**Chapter 1**

1.1 What are some of the major objections and deterrents to proper requirements engineering activities?

Some major objections and deterrents to proper requirements engineering activities include natural language problems, not distinguishing your role as a requirements engineer from your official title, and gold-plating requirements. Natural language problems, specifically ambiguity of an informal set of requirements, can make it difficult to pick out the true goals of the system. The inability to separate the role of requirement engineer from your official title may make it hard to start the process of requirements engineering to begin with, especially when your title is of a more technical nature. You are more likely to begin developing the system itself rather than the initial requirements. Finally, gold-plating can make the system requirements too complex and less likely to be followed closely. Since the requirement is unnecessary, it can distract from what the actual purpose of the system should be. Therefore, it makes the requirements engineering process more difficult.

1.2 How is requirements engineering different for “small” systems?

Requirements engineering is different for a small system mostly because of complexity. The less complex a system is, the easier it is to define accurate and realistic requirements. There are less features that need to be accounted for. For example, if a single room was a system, it is easier to define the requirements for the room then if you were to make a larger system, a house. For the room, you only have to worry about its function, what it looks like, what’s inside of it, etc. However, for a house, you have to consider multiple rooms, which means multiple subsystems. You have to know where all the rooms are going to be placed, how they are going to be connected with utilities, how many and what kinds of people are going to be in the house, etc. Each of these requirements of the larger system are complex in their own rights.

1.5 What are some factors that may cause customers to alter requirements?

Customers can change their requirements for many reasons. Some of these include customers not having a clear vision of the system, a poorly designed system, or a reaction to the current state of the system. When a customer does not have a clear vision of the system, they may be more likely to suggest new requirements as they think of them. This can lead to scope creep and make the system too complex for its needs. A customer that has a poorly designed system may be under the impression they have thought of all the needs for the system, but as they receive more input about the system, they may realize there is a lot missing and wish to change the original set of requirements.

1.6 What issues might arise when a requirements engineer, who is not a subject matter expert, enlists a subject matter expert to assist in defining requirements?

One issue that may arise when a non-SME requirements engineer enlists the help of a SME to define requirements is complication with domain specific language. The SME may use words with a unique meaning in the context of the domain that the non-SME confuses for its more common definition. Chapter 1 has a great example of this when talking about the international trucking company (starting on page 8). Another issue is a possible knowledge gap. The SME may define requirements at a very high assuming the engineer knows or can infer the implied requirements. An example of this would be if an engineer is designing a stoplight. The SME may have a requirement such as “The stoplight shall switch between the colors in the correct order.” In this case, the SME assumes the engineer a, knows the correct colors of the stoplight and b, knows the correct order of the stoplight.

**Chapter 2**

2.1 Why is it important to have a concept of operation or mission statement at the start of requirements engineering?

A good conops or mission statement is beneficial when starting requirements engineering because it provides the overall focus and purpose of the system. It will help to form and guide the expectations of the stakeholders. It will also provide the requirements engineer with context when trying to determine the wants and needs of the customer. As we know, customers can be confusing and contradicting when explaining their wishes. Having this context will lead you to having more refined and accurate requirements.

2.3 When should a domain vocabulary be established?

Domain vocabulary should be established at the start of the requirements engineering process. This will ensure that all of the different stakeholders are on the same page, thereby reducing any confusion and ambiguity during later stages of requirements engineering.

2.4 At what stage of the requirements development are additions to the requirements considered scope creep?

Additional requirements are considered scope creep when they begin to describe functionality outside the boundaries of the intent of the system. The context diagram is a good way to initially define what those boundaries are. As functional requirements begin to exceed what is depicted or described in the diagram, then those FRs would be part of scope creep.

2.7 Under what circumstances might the customer’s needs and desires be considered secondary?

A customer’s needs and/or desires would be considered secondary when a governing body’s rules, guidelines, or regulations contradict the customer. Another instance would be if the customer’s desires would harm or create serious hazards for other stakeholders. Continuing the baggage system example from the text, if the customer wants the bags to be processed at a specific speed, but moving the bags at said speed would cause bodily harm to the baggage handlers if there was a collision, that desire may have to be disregarded.