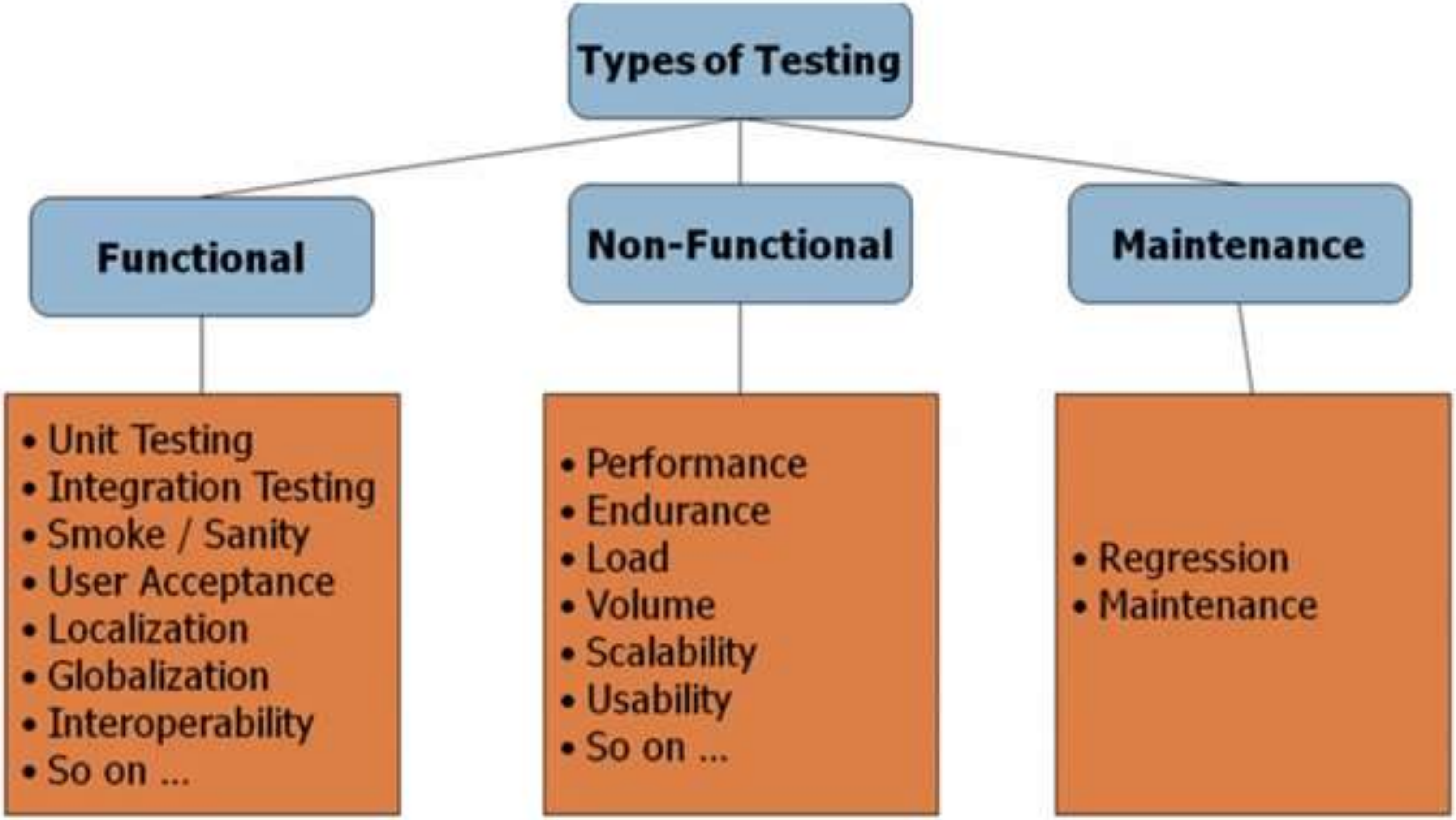
Module 4

Testing Non-functional Requirements

❖ Type of Software Testing

- Functional
- Non - Functional
- Maintenance



- Unit Testing
- Integration Testing
- Smoke / Sanity
- User Acceptance
- Localization
- Globalization
- Interoperability
- So on ...

