The following defects are common in analysis:

| **S/N** | **Item** | **Missing** | **Wrong** | **Extra** | **Comments** |
| --- | --- | --- | --- | --- | --- |
| A01 | Business rules (or information) are inadequate or partially missing.  The business rules are specific to FS, BUC, GUC, UCS and SS. They dictate the behavior of:   1. Functional Specifications (FS) or what is expected of the system   In further level of granularity:   1. Business Use Cases (BUC) or how the business is done 2. System Use Cases (UCS) or how the systems supports the business   **Notes:\*** Group Use Cases (GUC) are grouping of the UCS for the purpose of identification of modules and are not a must but rather a convenience.  \*\* The Business Rules of the BUC govern the Business Rules of the UCS and are applied subject to Generalisation and Inheritance principles.  \*\*\* Business Scenarios represent patterns of behavior for the purpose of Basic Flow, Alternate Flows and Exception Flows.  Omission of any of these will make it more difficult in determining what Test Cases (TC) and subsequent Test Suites (TSu) are necessary. It will further adversely impact the Coverage Analysis – Level 2 (Specifications to Test Cases) resulting in reduction of Level-Of-Confidence that ALL requirements have been addressed.  The requirements addressed here are in the categories of:   * Functional Requirements (FR) * External Interface Requirements (EIR) * User Interface Requirements (UIR) * Security Requirements (SECR) |  |  |  |  |
| A02 | Performance criteria (or information) is inadequate or partially missing.  The performance criteria are addressed through Supplementary Specifications (SS) or how the system is expected to behave or a further granularity level (as compared to the BRS). The requirements addressed here are in the categories of:   * Non-Functional Requirements (NFR) * Service Level Agreement Requirements (SLAR) * Legal and Legislative Requirements (LLR) * Design Constraint Requirements (DCR)   **Note:** The SS addresses any requirement that cannot be addressed as a Use Case.  Typically, performance criteria is an SLA requirement and appears here on a further granularity as compared to the BRS. It focuses on:   1. Response Time 2. System Capacity 3. Support to Concurrent Users 4. Identification of Stress breakpoints 5. Growth capability  * Inadequate means that the provided values (or information) cannot be used to measure (in quantifiable means) the attainment of the SLA requirements * Missing or partially missing means an oversight by the System Analyst (SA) or Test Analyst (TA) |  |  |  |  |
| A03 | Ambient environment information is inadequate or missing.  The information pertaining to the surrounding environment, on a further granular level (FS, BUC, GUC, UCS and SS), is insufficient to reach a meaningful conclusion. This has a particularly important meaning when new elements of hardware, firmware and / or software are introduced and may subsequently lead to a time consuming effort delaying, in a meaningful way, the assignment in hand. It further leads to better understanding of the surrounding environment and the constraints imposed.   * Inadequate means that the provided values (or information) cannot be used to measure (in quantifiable means) the attainment of the SLA requirements * Missing or partially missing means an oversight by the System Analyst (SA) or Test Analyst (TA) |  |  |  |  |
| A04 | Modeling NOT done  The purpose of modeling (using BUC and UCS Models) is to gain understanding of the system behavior. Absence of the models increases the probability that the SA / TA did not gain sufficient understanding of the system / product being developed and therefore its associated testing is suspect.  This may further prevent successful enhancement of the system with new features. At present it is the IS professional that keep the ‘Know-How’ and share it via presentations to the testing staff. This should not be the case as the modeling will facilitate better understanding of the system in hand to SA / TA yielding more comprehensive testing. |  |  |  |  |
| A05 | Components are incompatible with other components.  This is a situation that typically arises when conflicting requirements have been introduced. Examples of incompatibility are:   1. Two decimal points n.nn in one requirement vs. four decimal points n.nnnn in another requirements 2. Conflicting business rules – calculation formula in one requirement conflicts with another calculation formula in another requirement 3. Changes to the behavior of the system causing a reduction in resilience, robustness and maintainability   **Note:** This typically happens when one changes the system to do what it was not designed to do. |  |  |  |  |
