# Section 3: Understanding Quality Attributes

## Question 1

Name and describe the six quality attributes discussed in class (slides).

## Answer

1. **Availability**: The ability of the system to remain available in spite of abnormal processing conditions including system failures, incorrect inputs, and others.
2. **Modifiability**: The ability of the system to accommodate changes or extensions to the system’s requirements or environment the system executes within.
3. **Performance**: The ability of the system to meet expected execution requirements such as response time and transactions executed per minute.
4. **Security**: The ability of the system to prevent or detect unauthorized access to services or data maintained by the system.
5. **Testability**: The ease in which defects (bugs) in the system’s components can be identified.
6. **Usability**: The ease in which the system’s users are able to productively use the system.

## Question 2

Describe the meaning / purpose of Stimulus, Response, and Response Measure components of a Quality Attribute Scenario.

How many responses will a typical scenario describe?

## Answer

See Slide 6 or the Section “Quality Attribute Scenarios”.

1. Stimulus describes the event that triggers the scenario. The event is generally a design or runtime problem that the system must somehow deal with. In other words, a fault that the system will keep from becoming a failure.
2. Response is a feature of the system designed to address the Stimulus. That is, an action taken by the system to address event / fault.
3. Response Measure are metrics designed to measure the effectiveness of the Response. This allows several responses to be compared with a tradeoff decision to be made by the customer i.e. Performance vs Cost of implementation.

A scenario will typically present several responses each of which address the stimulus in a differ manner. Again, it is the decision of the Customer / SME to decide which solution best meets their application’s needs.

## Question 3

Match each of these stimuli to one of the six quality attribute it best matches.

1. Minimizing the potential for user entry errors. : Usability
2. Change in system configuration or adding additional features to the system design. : Modifiability
3. Denying unauthorized access to system services. : Security
4. System crash or failure in response to invalid inputs. : Availability
5. Determining the correctness of a completed project iteration. : Testability
6. The arrival rate of work / events to be processed. : Performance

## Answer

Given in question.

## Question 4

Which of the six quality attributes is most likely to be in contention with the other five? Explain why?

## Answer

System performance is often in contention with the other five. This is because maximizing performance often requires comprises to other aspects of the design. For example, isolating and/or encrypting sensitive data requires additional processing to access the data which reduces the number of transaction that can be processed per time unit. Increasing modifiability or availability often involves introducing additional layers of abstraction or indirection in the design which because of additional processing also decreasing performance. Introducing a database server cluster increases DBMS Performance (and Availability) but also increases cost.

## Question 5

Define “Integrity” in a system’s design.

## Answer

Put another way, integrity means ‘doing the same things the same way”. There are usually 1-2 overarching architectural design patterns that should be adhered to throughout the system’s design. If the pattern is ignored or violated by some aspect of the system’s implementation this is a violation of the architecture’s integrity.

For example, if a web application’s architecture uses the three-tier architecture, all the components having access to the database should be located in the ‘service tier’. If a developer embeds a database access in the presentation tier (GUI), it is a violation of the architecture and of the integrity of the system’s design.

## Question 6

What are the benefits that Quality Attribute Scenarios have over Non-Functional Requirements?

## Answer

Non-functional requirements are simple statements that describe some aspect of the system’s design.

Quality Attribute Scenarios are specific in that they:

1. Describe specific conditions (stimulus) that effect the system’s execution.
2. Describe specific features of the system’s design that address the issue or mitigate the effects of a failure.
3. Describe metrics that measure the effectiveness of the proposed solution to allow comparison of several proposed solutions.

## Question 7

What are the six practices that produce quality architectures?

## Answer

1. The architecture should be the product of a single architect or a small team with an identified leader. Move the team towards the system’s design goals.
2. Provide the architect the system’s non-functional requirements and a prioritized list of quality attributes i.e. security > performance > modifiability etc.
3. Communicate the architecture to the stakeholders and ensure the stakeholders understand how their concerns are being addressed.
4. Evaluate the architecture’s ability to meet functional and quality requirements early in the project lifecycle.
5. Maintain the architecture’s documentation.
6. Identify resource bottlenecks early in the project. Performance Bottlenecks i.e. Database performance is usually an issue. Project Resource Bottlenecks i.e. the size of the development team vs. the number of features and delivery schedule.

## Question 8

Briefly describe the Four Structural Qualities of well-designed architectures?

## Answer

1. Cohesive / Loosely Coupled Components: The architecture should be feature components whose functional responsibilities are allocated on the principles of information hiding and separation of concerns i.e. encapsulated, cohesive, and loose coupled designs.
2. Encapsulation w/ Interfaces: Each component should have an interface that encapsulates changeable aspects of the implementation. Interfaces should allow multiple teams to implement their components independently of each other.
3. Based on Architectural Patterns: The architecture should be based on architectural and design patterns. There should be a minimum of wheel reinvention when solid, well understood solutions are available.
4. Maintain the Design’s Integrity: The system should do the same things in the same way throughout. Consistently use the same patterns and control-strategies.

## Question 9 (20 Pts)

For these four quality attributes: Availability, Performance, Usability, & Testability, describe two response measurements (metrics) that can be used to quantify and compare alternatives designs.

## Answer

Note to TA: Please grade this question for 20 points with a total of 110 points for this assessment.

**Availability**: The time needed to recover from a system fault and / or cost of implementing the solution e.g. manual recovery will take longer to recover (but will cost less to implement) than an automated failover.

**Performance**: Any of the three aspects described in Question 10.

**Usability**: Some answers: The amount of time the user needs to complete a task repeated many times during their workday. The number of entry errors a user makes when performing the task. The amount of training needed before a new user becomes proficient with the interface.

**Testability**: Some answers: The number of regressions (broken components / classes) identified during the construction phase. The number of errors uncovered during integration testing or after released into production.

## Question 10

What are the three aspects of a system that can be used to quantify its performance.

## Answer

1. The number of items / requests the system can process over some fixed time e.g. the client requests per minute that a system can maintain.
2. The time the system requires to respond to an event e.g. the average time the system needs to produce a response to a client’s request. AKA Latency.
3. The quantity of work completed over some fixed time e.g. the number of insurance claims processed per day.