- Performance
- Endurance
- Load
- Volume
- Scalability
- Usability
- So on ...

- Regression
- Maintenance

- **Non-functional testing** is defined as a type of Software testing to check non-functional aspects (performance, usability, reliability, etc) of a software application
- It is designed to test the readiness of a system as per nonfunctional parameters which are never addressed by functional testing
- An excellent example of non-functional test would be to check how many people can simultaneously login into a software
- Non-functional testing is equally important as functional testing and affects client satisfaction

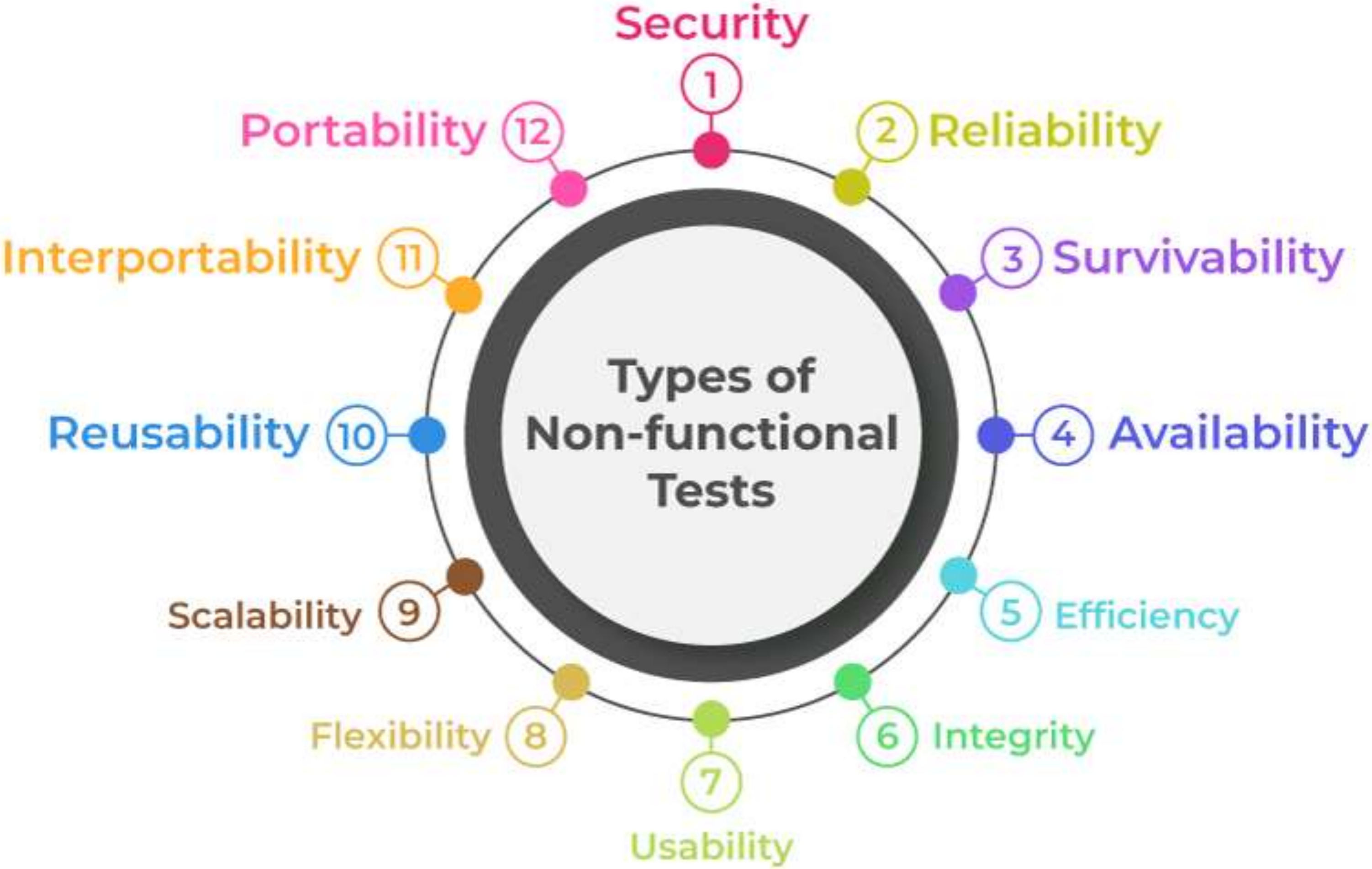
Objectives of Non-functional testing

- Non-functional testing should increase usability, efficiency, maintainability, and portability of the product
- Helps to reduce production risk and cost
- Optimize the way product is installed, setup, executes, managed and monitored
- Collect and produce measurements, and metrics for internal research and development
- Improve and enhance knowledge of the product behavior and technologies in use

❖ Characteristics of Non-functional testing

- Non-functional testing should be measurable, so there is no place for subjective characterization like good, better, best, etc.
- Exact numbers are unlikely to be known at the start of the requirement process
- Important to prioritize the requirements
- Ensure that quality attributes are identified correctly in Software Engineering

