CSCI 360 – Software Architecture and Design

Midterm Study Guide

1. OO Analysis and Design
    a. Differences between software processes
    b. Why did we develop software process?
    c. Why do we use one process over another?
    d. What is the difference between architecture and design?
    e. Apply software process <X> to software project <Y>
    f. Iterative development pros and cons
    g. Apply iterations to software project <Y>. Describe what you need to do.
2. Inception
    a. Why is there a phase of inception?
    b. Artifacts that come out of inception and why?
    c. Apply inception phase to software project <X>
    d. When does inception happen in UP? Why?
3. Evolutionary requirements
    a. Difference between evolutionary and waterfall requirements
    b. Why are there different types of requirements? Are they always necessary in every software project?
    c. How do we find requirements?
    d. Where do requirements fit in UP?
4. Use cases
    a. Why do we use “use cases” in software projects?
    b. What are the parts of the use case model?
    c. Where do use cases fit in the UP process? Where does the use case model fit in UP?
    d. Different use case formats
    e. Write a use case for software project <X> in <Y> format
    f. Create a UML use case diagram for software project <X>
5. Other requirements
    a. What is the role of supplementary specification in UP? Why is it useful?
    b. Give an example where a glossary is necessary for the success of a software project.
    c. Give supplementary specification for software project <X>
6. Ethical and legal issues
    a. Intellectual property and open source
    b. National/State and international laws as they relate to software development
    c. A case study related to ethical and legal issues is presented to you. What are the proper actions?
7. Iteration
    a. What are the guidelines to transition from inception to elaboration and in general between the phases of UP?
8. Domain Models
    a. What is the motivation behind creating a domain model?

    b. What are the pros and cons of a domain model?
    c. Create a domain model diagram for software project <X>
    d. Pros and cons of the domain model
    e. Fallacies and pitfalls of domain models
9. SSDs
    a. Difference between interaction diagrams and SSDs
    b. Given a main scenario <X> create the SSD
    c. Given an exception case scenario <Y> create the SSD
    d. Where do SSDs fit in UP?
    e. Why do we need SSDs?
10. Operation Contracts
    a. When do we create OCs? Where do they fit in the UP?
    b. Given SSD <X> write the Operation Contract for all the operations.
    c. Difference between pre and post conditions.
    d. When, where, and why are OCs useful?
11. Logical Architecture
    a. Why do we need a Logical Architecture?
    b. Why layers? What are the fundamental differences between layers?
    c. What are the criteria to create layers and packages?
    d. Given software application <X> create a logical architecture diagram with layers, subsystems, and packages.
12. Interaction diagrams
    a. What is the difference between dynamic and static diagrams?
    b. When would you use a collaboration diagram instead of a sequence diagram?
    c. Where do interaction diagrams fit in UP? Why are they needed?
    d. Given software application <X> design a collaboration/sequence diagram.
13. Class diagrams
    a. How do we derive class diagrams? Where do they fit in UP?
    b. What is the difference between class and domain diagram?
    c. What does a dependency mean in a class diagram?
    d. What is the difference between aggregation and composition?
    e. Why LRG design principle should be followed in your class diagram?
    f. Given software application <X> design a class diagram.
14. GRASP
    a. Differences and common characteristics between GRASP patterns
    b. Apply the appropriate design pattern given a sequence diagram, use case main scenario etc.
    c. Rewrite a diagram with the principles of low coupling, high cohesion
    d. Where do design patterns fit in the software process and UML diagrams?
    e. Apply design patterns to a specific set of objects for software <X> and justify your decisions based on GRASP