Final Exam Review

1. Give a sentence or two informally defining each of the following terms:
   - Software engineering
   - Software process
   - Functional requirement
   - Nonfunctional requirement
   - Requirements elicitation
   - Derived requirement
   - REST architecture
   - Model-driven architecture
   - Architectural pattern
   - Verification
   - Validation
   - Software maintenance

2. List the four fundamental software engineering activities, and include a short sentence summarizing what each activity does.

3. Give at least two software engineering “best practices” or processes meant to aid in the fundamental software engineering activities given below.

4. Draw the major stages of the waterfall process model.

5. State how each of the practices of XP given below can improve at least one factor of PQCT.

6. State how each of the software engineering processes or concepts given below can help improve at least one factor of PQCT.

7. Give the typical “life cycle” for a user story in a Scrum environment. (Practice Note: here I’m looking for a short description of how user stories are created and “updated”, as well as who performs each activity. For instance, the product owner creates the user story and places it in the product backlog, the user story is moved to the sprint backlog by the product owner and development team during a sprint planning event, the story is completed during the sprint by a developer, and so on.)

8. What is the difference between a domain model and a design model?

9. Give one reason Sommerville gives for why model-driven architecture has not been a “mainstream approach” to software engineering.

10. In class, we discussed class, state machine, sequence, and use case diagrams. Are these used to convey design models, domain models, or both?

11. What’s the difference between architecture and design?

12. What’s the difference between a model and a diagram?

13. What is the difference between the layered architectural pattern and the client-server architectural pattern?
14. What is the relationship between “agile software development” and “eXtreme programming”?

15. Give one way that software engineering differs from (“simple” software development, computer science).

16. The user stories below fail to meet at least one of the INVEST criteria. Identify one criteria each story does not meet and explain why it does not meet it. Note: I have left off the acceptance criteria for brevity. Do not consider this a violation.

17. Each of the requirements below fail to meet at least one of the IEEE’s criteria for a good single software requirement. Identify one criteria each requirement does not meet and explain why it does not meet it.

18. Consider the traceability matrix given below which maps system requirements to software requirements. Identify two errors or “smells” indicated by the matrix.

19. Answer the following questions regarding the (sequence, state machine, class, use case) diagram given below.

20. Which one of the four types of software maintenance best classifies each of the activities below?

21. Give an example scenario where a (plan-driven, change-driven) process model would be preferred over a (change-driven, plan-driven) model. Include a description of why the (plan-driven, change-driven) process model is preferred in your scenario.

22. Consider the (waterfall phase, Scrum task, XP practice) given below. Provide a paragraph describing how it helps meet one of the four “objectives” we gave for software engineering (PQCT – increase productivity and quality, reduce cost and time to market).

23. Write a user story (including acceptance criteria) that might have been created during the development of Google Drive. Size your story so a single developer could implement it in a week (40 hours). Your user story should satisfy the INVEST criteria.

24. Write a set of requirements for the simple method given below. Your requirements should satisfy the criteria specified by the IEEE for “good” requirements.

25. Enter a sequence of Git commands which would create the commit structure shown below. You may assume you are in the correct directory, and may simply leave comments where file creation and modification would take place.

26. Give two activities where the use of a model may be helpful. Include a description of how the model would be helpful over some other alternative.

27. Facebook uses a tool called “Gatekeeper”, and Google uses what they call “staging servers” and “canary servers”. These tools perform similar tasks to improve reliability. What do these tools do, and how do they improve reliability?

28. Consider the software project described below. State which process model (waterfall or XP) you would choose for the project, and give two reasons why you would make this choice. Include several sentences defending each of your two reasons.

29. Give one advantage and one disadvantage of using (interviews, ethnographies, stories and scenarios) for requirements elicitation.

30. Consider the sample project described below. How would you perform requirements elicitation? Justify your answer.

31. Why is it important for a software engineer to be familiar with architectural and design patterns?

32. Give one reason why tools such as Docker and Vagrant have, by some accounts, “significantly altered” the software verification and validation process.