❖ Non-functional testing Parameters

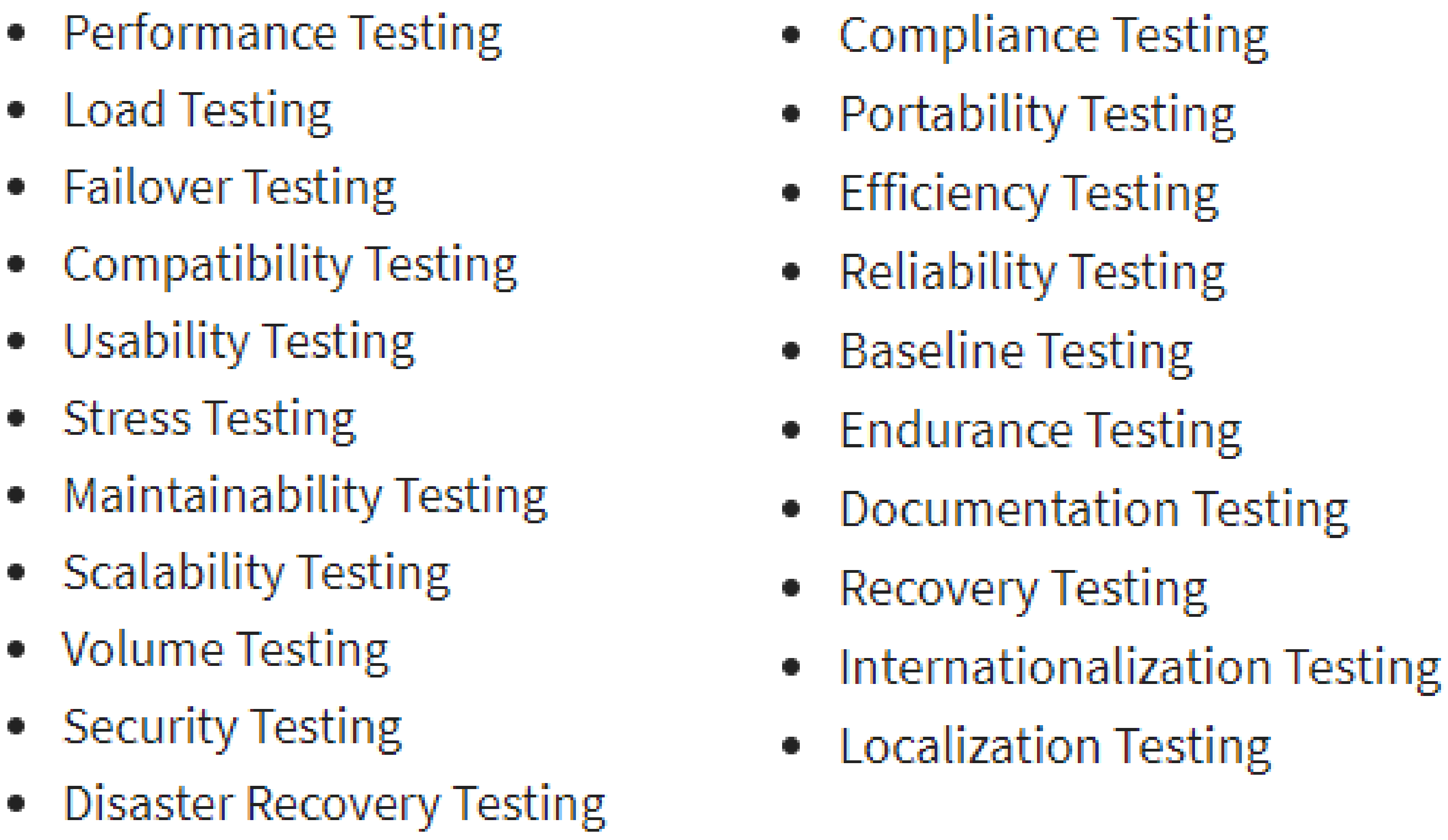


1. Security: This parameter is tested during Security testing. This parameter defines how the system is secure against sudden attacks from internal and external sources.
2. Reliability: This parameter is tested during Reliability testing. This defines the extent to which the system performs its intended functions without failure.
3. Survivability: This parameter is tested during Recovery testing. This parameter checks that the software system is able to recover itself in the case of failure and continuously performs the specified function without any failure.
4. Availability: This is tested during Stability testing. Availability here means the availability percentage of the software system to the original service level agreement. It means the degree to which the user can rely on the software during its operation.

5. Efficiency: This parameter means the extent to which the software system can handle the quantity and response time. This is tested in Performance testing
6. Integrity: This parameter measures how high the source code quality is when it is passed on to the QA.
7. Usability: This is tested in usability testing. This parameter means how easily usable the system is from the user’s perspective.
8. Flexibility: This parameter means how well the system can respond to uncertainty in a way that allows it to function normally.

9. Scalability: This parameter is tested during scalability testing. This parameter measures the degree to which the application can scale up or scale out its processing capacity to meet an increase in demand.
10. Reusability: This means how many existing assets can be reused in some form within the software product development process or in another application.
