CISC 7540X Midterm Exam

Name:

Short-answer questions. Each question is worth 5-points. Leaving an answer blank earns 1-point (writing a wrong answer earns 0-points).

1. Describe the underlying principles of Waterfall.

2. Describe the underlying principles of Agile.

3. You are starting a new project and believe Waterfall would work best. Justify your choice.

4. You are starting a new project and believe Agile would work best. Justify your choice.

5. Describe how Use Cases work in documenting user requirements.

6. After reviewing user requirements, you notice some complex use-cases, such as: "customer places an order, contacts customer support, receives order, contacts customer support again, returns order, and gets refund". What's the correct course of action? Justify your answer.

7. While reviewing the requirements, you are not sure about feasibility of some key part of the system. How can you convince yourself that it is in fact feasible?

8. While designing the system, there are several alternatives: the more complicated design that is flexible to future needs, and a simpler design that solves the immediate problem. You decide on one of them. Justify your choice.

9. You need to decide on the programming language for the new system. What factors would go into your decision making?

10. Midway through the implementation, you discover that a key architectural decision has become a major problem. How can the situation be salvaged? How can you avoid this issue on a future project?

11. What is technical debt? Give an example.

12. Why are integration bugs worse than implementation bugs?

13. What is the purpose of a unit-test? Give an example of a unit-test for function $f(x)$.

14. How would you go about unit-testing a SQL query?

15. What architecture is appropriate for an ETL project? Explain.

16. Briefly describe Object Oriented Architecture. Give an example of a project where Object Oriented architecture would be a bad choice.

17. Briefly describe Data Centric Architecture. What projects is it appropriate for?

18. You need to add a feature to the last production release of the system. From source version control perspective (e.g. `git`, `subversion`, etc.) explains the steps that are needed to make the change. Assume that development has continued after the last production release.

19. You are building an elevator controller. How can you prove to yourself that your software will not kill anyone?

20. Explain cost-of-quality equation: when is it worthwhile to ship software with known defects?