**

1. **[10 pts] In RUP, we are leveraging the axiomatic design process in the inception and elaboration phase. Describe the requirements engineering and elicitation process in more detail from initial gathering of VOC to CN to FR in axiomatic design, including risk concepts.**

**Requirement engineering is the process of identifying, documenting, and communicating the purpose of software by identifying all stakeholders and their needs. On the other hand, requirements Elicitation is an activity of the requirement engineering which consists of discovering the requirements for a system by communicating with all stakeholders.**

* 1. **Design matrix results in which UML diagram type?**

**Class diagram**

* 1. **DSM results in which UML diagram type?**

**Component diagram**

* 1. **What is the V-Model**

**The V-model is a software development method that uses both top-down and bottom-up approaches to develop software. The first step in V-model is to use axiomatic design’s top-down approach to get the CNs, define FRs, map the FRs to DPs, decompose the FRs/DPs further, and then generate the full design matrix table to define modules. The next and final step is to build object oriented model using bottom-up approach to identify classes, establish interfaces, and write program code with system architecture.**

**Steps for top-down:**

**Get CNs**

**Define FRs**

**Map FRs to DP**

**The FRs/DPs further**

**Full design**

**Define modules**

**Bottom-Up**

**Identify classes**

**Establish interfaces**

**Coding with system architecture**

**Software product is ready**

* 1. **QFD**

**Accordign to Dean, E. B., the QFD is a technique used to translate often subjective quality criteria into objective ones that can be quantified and measured.**

**In Acclaro, the QFD is used to map the relationships between quality characteristics that customers desire and substitute quality requirements expressed in engineering terms.**

* 1. **FMEA**

**In engineering, FMEA is a technique used to prevent failures by identifying and countering weak points in a product, process, or design.**

1. **[5 pts] We have covered how to establish class architecture first as a foundation to detailed design with UML. When using Axiomatic Design process to develop classes, how do the following map to OO Design object elements [hint: see paper handout on “Object Oriented Design with Axiomatic Design” :**
	1. **FR**

**Object**

* 1. **DP**

**Data**

* 1. **FR/DP design matrix intersection**

**Methods**

1. **[5 pts] Describe the purpose of these tools/software used in your project:**
	1. **Acclaro DFSS**

**Acclaro DFSS was used as a tool to develop Neighborhood Food Network Management System using axiomatic design methodology. For instance, Acclaro helped me get customer needs (CNs), define and decompose FRs/DP, generate the full design matrix, develop QFD, and analyze and mitigate risks using FMEA.**

* 1. **Visio**

**Visio helped me design application architecture and use UML to capture the interaction between actors and use cases.**

* 1. **Basecamp**

**Basecamp was used to keep in touch with other team members by posting all new documents, updating existing ones, scheduling work among team members, and sharing in other information pertaining to our project.**

* 1. **MS Project**

**MS Project was used to make Gantt chart for the project.**

* 1. **MS Powerpoint**

**MS PowerPoint was used to make PowerPoint presentation that we had to show in class as it was required.**

1. **[5 pts] Describe the standards applied in this project**
	1. **IEEE-830 SRS**

**IEEE-830 SRS is the recommended practice to develop and good Software Requirement Specification (SRS) document. It describes the content and quality of a good SRS. The IEEE-830 was developed by Institute of Electrical and Electronic Engineers**

* 1. **IEEE-1058 PMP**

**IEEE-1058 is a standard that describes the recommended practice (format and content) for writing a good project management plan. It describes all elements that should appear in a project management plan.**

* 1. **IEEE-1016 SDD**

**IEEE-1016 is a standard that describes the recommended practice (the required content and organization) for Software Design Descriptions (SDD). SDD is a representation of a software design to be used for communicating design information to with stakeholders**

1. **[5 pts] In OO design, describe the concept [with symbol]:**
	1. **Aggregation**

**Aggregation represents “is part of” relationship between two classes or components.**

****

* 1. **Composition**

**A strong form of aggregation in which the “whole is completely responsible for its parts and each “part” object is only associated to the one “whole” object.**

****

* 1. **Polymorphism**

**Means different objects can respond to the same message in different ways, enabling objects to interact with one another without knowing their exact type.**

****

* 1. **Inheritance**

**Represent “is a” and “is like” relationships.**

****

* 1. **Blackbox**

**Black-box is a type of views of components in UML that shows how the component looks like outside (input and output), but it does not say anything about the internal structure (how input is transformed to output) of the component.**

****

1. **[5 pts] In OO design, describe the concept [with symbol]:**
	1. **Public operations**

**Public operation is represented by a plus sign (+) to the left of an attribute or operation name as shown below.**

|  |
| --- |
| Class-A |
| + Attribute1+ Attribute2 |
| +Operation1( )+Operation2( ) |

* 1. **Private operations**

**Private operation is represented by a minus sign (-) to the left of an attribute or operation name as shown below.**

|  |
| --- |
| Class-A |
| - Attribute1- Attribute2 |
| - Operation1( )- Operation2( ) |

* 1. **Inclusions**

**Inclusion is a way of reusing the “use case” by using the steps from one “use case” as part of the sequence of steps in another “use case” as shown below.**

****

* 1. **Extensions**

**Extension relationship declares that the use case at the tail of the dotted arrow completely reuses all the steps from the use case being included.**

****

1. **[5 pts] You were given a handout on software project management by Fairley. Describe how the following concepts are used in your project.**
	1. **Architecture Decomposition View (ADV)**

**ADV was used by first defining the problem as FR0, second decompose FR0 to more detailed levels, and third map each FR to a DP.**