11. Interoperability: This parameter is tested during the Interoperability testing. This checks that the application interfaces properly with its components or other application or software.
12. Portability: This parameter checks the ease with which the software can be moved from one environment to another.

❖ Non-functional testing types



- Performance Testing
- Load Testing
- Failover Testing
- Compatibility Testing
- Usability Testing
- Stress Testing
- Maintainability Testing
- Scalability Testing
- Volume Testing
- Security Testing
- Disaster Recovery Testing
- Compliance Testing
- Portability Testing
- Efficiency Testing
- Reliability Testing
- Baseline Testing
- Endurance Testing
- Documentation Testing
- Recovery Testing
- Internationalization Testing
- Localization Testing

Benefits of Non-functional Testing

- Improved performance: Non-functional testing checks the performance of the system and determines the performance bottlenecks that can affect the performance.
- Less time-consuming: Non-functional testing is overall less time-consuming than the other testing process.
- Improves user experience: Non-functional testing like Usability testing checks how easily usable and user-friendly the software is for the users. Thus, focus on improving the overall user experience for the application.
- More secure product: As non-functional testing specifically includes security testing that checks the security bottlenecks of the application and how secure is the application against attacks from internal and external sources.

Limitations of Non-functional Testing

- Non-functional tests are performed repeatedly: Whenever there is a change in the application, non-functional testing is performed again. Thus, it is more resource intensive.
- Expensive in case of software update: In case of software update, non-functional testing is performed again thus incurring extra charges to re-examine the software, and thus software becomes expensive.