| A06 | Incomplete requirements.  During the Analysis phase Coverage Analysis – Level 1 (Function to Requirements) takes place. The following is checked and should highlight:   1. Requirements that do not have corresponding detailed requirements in FS, BUC, UCS, SS 2. FS, BUC, UCS, SS that do not have corresponding Requirements (i.e. have been added just because somebody thought it is a good idea. |  |  |  |  |
| A07 | Missing requirements  Requirements that are NOT included in the BRS and should be there in the first place. These are typically identified later in the processes and may be introduced through:   1. Change Request 2. New version of the BRS 3. Missing Requirements and subsequent missing FS, BUC, UCS, SS   **Note:** Another reason for these requirements is late comers (i.e. stakeholders). |  |  |  |  |
| A08 | Un-testable requirements.  Requirements, on a further granular level (FS, BUC, UCS, SS), which attainment cannot be measured through means of:   1. Formula (e.g. x = a\*\*2+b\*\*2) 2. Boolean (e.g. if <Condition> then True else False 3. Logical (e.g. If <Condition> Do <Action> else <Condition> Do <Action>) 4. Truth Table   For example:   1. Maximum and / or Minimum Values (i.e. Less Than, Greater Than, Equal to, Not Equal to, 2. Binary decisions (i.e. Yes or No) 3. Zero or negative values (i.e. do not send an invoice to customer if invoice total value below $1) |  |  |  |  |
| A09 | Incorrect requirements (i.e. Requirements are wrong)  Requirements, on a further granular level (FS, BUC, UCS, SS), that do not make sense in the English language. Typical reasons are:   1. The BA is still waiting to get a final response from the End-User 2. Cut & Paste situation where some of the components were not copied successfully 3. A requirement that requires further elaboration |  |  |  |  |
| A10 | Accuracy specified does not conform to need.  This is typically a requirement, on a further granular level (FS, BUC, UCS, SS), that is expressed in too general terms. Typical examples are:   1. Formula calculations of different accuracy levels (e.g. n.nn \* n.nnnn = n.nnnn) 2. Formulas referencing other subroutines (call to acquired libraries) 3. Differing the requirements until future conditions will be clarified   This defect is typically hard to find at the BRS level hence, this further granular understanding may highlight changes necessary to the BRS. This approach is also known as a suitability check. |  |  |  |  |
| A11 | Erroneous exterior interface definitions.  The exterior interface is typically described using the following terms:   1. Adjoining systems that provide input to the system 2. Adjoining systems that use output generated by the system 3. Adjoining systems that share input / output with the system 4. Adjoining systems that provide an API for communicating 5. Agreed protocols that are shared by adjoining systems   These systems provide strict interface protocols (typically national or international). It is easy to cause oversight mistakes in this respect. When examining it on a further granularity level it is easier to identify. |  |  |  |  |
| A12 | Training of users inadequately considered.  During the introduction of new systems and / or new features to existing systems some end-user training may be warranted. Ignoring this requirement is something that frequently happens from the viewpoints of:   1. Cost 2. Schedule and effort 3. Business necessity   In new systems with a lot of new functionality, a more granular view is necessary (i.e. different end-users may need different training. Another aspect to review includes the interactions between the various end-users. |  |  |  |  |
| A13 | Initialization of the system state not considered.  An operational existing system has typically two states:   1. Regular operation 2. Special operation   This defect refers to the special operation when a system needs to:  2.a Set to an earlier date of operation (for purpose of repeating operations)  2.b Following major migration of the system as a result of upgrade  2.c Be initialised after recovery from a disaster  From the perspective of FS, BUC, UCS these are known as Start and Finish States, Typically a separate UCS is allocated to the Start and the Finish and these are typically combined with Security aspects of the system and subsequently warrant special TC / TSu. |  |  |  |  |