* 1. **Work Breakdown Structure (WBS)**

**WBS was used by dividing application architecture in layers and then divide each layer into components.**

1. **[5 pts] In Object oriented (OO) design, describe the concept [with symbol]:**
	1. **Association**

**In a use case diagram, an association line joins an actor to a use case.**

****

* 1. **Generalization**

**Generalization is a relationship in which one use case inherits the meaning and behaviors of another.**

****

* 1. **Dependency**

**Dependency is a relationship in which one class uses another.**

****

* 1. **Realization**

**Realization shows a relationship between an Interface and the class that provides the implementation for the interface.**

****

* 1. **Annotation**

**In UML, the annotation is the note that enables the attachment of constraints, comments, requirements, and explanatory graphics in the models**

****

* 1. **Interface**

**An interface is a set of operations that specify some aspect of a class's behavior; it is a set of operations that a class presents to other classes.**

****

1. **[5 pts] Define each performance attribute:**
	1. **Efficiency**

**Efficiency describes software that does not make wasteful use of system resources such as memory and processor cycles.**

* 1. **Flexibility**

**Flexibility refers to designs that can adapt when external changes occur.**

* 1. **Integrity**

**Integrity refers to maintaining the correctness and consistency of data or system.**

* 1. **Security**

**The security of a system is a judgment of how likely it is that the system can resist accidental or deliberate intrusion.**

* 1. **Maintainability**

**Maintainable software is software that can be adapted economically to cope with new requirements.**

* 1. **Portability**

**Portability is a property that reflects how software can be used from one platform to another with little or no change.**

* 1. **Reliability**

**The reliability of a system is the probability, over a given period of time, that the system will correctly deliver services as expected by the user.**

* 1. **Usability**

**Usability reflects how easy it is to use the system.**

1. **[5 pts] Describe how you have used your concept map and basecamp tool to organize your work as a team using RUP as a guide.**

**I have used concept map to organize the project according to RUP development phases and attach all documents pertaining to the project to help me find what I need and update document easily. Basecamp was used as a communication tool to communicate with other team members and give every team member the flexibility to update the project with work assigned to him or her whenever convenience.**

1. **[5 pts] You are working on SWEBOK reviews KA-1 through KA-11. Which three areas apply to your role in the project and why?**
	1. **Area 1**

**Software requirement area applies to me because I helped broke down the project into functional requirements and find all actors and use cases for the project.**

* 1. **Area 2**

**Software Design area applies to me because I designed the application architecture of the project.**

* 1. **Area 3**

**Software management area applies to me as the project manager. I managed all aspects of the project.**

1. **[10 pts] Other than class and component diagrams that all teams are required to use, select 3 UML behavior diagram types that your project could use and why?** 

**Use case diagram**

**Our project will use “use case diagram” because it is a sponsored project that needs to be extensively documented to avoid missing any requirement. I also consulted with Professor Sedlmeyer who wrote the Vision Document for the project and he suggested that I capture the interaction between actors and use cases.**

**Sequence diagram**

**Our project will use “sequence diagram because we want to be able to test each module of our project using scenarios to help us understand how each module will function. This will also help us validate our design.**

**Package diagram**

Package diagram will help us regroup similar model elements to simplify complex diagrams.

1. **[10 pts] We reviewed UML and related concepts for automation in software systems engineering.**
	1. **What is the relationship between UML and SysML?**

**Both UML and SysML are general purpose modeling language defined by OMG (object Management Group). However, the Systems Modeling Language, is used to specify complex systems that include hardware, software, data, personnel, procedures, and facilities. On the other hand, UML is mostly used** **to model the business process, structure, behavior, and architecture of software system.**

* 1. **What is executable UML?**

**According to** [**www.xUMLbook.comweb**](http://www.xUMLbook.comweb) **site, executable UML is a language used to produce a comprehensive and understandable model of a solution independent of the organization of the software implementation. It is a highly abstract thinking tool that aids in the formalization of knowledge, and is also a way of describing the concepts that make up abstract solutions to software development problems.**

1. **[10 pts] Describe how this course has helped you organize your team**
	1. **Management**

**I have learned a lot of management skills that I did not have before. For instance, I never managed any project before I took this class. This class has taught me how assign task to team members, how to share my ideas with other members, how to hold brainstorming session with team members, how to set objectives every week and how to evaluate those objectives.**

* 1. **Architecture**

**I have learned how to decide on the type of application architecture to use based on the type of system you are trying to build. I have also learn how design application architecture from a generic application architecture instead of building everything from scratch.**

* 1. **Detail design**

**I have learned how to design software system using AD approach and RUP software development process.**

* 1. **Documentation**

**I have learned how to document software using all five views (process, physical, use case, logical, development) of UML. I have also learn how to document software using IEEE-380 SRS, IEEE-1058 PMP, and IEEE-1016 SDD.**

**Bonus: [10 pts] List up to 10 aspects of this course you enjoyed/learned from the most?**

1. **The approach that was used to teach this class (hands on approach).**
2. **The spot checks every day. This forced to me to prepare myself every day knowing that I will be asked questions regarding the course or the project.**
3. **The things that I have learned by being project manager for my group.**
4. **The confidence that I have gained in the area of software engineering.**
5. **I enjoyed working with other team members.**
6. **Things that I learned from other team members.**
7. **I enjoyed using Acclaro as a tool for applying AD methodology.**
8. **From this class, I decided to do more research on Axiomatic Design and I purchased a book called Axiomatic Design Advanced Applications by Nam Pyo Suh. This book contains more information that will help me in many ways.**
9. **This course has reinforced my decision to study software engineering in graduate school.**
10. **I enjoyed knowing Cyber Physical Systems and learning more about it.**