**Non-Functional Requirements**

**Security**

1. Login requirements - access levels, CRUD levels

2. Password requirements - length, special characters, expiry, recycling policies

3. Inactivity timeouts – durations, actions

**Audit**

1. Audited elements – what business elements will be audited?

2. Audited fields – which data fields will be audited?

3. Audit file characteristics - before image, after image, user and time stamp, etc

**Performance**

1. Response times - application loading, screen open and refresh times, etc

2. Processing times – functions, calculations, imports, exports

3. Query and Reporting times – initial loads and subsequent loads

**Capacity**

1. Throughput – how many transactions per hour does the system need to be able to handle?

2. Storage – how much data does the system need to be able to store?

3. Year-on-year growth requirements

**Availability**

1. Hours of operation – when is it available? Consider weekends, holidays, maintenance times, etc

2. Locations of operation – where should it be available from, what are the connection requirements?

**Reliability**

1. Mean Time Between Failures – What is the acceptable threshold for down-time? e.g. one a year, 4,000 hours

2. Mean Time To Recovery – if broken, how much time is available to get the system back up again?

**Integrity**

1. Fault trapping (I/O) – how to handle electronic interface failures, etc

2. Bad data trapping - data imports, flag-and-continue or stop the import policies, etc

3. Data integrity – referential integrity in database tables and interfaces

4. Image compression and decompression standards

**Recovery**

1. Recovery process – how do recoveries work, what is the process?

2. Recovery time scales – how quickly should a recovery take to perform?

3. Backup frequencies – how often is the transaction data, set-up data, and system (code) backed-up?

4. Backup generations - what are the requirements for restoring to previous instance(s)?

**Compatibility**

1. Compatibility with shared applications – What other systems does it need to talk to?

2. Compatibility with 3rd party applications – What other systems does it have to live with amicably?

3. Compatibility on different operating systems – What does it have to be able to run on?

4. Compatibility on different platforms – What are the hardware platforms it needs to work on?

**Maintainability**

1. Conformance to architecture standards – What are the standards it needs to conform to or have exclusions from?

2. Conformance to design standards – What design standards must be adhered to or exclusions created?

3. Conformance to coding standards – What coding standards must be adhered to or exclusions created?

**Usability**

1. Look and feel standards - screen element density, layout and flow, colours, UI metaphors, keyboard shortcuts

2. Internationalization / localization requirements – languages, spellings, keyboards, paper sizes, etc

**Documentation**

1. Required documentation items and audiences for each item