Santiago Bermudez 301118090

1. Explain the difference between Bespoke and Generic Software with your own examples.

Bespoke software is software that is generally developed to meet the specific needs of an individual client or specific consumer, like if Amazon were to use software to handle its inventory management. Generic software is software developed for a broad market and to meet the needs of the masses, rather than just an individual client. An example of generic software can be email, as it is made for everyone and used by virtually everyone.

2. Explain "Software" and explain "Engineering" and combine the two to define "Software Engineering".

Software is just a sequence of statements that can be executed on a computer, or a collection of human-readable statements that can be converted into such a sequence. Engineering is the application of methods and tools to design, create, and maintain products. Putting these together, software engineering is the systemic application of engineering discipline to develop software.

3. List the two areas of concern in Software Engineering. Pick two concerns in each area (total four concerns) and describe in your own words the concerns of your choice.

The two areas of concern in software engineering are managerial concerns and technical concerns. Two managerial concerns are project cost (whether or not the project can be finished within a certain budget) and project resources (which are resources that need to be obtained and ready during a project and may also need to be maintained and improved over the course of a project). Two technical concerns are software requirements (which are the functions,

characteristics, or properties that a software must have) and software design techniques (which could be the process of envisioning and defining software solutions to one or more problems).

4. There are several questions that you will ask before developing a software product in your company. Please choose any two questions and answer in your own words how you would answer the questions if you were given the responsibility.

If asked “How exactly should the program interact with its users?”, I would state that, “The program should be simple and helpful like a compass”. If asked “What standards and practices should be used in writing the code?”, I would say, “It is important to use commenting and to know how to debug well as one may not write the perfect code on the first try”.

5. Name two issues you may face while developing large products and answer how you will address these large scale product or project issues.

Two issues I may face are that code for large programs is enormous (millions of lines) and that testing large programs is extremely difficult because there are so many possible execution paths. For coding I would encourage people to accomplish things with as few lines of codes as possible and to make them reusable so we don’t have multiple codes doing the same thing. For testing, I would encourage modular programming and to focus on compiling the interchangeable modules rather than the whole program if possible.

6. In the Project Management Iron Triangle, explain what it means to have Quality at the center of the triangle.

Generally, it is assumed that quality will never be compromised (although it could be), which is why it is pictured in the center of the triangle. However, if you change one of three

things like scope, time, and cost, then the other two must be changed as well, with quality remaining more consistent.

7. Google and research the Scrum Alliance and Agile Alliance and choose one agile value out of the four Agile values and one agile principle out of the twelve agile principles and explain how you would creatively follow the two choices at your company in your own words.

The agile value I would follow is “customer collaboration over contract negotiation” and I would do so with public surveys, interviews, and polls to have customers participate as customers who get more involved in their products might be more vested in their outcomes. The agile principle I would follow is “build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.” I would follow this principle by giving people their space and avoiding micromanagement.