ICT Project Guidance

Definition:  
System Requirements Development

Author:

Sky Sigal, Solution Architect

Version:

0.1

## Purpose

This document provides guidance on developing System Requirements suitable for reducing risk related to delivering solutions that incorporate ICT Services.

## Synopsis

System Requirements are contractual in nature, defining obligations, permissions and prohibitions requiring being followed.

## Contents

[Purpose 1](#_Toc156213157)

[Synopsis 1](#_Toc156213158)

[Contents 2](#_Toc156213159)

[Background 3](#_Toc156213160)

[Structure 3](#_Toc156213161)

[Appendices 5](#_Toc156213162)

[Appendix A - Document Information 5](#_Toc156213163)

[Versions 5](#_Toc156213164)

[Images 5](#_Toc156213165)

[Tables 5](#_Toc156213166)

[References 5](#_Toc156213167)

[Review Distribution 5](#_Toc156213168)

[Audience 5](#_Toc156213169)

[Structure 5](#_Toc156213170)

[Diagrams 6](#_Toc156213171)

[Terms 6](#_Toc156213172)

[Appendix B – FAQ 6](#_Toc156213173)

[What is the difference between Requirements versus User Stories? 6](#_Toc156213174)

[Appendix C – System Requirement Attributes 6](#_Toc156213175)

[ID 7](#_Toc156213176)

[[Diagram] Title 7](#_Toc156213177)

[Statement 7](#_Toc156213178)

[Fit/Acceptance Criteria 7](#_Toc156213179)

[Details 7](#_Toc156213180)

[Questions, Response and Analysis 7](#_Toc156213181)

[Metadata 8](#_Toc156213182)

[Appendix D – Requirements of Requirements 8](#_Toc156213183)

## Background

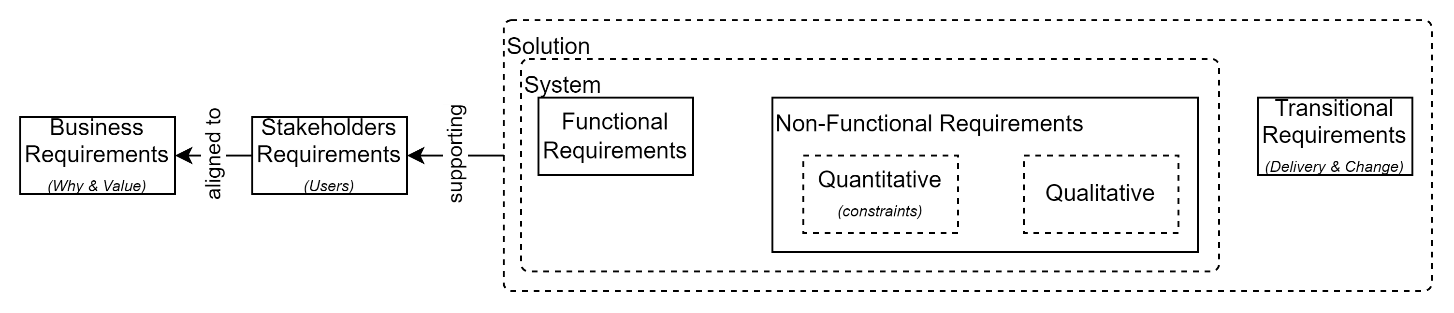


Figure 1: BABOK defined Requirement types

System Requirements are one of the 4 key types of requirements defined in the latest version[[1]](#footnote-2) of *International Institute of Business Analysis* (IIBA)’s *Business Analysis Body of Knowledge* (BABOK).

*Solution Requirements* is the logical set[[2]](#footnote-3) comprised of *System Requirements* and *Transitional Requirements*.

*System Requirements* are the logical set of *System Functional Requirements* and *Non-Functional Requirements*.

*System Functional Requirements* capture the functionality required to meet the expectations of SMEs representing Users defined within *Stakeholder Requirements*.

*System Non-Functional Requirements*[[3]](#footnote-4) capture the quality of service expected of the service used to deliver the functionality.

*Transitional Requirements* are requirements on the process of transitioning from the current state to the future state when the Service is available to end users[[4]](#footnote-5). They are not to be confused with System Non-Functional Requirements.

### Language

The language used within the different requirement types do differ, reflecting the intended audience and purpose of the requirements.

