

CSCE 740 - Practice Midterm

Name:

This is a 75-minute exam. On all essay type questions, you will receive points based on the quality of the answer - not the quantity.

Make an effort to write legibly. Illegible answers will not be graded and awarded 0 points.

There are a total of 11 questions and 100 points available on the test.

Question 1—7 Points.

Briefly explain why a software system must change or become progressively less useful.

Question 2—7 Points.

The properties of the environment of a system are generally of critical importance for the system under development to be able to satisfy its stated system requirements. Therefore, it is essential to capture environmental assumptions in a requirements document. Briefly discuss how the environment may influence a system's ability to satisfy its requirements.

Question 3—8 Points.

Pick two of the following key tenets of extreme programming and briefly explain why each is important:

1. Collective Ownership
2. Sustainable Pace
3. Open Workspaces
4. Customer as a Team Member
5. Test First Design
6. Short Iterative Cycles
7. Stories as Requirements

Question 4—12 (7+5) Points.

You are involved in a development of a new software product. The product is an insurance application intended to determine what insurance products a potential customer is eligible for. The eligibility requirements are captured in various laws and regulations. An external contractor is doing most of the work. Your organization has hired this contractor to assist with all aspects of planning, management, and development since your organization is lacking the expertise to complete the project on its own. The plan was to do a traditional waterfall process. The product will be long lived and good documentation is a must. In addition, the laws and regulations were thought to constitute a good start for the requirements of the project—thus, a waterfall process was deemed the best choice.

During the requirements capture process, the team discovers that the laws and regulations are incomplete, ambiguous, and generally obtuse—the capture is taking way longer than planned and required a lot of interaction with case workers that possess the knowledge of how these laws and regulations are applied in practice. Towards the end of the requirements process, the requirements document is incomplete and there is much more requirements work to be done.

At this point, the contractor decides that since there is so much requirements risk and we are far behind schedule, the project will switch on an agile method—eXtreme Programming—in an attempt to recover. In addition, to save money, the contractor is offshoring the coding efforts to a development center in a low-cost country.

Part 1:

Is this approach likely to succeed? Briefly justify your answer.

Part 2:

What would your recommendation have been? Briefly explain why you provide that recommendation.

Question 5—10 (6+4) Points.

Part 1:

Explain the difference between *validation* and *verification*.

Part 2:

Validation is generally considered harder. Why?

Question 6—9 Points.

You are all familiar with an Automatic Teller Machine (cash machine). Please draw a use-case diagram with at least three (3) use-cases and the actors involved in these use-cases.

Question 7—8 Points.

Pick one of the use-cases from the previous question and write down the typical scenario for this use-case.

Question 8—15 Points.

The airport connection check is part of a travel reservation system. It is intended to check the validity of a single connection between two flights in an itinerary.

`validConnection(Flight arrivingFlight, Flight departingFlight)` returns `ValidityCode`.

A `Flight` is a data structure consisting of:

- A unique identifying flight code (string, three characters followed by four numbers).
- The originating airport code (three character string).
- The scheduled departure time (in universal time).
- The destination airport code (three character string).
- The scheduled arrival time (in universal time).

There is also a flight database, where each record contains:

- Three-letter airport code (three character string).
- Airport country (two character string).
- Minimum connection time (integer, minimum number of minutes that must be allowed for flight connections).

`ValidityCode` is an integer with value 0 for OK, 1 for invalid airport code, 2 for a connection that is too short, 3 for flights that do not connect (`arrivingFlight` does not land in the same location as `departingFlight`), or 4 for any other errors (malformed input or any other unexpected errors).

In order to design requirements-based test cases, perform input partitioning and derive representative values of the parameters from this specification for the `validConnection` function.

Question 9—8 (4+4) Points.

The following requirements are unclear and ambiguous. Explain why, and then rewrite the statements so that they can be objectively evaluated.

- a. The response time should be minimized.
- b. The alarm should be raised quickly after a high fuel level has been detected.

Question 10—8 Points.

In class we discussed the importance of defining a test case for each requirement. What are the two primary benefits of defining this test case?

Question 11—8 Points.

You are setting out to develop a new GUI for an old application. The system has a diverse set of users and the system has to be acceptable to all of the user types.

What development process would you use? Justify your answer.