Quiz 2

1- What are the metrics used to validate and measure the performance of an application?

- Quality attribute are part of an application's nonfunctional requirements. These requirements capture the many sides of how the functional requirements of an application are achieved. In order to achieve a specific need, a detailed quality attribute requirement must be specified.

2- Is it sufficient for an application to meet the average throughput to guarantee efficiency? Justify your answer.

- No, Throughput is a measure of the amount of work an application must perform in unit time. Therefore, it will be ineffective for an application to meet average throughput. Moreover, this will cause the application to do little work so it would not really be sufficient.

3- Why is security considered as an important quality attribute? What are the main requirements?

- A quality attribute is defined as "an application's requirements in terms of scalability, availability, ease of change, portability, usability, performance, and so on." One of the general software quality attributes is security, security helps in the scalability of the system which relates to different application characteristics; also, it provides authentication for users which extends to handling third party tools.
- A functional security requirement is something that describes functional behavior that enforces security. It can be directly tested and observed. Requirements that have things to do with access control, data integrity, authentication, and wrong password lockouts fall under functional requirements.

4- Distinguish between scale up and scale out mechanisms. Which approach delivers better performance?

- Scale up works well if an application is multithreaded, or multiple single threaded process instances can be executed together on the same machine. However, towards the end it will consume additional memory and some resources, the reason is that processes sometimes need multiple resources at once.
- Scale out works well if there is little or ideally no additional work required managing the distribution of requests amongst the multiple machines. Our purpose is to keep every and each machine equally busy because if one of the machines if loaded to its fullest, more hardware would be wasted.
- If there is little work then scale out will provide better performance; however, if there is a lot of work to be done in an application (multithreaded application), then the scale up approach will deliver a better performance. In my opinion, I would choose scale out since there is a better chance for success and productivity.

5- How are design trade-offs made in the case of conflicting quality attributes?

- A design that satisfies one quality attribute may have some effect on another attribute. in order to achieve greater availability, an available application may have to trade-off lower performance. It isn't plausible to fully satisfy all requirements. Sometimes, an architect ay need to trade-off something for the higher quality of another thing.