Business Requirement statements are high level statements that an organisation hopes to achieve to meet strategic outcomes. They are often directional, while remaining non-measurable. An example might be “Enhance student learning outcomes by implementing a comprehensive e-learning platform to facilitate remote education.”

Stakeholder Requirements describe the needs and expectations of stakeholders, such as actual system users, and regulatory bodies. They are achievable, attributable, but again, not easily measured. An example might be “Ensure the user interface of the new student management service is intuitive and accessible to teachers, learners, parents and students.”

Functional Requirements describe the system’s functionality and features, often express as capabilities. An example might be: “The system must allow teachers to upload, organise and grade assignments securely within the e-learning platform.” The requirement is expressing which role can do what function. Note that this requirement is a touching on some non-functional qualities as well (“securely”), and the whole thing could and should be broken down to achievable system functionality more abstractly stated before it can be implemented without ambiguity (for example “Permitted Users can manage -- including uploading -- media within the system”.

Transitional Requirements guide the transition from the current state to the desire future state. An example might be “During the implementation phase, provide training sessions for pilot teachers to familiarize them with the new system and collect information to document for more extensive role outs.”

### Schema

The schema, or structure, used to develop and record requirements is more complete than what is required in different contexts.

The same requirements can be exported for use in two different documents, for different audiences. For example, an RFP may go out with two documents: an RFP with only the ID, Statement and an empty Response column, and to evaluators of the response, the same but with the [Response] Analysis column added as well.

### Structure

#### ID

Each requirement has a unique ID for unambiguous reference later from contracts and work items.

It is customary to start Functional Requirements with FR, Non-Functional Requirements with NFR or QR, and Transitional requirements with TR.

Quality Requirements should also capture what Tier and Category they belong to: *NFR-{Tier}-{Quality}-{number}*.

#### [Diagram] Title

Each requirement is provided a short title. The title is intended to be unique, while being both less opaque than an ID and short enough that they could be used within diagram boxes.

#### Statement

A Requirement Statement is expected to be succinct, define its prioritisation, and developed by following SMART guidance.

Good practice dictates that requirements are:

* SMART[[5]](#footnote-6) (**S**pecific, **M**easurable, (pragmatically) **A**chievable, **R**elevant, **T**imely) statements, prioritised as MUST, SHOULD, MUST NOT, SHOULD NOT[[6]](#footnote-7),
* developed in a CLEAR manner (**C**ollaboratively, **L**imited scope, **E**valuated, **A**ppropriate, **R**esource Conscience) manner,

A Statement may use IF statements to define a condition as to when the requirement is applied.

#### Details

Requirement statements are expected to be terse, therefore Implementation and Outcomes may require further description.

#### Acceptance Criteria or Fit

Functional Requirements are clarified using Acceptance Criteria, and are either Accepted or Rejected.

Non-Functional Requirement are measured by Fit Criteria (a collection of Criterion).

Verification & Validation (V&V) is used to qualify requirement statements.

Note:  
Validation is validating that the requirement is appropriate[[7]](#footnote-8), whereas   
Verification is validating that the outcome works as defined.

Acceptance Criteria, sometimes referred to as Fit statements, are used to *verify* the outcome meets quantitative targets, and validate the methods to :

* **Inspection**, using basic senses to observe the outcome (excluding operations).
* **Analysis**, using
* **Demonstration**, operation of the system to demonstrate process.
* **Test,** controlled repeatable form of demonstration.

The result is either Success -- or an Issue, Error, Fault, or Failure.

#### Prompts, Response and Analysis

RFx Respondents should review the Requirements Statement, Fit/Acceptance Criteria, Details and any guiding Questions, to succinctly respond in the Response column as to how their proposed solution meets the intended outcomes.

Their response is analysed, and notes are recorded as to whether the proposed solution sufficiently aligns with the requirements statement and its Fit/Acceptance Criteria.

#### Traceability

***NOT*** recommended is the inclusion of references to identifiers of statements in other documents, as can happen when developing based upon artefacts developed by other organisations.

The reason is the contractual nature of this document’s statements. A reference to external statements may be interpreted as inclusion, potentially adding unexpected and undesired modifiers (e.g., disclaimers, conditions, etc.).

