Quiz 1 – 2015. 4.27

①Which UML diagrams are useful for analysis modeling? Provide an example of each.

②List three characteristics that can serve as a guide to evaluate design quality

③List the four design models required for a complete specification of a software design and the role of each.

Use-case diagram
Activity diagram
Class diagram
State diagram

- Design implements all explicit requirements from the analysis model, as well as accommodating implicit customer requirements.
- Design must be understandable to the people who generate the code to implement design, those who test it, and those who support it.
- Design must provide a complete picture of the software, addressing the data, functional, and behavioral domains from an implementation perspective.

- Data design - high level model depicting user's view of the data or information.
- Architecture design – shows relationships and collaborations among specific analysis model software and hardware elements
- Interface design - interface depicts a set of operations that describe the externally observable behavior of a class and provides access to its operations
- Component-level design - Describes the internal detail of each software component

(1) How does the object-oriented view of component-level design differ from the conventional view?

The object-oriented view focuses on the elaboration of design classes that come from both the problem and infrastructure domains. Classes are elaborated by specifying messaging details, identifying interfaces, defining attribute data structures, and describing process flow for operations. In the traditional view, three of components are refined: control modules, domain modules, and infrastructure modules. This requires representations to be created for data structures, interfaces, and algorithms for each program module in enough detail to generate programming language source code.

(2) List four interface design issues present in the development of most user interfaces.

- System response time
- User help facilities
- Error information handling
- Menu and command labeling
- Application accessibility
- Internationalization

(3) What are the attributes of a good software test?

- Has a high probability of finding an error
- Not redundant
- Should be capable of uncovering a whole class of errors
- Should not be too simple or too complex

(4) Describe the differences between black-box testing and white-box testing.

- Black-box testing involves testing the functionality of a software component without knowing the details of its internal logic. White-box testing involves testing the independent logic paths with full implementation knowledge.

(5) List four types of systems tests.

- Recovery testing
- Security testing
- Stress testing
- Performance testing