Non-functional Testing focuses on the quality characteristics of the product (i.e. how well it works). Types of Non-functional testing include:

- **Performance/Load testing** – the degree to which a system or component accomplishes its designated functions within given constraints regarding processing time and throughput rate.
- **Performance/Stress testing** – the degree to which a system or component accomplishes its designated functions beyond its limits.
- **Performance/Reliability testing** – the degree to which a system or component accomplishes its designated functions within given constraints regarding processing time and throughput rate for a specified period of time.
- **Information Security testing** – the ability of the product to prevent unauthorized access, whether accidental or deliberate, to programs or data.
- **Recovery/Operational testing** – the ability of the product to recover from crashes, hardware failures and other similar problems.
- **Usability testing** – the ability of the software to be easily understood, learned and/or used.
- **Portability testing** – the ability of the software to perform in different hardware or software environments.

- **System Integration Testing –**

System Integration Testing is a type of testing carried out on an integrated set of components/systems that interact with one another to verify the behavior of the complete system.

> **Example**: SFM test cases use the SF3 UI to construct and execute SFM cases that in turn run the SFM model. As FDS delivers the support for each SFM input/output file and SFM in turn makes code changes to read/produce the FDS file, the 'configuration' files used to provide the location of files are adjusted to point to FDS so that future testing continues to exercise the FDS interface.
>
> **NOTE**: Integration testing is built into the overall EaR testing strategy. Each component build of EaR that is delivered for testing also becomes a part of another impacted component's testing suite, generally one Sprint/Cycle later. Since each component uses the latest instance of the other components that exist in the environment, the Integration is automatically up to date

- **Reconciliation Testing –**

Reconciliation testing compares the results of the system under test to some reference outputs accepted as accurate to ensure that the new/enhanced system is producing the correct results. This type of testing is particularly effective for some specific situation, like when a system is being re-platformed.

> **Example**: As the FDS delivers each of the SFM inputs, SFM testing will compare the interim dataset created by SFM when reading the original input source file and the interim dataset created by SFM when the FDS dataset is read. The expectation is that they will be identical.

Reconciliation testing, however, is not a substitute for detailed Acceptance Criteria specific 'rule based' testing. Reconciliation testing implicitly tests and can result in false