Appendices

Appendix A - Document Information

### Versions

0.1 Initial Draft

### Images

[Figure 1: BABOK defined Requirement types 3](#_Toc156396851)

### Tables

[Table 2: Requirements of Requirements 9](#_Toc156396853)

### References

* *ICT Project Guidance - Definition - System Quality based Non-Functional Requirements Development*
* *ICT Project Guidance – Definition – System Quality Transitional Requirements Development*

**There are no sources in the current document.**

### Review Distribution

The document was distributed for review as below:

|  |  |
| --- | --- |
| Identity | Notes |
| Sandy Britain, Enterprise Architect |  |
| Amy Orr, Data Domain Architect |  |
| Russell Campbell, Project Manager |  |
| Rodney Snear, Project Lead |  |

### Audience

The document is technical in nature, but parts are expected to be read and/or validated by a non-technical audience.

### Structure

Where possible, the document structure is guided by either ISO-\* standards or best practice.

### Diagrams

Diagrams are developed for a wide audience. Unless specifically for a technical audience, where the use of industry standard diagram types (ArchiMate, UML, C4), is appropriate, diagrams are developed as simple “box & line” monochrome diagrams.

### Terms

Refer to the project’s Glossary.

##### IT

: acronym for Information, using Technology to automate and facilitate its management.

##### ICT

: acronym for Information & Communication Technology, the domain of defining Information elements and using technology to automate their communication between entities. IT is a subset of ICT.

Appendix B – FAQ

### What is the difference between Requirements versus User Stories?

Each has a different specific purpose: Requirements are Contractual artefacts - User Stories are work item management artefacts.

Requirements are used to outline Obligations, Recommendations and Prohibitions *prior* to contracts being agreed upon and work commencing, and User Stories are developed after agreement has been reached, to track work that aligns with them.

If Requirements are complete, the User Stories define How the Requirements are to be delivered. If Requirements are not complete, User Stories are used to develop How the delivery of new outcomes are to be delivered.

Both have a distinct format. For example, whereas a User Story often follows a "As a  [StakeholderType], I Desire [Outcome] for [Benefit|Value]" format, and MUST have multiple specific Acceptance Tests, the structural parts of a Requirement are outlined in another Appendix.

Note:  
User Stories that are not traceable to Requirements are valuable during Postmortems to add requirements to default requirements to improve delivery of subsequent projects.

Each requirement has a unique ID for unambiguous reference later from contracts and work items.

[Diagram] Title

Each requirement is provided a short title. The title is intended to be unique, while being both less opaque than an ID, and short enough that they could be used within diagrams.

Statement

A Requirement Statement is expected to be succinct, define its prioritisation, and developed by following SMART guidance.

Good practice dictates that requirements are:

* SMART[[8]](#footnote-9) (**S**pecific, **M**easurable, (pragmatically) **A**chievable, **R**elevant, **T**imely) statements, prioritised as MUST, SHOULD, MUST NOT, SHOULD NOT[[9]](#footnote-10),
* developed in a CLEAR manner (**C**ollaboratively, **L**imited scope, **E**valuated, **A**ppropriate, **R**esource Conscience) manner,

Acceptance Criteria

Appendix C - Common Issues & Mitigations

#### Numbering

It is common practice to use an incrementing number to identify requirements (REQ-01, REQ-02, etc.). We recommend instead to avoid the risk and common issue of duplicate identifiers when cut and pasting, the use of more descriptive Identifiers (QR-SEC-CON-01), using shorter sequences of localised numbers (e.g.: QR-SEC-CON-01 to QR-SEC-CON-05, then QR-SEC-ASS-01 to QR-SEC-CON-05, etc.).

Appendix C – Requirements of Requirements

It seems only fitting to use Requirements to define Requirements for Requirements.

