1. "Downtime" - the time period in which a service/IT System is unavailable. There can be several causes including scheduled maintenance, software/hardware malfunctions and natural disasters (power outtage, flooding, etc.)

2. "Reliability" - the probability of a system/service functioning as expected over a period of time (no unscheduled downtime, lack of failures). Reliability can be improved in several ways including redundancy

3. "Product measures" - attributes of a product/service/system. Can be static (lines of code, complexity, etc.) and dynamic (execution speed, response time, resources usages)

4. "Stages of design" - includes several steps which are not always the same: idea generation, product research, creating specifications, creating a prototype, testing and quality assurance, launch. The first step of designing a product is realising what you will be making and why. Afterwards you need to research at a deeper level to find out the potential users' needs and desires. After having deeper insight you need to define what the goals of your product will be and then create a prototype to make sure it is possible to achieve. However, it is also very important to rigorously test the product before the final launch

5. "Design quality" - the effectiveness and efficiency of a product/system in fulfilling the intended purpose and satisfying the users' needs. This includes various aspects - functionality, usability, reliability and innovation. However, it also encompasses the ability to change over time and be expanded rather easily.

6. "Benefit of code coverage measurement" - The benefits of code coverage measurement are many. The first and most obvious one is finding and identifying any untested code which helps avoid unnoticed bugs and possible future bugs in the system. Furthermore, this helps make sure that unit and integration tests will be efficient and allow for seamless Continuous integration. Last but not least, a high code coverage allows easier refactoring of code because you can find out if a change has broken some functionality/logic.

7. "Change Request metrics" - used to measure the efficiency and effectiveness of change management process. Include the number of change requests, their nature (bug fixes, new features, enchancements, etc.), the time taken to implement and if they were implemented without introducing new bugs.

8. "Maintenance Quality Metrics" - used to measure the efficiency and effectiveness of software maintenance activities. Include the time to repair, rate of reoccuring issues. Can be used to also measure the reliability of the product/system.

9. "Identify the benefits of applying the metrics" - the most important benefit is helping measure the effectiveness and efficiency of the product and allow decisions to be made using concrete data. This enables better risk management and resource allocation as well as identifying problematic areas. Additionally, this allows investors and stake holders to see how the product is working and any issues with it.

10. "Failure characteristics" - used for diagnosing and addressing issues in a product/system. Include frequency and severity of failures and the root causes of them. identifying them can allow finding problematic areas and prioritize critical issues before they arise.