| A14 | Vague specification of the function to be performed.  The specification is not clear for reasons of:   1. Forecast (i.e. definition of the expected results) 2. Limited (i.e. does cater for all permutations / business scenarios) 3. Resilience (i.e. does not consider future potential states) 4. Robustness (i.e. may not survive future hardships)   When reaching a granular level of UCS coupled with SS a Function Point check is recommended to ensure that all functions:   * have been addressed * are defined as expected * Support the predefined test sequence (i..e. scenarios) * Allow for an addition and / or removal of a discrete function |  |  |  |  |
| A15 | User's needs not adequately understood or reflected in the requirements specifications.  This defect is typically identified at a very late SDLC stage (i.e. when the solution is already presented). One of the techniques associated with early detection of this defect is to:   1. Initiate control meetings with the end-user (with a BA other than the one that developed the BRS) 2. Usage of prototyping and modeling tools 3. Have a catalogue of screens and reports 4. A prototype to demonstrate what is required is available   If none of the techniques above was deployed, it is highly likely that IS will be faced with the defect described above. |  |  |  |  |
| A16 | Elements of quality software, such as robustness, transportability, flexibility, maintainability, not specified (note that this item leads to problems in the production live cycle when modification to the original requirements are necessary).  Quality Assurance (QA) is expected to provide input (through the QA Delegate) during the preparation of the BRS. These requirements are typically considered standard to the production environment. These requirements are of the SLA class that include:   1. robustness, 2. transportability, 3. flexibility, 4. supportability 5. resilience 6. maintainability 7. etc.   When reviewing the BRS, the reviewing BA should look for input from other IS sections like:   * Design & Architecture * Development * Testing * Release * QA * Production   In terms of Design Constraints Requirements (DCR), their absence will indicate implementation problems. Any changes from this point onwards will require the use of Change Requests (CR). |  |  |  |  |
| A17 | Requirements are Ambiguous  This defect refers to the possibility that requirements may be interpreted in more than one way. This is primarily associated with the English language that allows more than a singular interpretation to the written word. Another element to consider is the insufficient control / mastering of the English language that is typically subject to the background culture where it is used and the school where is was acquired. Therefore, the reviewing SA / TA should look for potential ambiguity possibilities. It is imperative to verify that the specifications (FR, BUC, GUC, UCS, SS, Business Scenarios) do not suffer from this exact defect. |  |  |  |  |
| A18 | Requirements are unreasonable and non-achievable  It is possible that the stated requirements are outside the technical ability of the IS. The requirements may drive towards use of technology that presents a challenge to the IS personnel but at the same time a risk and exposure to StarHub. The reviewing SA / TA should look for recording of these risks and associated mitigation activities. In terms of BUC, GUC, UCS, SS and Business Scenarios these are expressed as Extends and Includes. It is their further responsibility to highlight these risks if they have not been highlighted before. |  |  |  |  |
| A19 | Requirements are unclear  These kind of requirements will lead to interactions between various IT professionals (e.g. architects, designers, Developers, Testers, etc.) and the end-users. Typically these requirements require further componentization and may lead changes in the sizing (in terms of schedule and budget) of the IS Service Request. In other instances, where external system integrators are involved, these requirements may provide a presumed advantage:   1. On the supplier end – getting more business 2. On the customer end – getting something for nothing   This is an illusion that is going to cost both sides dearly.  At this point in time architectural considerations come into play as well. |  |  |  |  |
| A20 | Specifications are untraceable (to Requirements and to Business Scenario documents)  Traceability is a Level-Of-Confidence “Booster” that will cater for better visibility of the project progress from the executive level viewpoint, Further links (managed in DOORS) include:   1. Backwards Traceability:   1.a From FS to Requirements  1.b From BUC to Requirements  1.c From GUC to BUC  1.d From UCS to GUC  1.e From SS to all BRS  1.f From Business Scenario to UCS   1. Forwards Traceability:   2.a From Requirements to FS (at section level)  2.b From Requirements to BUC  2.c From BUC to GUC  2.d From GUC to UCS  2.e From BRS to SS  2.f From UCS to Business Scenario |  |  |  |  |