Table 2: Requirements of Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | # | ID | Statement | Rationale & Details |
|  | RR-01 | Default/ Requirements/ When | MUST NOT develop SoWs or other Contracts that are not bound to Requirements. | It is irrational to develop agreements without constraints defined. |
|  | RR-02 | Default/ Requirements/ UserStories | MUST NOT use User Stories as the basis for Contracts. | Requirements are Contractual obligations, defined and agreed upon before Contracts and Agreements are made, and work commences.  User Stories are work item definition and prioritisation artefacts, developed after contracts are signed and work is in process. |
|  | RR-03 | Default/ Requirements/ Wording/ SHOULD | SHOULD avoid the use of SHOULD in contractual requirements - they lead to ambiguity as to whether they will be delivered. | MUST is an obligation. SHOULD is only a recommendation with no contractual obligation.  Instead of using SHOULD, let the RFx Respondent respond they will not meet the requirement, propose an alternate approach that they agree to implement. |
|  | RR-04 | Default/ Requirements/ Wording/ SHALL | MUST NOT use "SHALL" - use the imperative MUST. | Reduce litigation: See: <https://www.plainlanguage.gov/guidelines/conversational/shall-and-must/> |
|  | RR-05 | Default/ Requirements/ Wording/ WANT | MUST NOT use "WANT/DESIRE" for the same reasons. | It's one of the reasons why User Stories are inappropriate for the use in Contract. |
|  | RR-06 | Default/ Requirements/ Outcomes | SHOULD Describe the Outcome desired and the level to which it is to be delivered, while avoiding describing How it is achieved (to in turn permit the maximum design and cost flexibility by implementors). | The outcome is increased value and/or reduction of risk.  The output is an actual artefact or product, which constrains how RFx respondents can implement the solution. |
|  | RR-07 | Default/ Requirements/ Criteria | Whereas projects should expect to develop full Business Requirements, they SHOULD expect to only have to add Criteria metrics (Throughput, Storage amount, etc.) to Quality Requirements. | If requirements are developed intelligently, the Quality Requirements will use statements that refer back to a short list of Quantities that is reviewed and/or updated per project.  Don’t embed Quantities into Statements as they are easy to miss when preparing the Requirements. |
|  | RR-08 | Default/ Requirements/ Eliciting/ Stakeholders | When developing Business and User Requirements, ensure the SMEs of all Stakeholder groups are engaged - not just Business Service Providers.  Consider finding distinct SMEs for the list below -- Service Consumers, Service Providers, Cultural Representative Specialists, Support Specialists, Operations Specialists, Maintenance Specialists, Assurance (Security, Privacy, Performance, Quality) Specialists. | Stakeholder Requirements are not fit for purpose if they only capture Business Service Provider SME’s perspective, acting as a Service Consumer SME.  It is important to get SMEs for all groups of users who will actually be using the service. |
|  | RR-09 | Default/ Requirements/ Postmortems | SHOULD include a requirement to perform a Post-Mortem to improve the Enterprise Requirements. | Start from the beginning, looking for UserStories that had to be added due to missing Requirements. |
|  | RR-10 | Default/ Requirements/ Wording/ All | To avoid ambiguity MUST not use the word "all". | "Deliver all documentation" will mean different things to different stakeholders. |
|  | RR-11 | Default/ Requirements/ Wording/ IF | You MAY use Conditional Statements (e.g.: "IF hosted as an Service (SaaS)...") to permit general Requirements being excluded/included under certain conditions. | Very useful to exclude the requirement if not required.  Examples: - “IF developing custom systems,…” - “IF hosted and managed by the organisation,…” |
|  | RR-12 | Default/ Requirements/ Improvements | Requirements MUST be reviewed regularly to look for whether outcomes are still achievable, and not introducing technical How statements. | Continuously improve requirements. |

1. V3.0, first released in 2015, iteratively updated since. [↑](#footnote-ref-2)
2. A logical set is term for addressing a number of artefacts without being an artefact or document in itself. [↑](#footnote-ref-3)
3. See: *ICT Project Guidance - Definition - System Quality based Non-Functional Requirements Development* [↑](#footnote-ref-4)
4. See *ICT Project Guidance – Definition – System Quality Transitional Requirements Development* [↑](#footnote-ref-5)
5. BABOK proposes that requirements have: “Atomicity, Completeness, Consistency, Conciseness, Feasibility, Unambiguity, Testability, Prioritization and Understandability” [↑](#footnote-ref-6)
6. [↑](#footnote-ref-7)
7. Ie, it is equivalent to ISO-25010/Functionality/Appropriateness. [↑](#footnote-ref-8)
8. BABOK proposes that requirements have: “Atomicity, Completeness, Consistency, Conciseness, Feasibility, Unambiguity, Testability, Prioritization and Understandability” [↑](#footnote-ref-9)
9. [↑](#footnote-ref-10)