Honor Pledge: “I pledge my honor that I have abided by the Stevens Honor System.”

Midterm Exam 1: 100 Points   Name: Julia Nelson
Section B

1. (10 Points) How can Software Development Process patterns assist a development team build software products efficiently?

   ANSWER:

   Software development processes patterns are templates to organize, plan, and prevent incorrect/unnecessary work during the software development process. Different software development process patterns can assist a development team build efficiently depending on their specific needs/requirements. Some provide more communication for those with unclear requirements, others allow for multiple iterations providing growth off of previous designs and prototypes. Each pattern has different advantages and disadvantages for some software development process that increase efficiency, but they can also be combined to better suit the team’s process.

2. (10 Points) Explain what is wrong with the notion that computer software does not need to evolve over time.

   ANSWER:

   To say that computer software does not need to evolve over time is incorrect and could leave that software at risk of bugs, faults, failures and security attacks. Software that does not evolve would not be able to meet newer requirements by customers as well as keep up to date with better and more current performance requirements. Software that doesn’t evolve could face a higher risk of security attack, using out of date security measures.

3. (10 Points) Why is it important for software processes to be agile?

   ANSWER:

   To be agile is to be “quick and ready” and is important for its improvement of software processes when adapted. The team of an agile process is able to quickly develop and meet all of the requirements. In addition to improved efficiency, the software is also more prepared for any possible additions or changes through its self-organizing team (meaning strong, knowledgeable work), that heavily relies on communication to ensure that all customer requirements are met.

4. (10 Points) Describe the role of customers and end-users on an agile process team?

   ANSWER:

   Customers and end-users on an agile process teams provide the necessary information for the team to meet their requirements. They will continually communicate and collaborate throughout the process. The customer provides their functionality requirements and businesses problem that this software is solving, while end-users provide feedback and collaboration on the models and prototypes to suggest improvements.

5. (10 Points) Why should requirements engineering be an iterative process?

   ANSWER:

   Requirement engineering needs to be an iterative process in order to fully understand, analyze and develop requirements to their best capabilities and to be as in-depth and fully described as possible. The iterative process allows for a general cycle of the development process the build a foundation of the software. It then iterates, returning to add onto the previously created model and requirements.

6. (10 Points) List at least five characteristics all software engineers should poses?

   ANSWER:

   Software engineers need to possess a range of strong and positive characteristics that can be beneficial to a development team. However, all software engineers should have strong communication skills, be a strong/positive team member, be agile, detail-oriented, creative, and most importantly, highly motivated. These characteristics provide for strong, efficient and customer-need oriented team members and creative new solutions to software development problems.

7. (10 Points) Why are nonfunctional requirements important to the requirements engineer?

   ANSWER:

   Non-functional requirements make the system usable and are important to requirement engineers because they are the details that decide if a software is perfect or just good-enough. These requirements can include industry standards, reliability, availability, performance, maintainability, and security; all of which help ensure usability of the software and safety of the user. Non-functional requirements also help create the basis of the test cases.

8. (10 Points) What is the most significant result of the written requirements engineering process?

   ANSWER:

   The most significant result of the written requirements engineering process is the fully understood, precise, collaborated, and agreed-upon requirements of the system. It ensures there is no ambiguity between the development team and the customer. It essentially creates a contract and concrete plan description of the software and can then start the next step.

9. (10 Points) Which UML diagrams are useful in scenario-based modeling?

   ANSWER: Use Cases are useful UML diagrams in scenario-based modeling. They are specific interaction cases between the software and an actor with a goal in context. They can be used to identify normal usage, unusual events, and possible fault use cases. They allow an insight into the projected final functionality of the software.

10. (10 Points) What are the most important use of Use Cases as we go from written Requirements phase to Requirements Analysis and Modeling phase?

    ANSWER:

    Use cases are important between the written Requirements phase and the Requirements analysis and modeling phase. They clarify the requirements recently created in the written requirements section, as well as raise-light to any missing or incorrect use case. Use cases are easy to comprehend and provide detail into the collaboration of components of the software and visually representing them.