CMSE-201 problem session 06.06.2022

Ch 11 Configuration management

1. What the configuration is? What is configuration item? What is chek-in and check-out?
2. What are the four configuration management activities?
3. What is version? What is baseleine? What is codeline? What is mainline?
4. What is branching and merging? How versions are identified?
5. What is storage management using deltas? What version is stored in full? How other versions are created?
6. What the build server is? How minimal re-compilation is reached?

Ch 10 Quality management

1. What are the aims of quality management on an organization level? Project level?
2. What is the aim of the quality management process?
3. How shall be composed quality team?
4. What a quality plan is? What is its structure?
5. What is “software fitness for purpose”?
6. How to resolve quality conflicts?
7. How product and process quality are related?
8. What is process-based quality?
9. Why standards are important? What are the five types of standards?
10. What are the core processes defined in ISO 9001?
11. What are the requirements of ISO 9001 certification?
12. What are the three review types and their objectives?
13. What are the three phases of review activities?
14. What is inspection check-list?
15. What are the three weaknesses of the pairwise programming?
16. What is fog index? <https://corporatefinanceinstitute.com/resources/knowledge/other/fog-index/> ; <https://www.howmanysyllables.com/syllables/syllable> ; <https://en.wikipedia.org/wiki/Gunning_fog_index>
17. What is the cyclomatic complexity? <https://en.wikipedia.org/wiki/Cyclomatic_complexity>
18. What are the three types of the process metrics?
19. Why reliability depends on cyclomatic complexity, number of lines of code, and number of error messages?
20. What is weighted methods per class? Sum of cyclomatic complexities.
21. How the process of product measurement is structured?

Ch 9 Testing

1. What are the two goals of testing?
2. What is static testing? Dynamic testing?
3. What are the four advantages of static inspection?
4. What are the three stages of testing?
5. What are setup and assertion parts of a test?
6. What are normal and abnormal inputs?
7. What is partition testing?
8. What is guide-line based testing? What are the five general testing guidelines?
9. What are stubs and drivers? When they are used and what for?
10. What is interface testing? What are the four types of interfaces?
11. What are three types of interface errors?
12. What are five interface testing guidelines?
13. What is Top-down Breadth-First system testing? Depth-first?
14. How use-case description can be used for testing?
15. What is test-driven development?
16. What is regression testing? Why, after integration, previously tested subsystems shall be re-tested?

Ch 8 System modeling

1. What is context model? What is use-case diagram?
2. What is UML activity diagram? How processes are represented? How barrier synchronization is represented? How start and end nodes are shown? How decision node is represented?
3. How use-cases are represented by structured text (template)? What six questions shall be answered? What is main success scenario? What is alternative scenario?
4. How sequence diagrams can be used for use-case scenarios description? How main scenario is represented? How alternative scenario is represented? How time-line is represented? How period of an actor activity is represented? How message exchange is represented? How response is represented?
5. What is an object class? What three parts class definition has?
6. How associations and mapping cardinality constraints are defined for object classes? How they are represented?
7. What is specialization/ generalization relation of classes? How is it represented?
8. What is aggregation relation of the classes? Composition? **Aggregation** implies a relationship where the child can exist independently of the parent. Example: Class (parent) and Student (child). Delete the Class and the Students still exist.**Composition** implies a relationship where the child cannot exist independent of the parent. Example: House (parent) and Room (child). Rooms don't exist separate to a House
9. What navigable association of classes is? The normal kind of association is bidirectional by default.  This means that if an association exists between two classes, then both objects know about each other.  However, when creating a UML class diagram it is possible to use a navigable association between two classes.  This is shown by adding an arrowhead at one end of the association.  If “A” is the source class and “B” is the target class, the arrowhead would be placed on the “B” side of the association. A navigable association of this type means that at runtime object “A” knows about object “B”, but object “B” has no knowledge of or visibility into object “A”.
10. What is dependency class association? A depends on B if change of B affects A.
11. How coupling of objects is calculated?
12. What the package diagram is?
13. What the communication diagram is?
14. What the data-flow diagram is?
15. What the state machine diagram is? How initial and end states are shown? When transition happens? How actions are specified on entry? On exit? How sub-states are represented?