**CS 347 Homework 1 (Solution) 100 Points**

Answer all questions in a document. Be clear and describe your answer well and convincing. Most answers require a paragraph or several thoughtful sentences to convey the response.

Use MS, PDF, or other popular formats. Upload your answer in this Canvas Homework. Questions are from problems at the end of Chapters and numbered as in the 9th Edition. There are ten questions, each has 10 points.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1.6. As software becomes more pervasive, risks to the public (due to faulty programs) become an increasingly significant concern. Develop a doomsday but realistic scenario in which the failure of a computer program could do great harm, either economic or human.

ANSWER: There are literally dozens of real-life circumstances to choose from. For example, software errors that have caused major telephone networks to fail, failures in avionics that have contributed to plane crashes, computer viruses (e.g., Michelangelo) that have caused significant economic losses and attacks on major e-commerce sites.

2.8. Is it possible to combine process models? If so, provide an example.

ANSWER: The process models can be combined, each model suggests a somewhat different process flow, but all perform the same set of generic framework activities: communication, planning, modeling, construction, and delivery/feedback. For example, the linear sequential model can serve as a useful process model in situations where requirements are fixed, and work is to proceed to completion in a linear manner. In cases, where the developer may be unsure of the efficiency of an algorithm, the adaptability of an operating system, or the form that human-machine interaction should take. In these, and many other situations, a prototyping model may offer the best approach. In other cases, an incremental approach may make sense and the flow of Spiral model may be efficient. Special process models take on many of the characteristics of one or more of the traditions.

2.9. What are the advantages and disadvantages of developing software in which quality is “good enough”? That is, what happens when we emphasize development speed over product quality?

ANSWER: The advantages of developing software in which quality is “good enough” is that the product or software will meet the deadline, it may however lead to the delivery of software that is low in quality and requires time to improper the quality. When speed is emphasized over the product quality it may lead to many flaws, the software may require more testing, design and implementation work then done. Requirements may be poorly defined and may need to continuously change. Halfhearted and speed may cause the risk management to fail to detect major project risks. Too little quality may result in quality problems and later rework.

3.2. Describe agility (for software projects) in your own words.

ANSWER: Agility can be applied to any software process. However, to accomplish this, it is essential that the process be designed in a way that allows the project team to adapt tasks and to streamline them, conduct planning in a way that understands the fluidity of an agile development approach, eliminate all but the most essential work products and keep them lean, and emphasize an incremental delivery strategy that gets working software to the customer as rapidly as feasible for the product type and operational environment.

5.1. Based on your personal observation of people who are excellent software developers, name three personality traits that appear to be common among them.

ANSWER: Answers will vary. Three traits must be identified.

6.6. Of the eight core principles that guide process (discussed in Section 6.1.1), do you believe one is more important?

ANSWER: The software design model is the equivalent of an architect’s plans for a house. It begins by representing the totality of the thing to be built (e.g., a three-dimensional rendering of the house) and slowly refines the thing to provide guidance for constructing each detail (e.g., the plumbing layout). Similarly, the design model that is created for software provides a variety of different views of the system, hence it is always necessary.

7.1. Why is it that many software developers don’t pay enough attention to requirements engineering? Are there ever circumstances where you can skip it?

ANSWER: Understanding the requirements of a problem is among the most difficult tasks that a software engineer face since requirements change continuously, hence they tend to pay little attention to it. In some cases, an abbreviated approach may be chosen. In others, every task defined for comprehensive requirements engineering must be performed rigorously. Requirements engineering builds a bridge to design and construction and cannot be skipped.

7.5a. Develop a complete use case for making a withdrawal from an ATM.

ANSWER: Answers will vary, but it should meet the criteria for Use Cases as stated in the textbook.

Here is an example I found online, I like it more than others I saw on internet for simplicity, atomic, and clarity. It has also presented the exception cases as Alternate Flows, which is a very clear way of stating use cases: <http://www.selmanalpdundar.com/atm-withdraw-cash-use-case.html> We do not expect you to provide the exceptions or alternate flows to get full credit.

8.1. Is it possible to begin coding immediately after a requirements model has been created? Explain your answer, and then argue the counterpoint.

ANSWER: The analysis model will serve as the basis for design and construction of the domain objects. It is possible to begin coding after the objects and attributes are defined, and relationships are known.

8.10. How does a sequence diagram differ from a state diagram? How are they similar?

ANSWER: State diagrams depict the state of the system and show how events affect system states. Sequence diagrams indicate how events cause transitions from object to object.