DRAFT - IT Project Guidance

Requirement Types and their Development

Version:

0.1

## Purpose

This document provides guidance on defining and structuring requirements for government IT initiatives. It is designed to support both new and experienced Business Analysts in delivering actionable, well-structured, and complete requirement sets that enable effective procurement, stakeholder confidence, and project delivery.

## Synopsis

This document builds upon the Business Analysis Body of Knowledge (BABOK) and introduces a structured requirement model: BUQCS[F&NFR]TUT. It reorders and expands the commonly known BUST framework into a more detailed and useful model suitable for New Zealand public sector contexts. The guidance applies equally to those designing RFPs as it does to vendors proposing solutions, ensuring common structure, traceability, and verifiability of needs. It is intended to bridge the gap between senior analysts seeking authority and structure, and newer analysts needing clarity and templates.

## Contents

[Purpose 1](#_Toc195698126)

[Synopsis 1](#_Toc195698127)

[Contents 2](#_Toc195698128)

[Introduction 4](#_Toc195698129)

[Background 4](#_Toc195698130)

[Stakeholder Mapping 4](#_Toc195698131)

[Business Requirements 4](#_Toc195698132)

[User Requirements 5](#_Toc195698133)

[Qualities of Service Requirements 5](#_Toc195698134)

[Capability Requirements 5](#_Toc195698135)

[System Requirements 6](#_Toc195698136)

[Functional Requirements 6](#_Toc195698137)

[Non-Functional Requirements (NFRs) 6](#_Toc195698138)

[Transitional Requirements 6](#_Toc195698139)

[User Stories 6](#_Toc195698140)

[Technical Requirements 7](#_Toc195698141)

[Conclusion 7](#_Toc195698142)

[Next Steps 7](#_Toc195698143)

[Appendices 8](#_Toc195698144)

[Appendix A - Document Information 8](#_Toc195698145)

[Versions 8](#_Toc195698146)

[Images 8](#_Toc195698147)

[Tables 8](#_Toc195698148)

[References 8](#_Toc195698149)

[Review Distribution 8](#_Toc195698150)

[Audience 8](#_Toc195698151)

[Structure 8](#_Toc195698152)

[Diagrams 9](#_Toc195698153)

[Acronyms 9](#_Toc195698154)

[API 9](#_Toc195698155)

[GUI 9](#_Toc195698156)

[ICT 9](#_Toc195698157)

[IT 9](#_Toc195698158)

[UI 9](#_Toc195698159)

[Terms 9](#_Toc195698160)

[Application Programming Interface 9](#_Toc195698161)

[User 9](#_Toc195698162)

[User Interface 9](#_Toc195698163)

[Appendix B – Example Stakeholder Map 9](#_Toc195698164)

[Appendix C – Example Business Requirements 10](#_Toc195698165)

[Appendix D: Example User Requirements 10](#_Toc195698166)

[Appendix E: Example Qualities 11](#_Toc195698167)

[Appendix F: Example Capability Requirements 11](#_Toc195698168)

[Appendix G: Example Functional Requirements (FRs) 12](#_Toc195698169)

[Appendix H: Example Non-Functional Requirements (NFRs) 12](#_Toc195698170)

[Appendix I: Example Transitional Requirements 12](#_Toc195698171)

[Appendix J: Example User Stories 13](#_Toc195698172)

[Appendix K: Example Technical Requirements 13](#_Toc195698173)

# Introduction

Projects fail without clear definition. The success of an IT project hinges on precise articulation of why the project is being done, who it serves, what the desired outcomes are, how they will be achieved, and what constraints or dependencies must be acknowledged. BABOK defines Business Analysis as the practice of enabling change in an enterprise by defining needs and recommending solutions that deliver value. This document follows that principle but provides concrete structures and terms to ensure consistent practice and language across projects.

## Background

The IIBA’s BABOK is recognised globally as the primary reference for the work of business analysis. It outlines how analysts should identify stakeholders, find subject matter experts, elicit needs, and define structured artefacts. However, it leaves room for national or organisational conventions in how those artefacts are structured. This document offers such a structure—specifically designed to improve procurement clarity, align with ISO quality standards, and simplify cross-agency collaboration. The BUQCS[F&NFR]TUT model provides the granularity and ordering required to meet these goals.

## Stakeholder Mapping

Mapping Stakeholders are individuals or groups who influence, own, use, or are impacted by the system. A clear stakeholder map is essential before any elicitation occurs. It should distinguish sponsors, business owners, operational leads, system users, service recipients, policy authorities, and integration partners. New analysts should understand that not all stakeholders are visible at project inception. The stakeholder map should evolve as analysis progresses. Appendix C provides patterns for stakeholder roles.

## Business Requirements

Business requirements are the strategic drivers for the project. These are not what the system will do, but why the system must exist. Each must be tied to a clearly articulated opportunity or threat, such as improving service delivery, reducing operating costs, improving compliance, or avoiding risk. Each must include a Definition of Success—a measurable condition that indicates when the business outcome has been achieved. For example: “Reduce manual processing of enrolments by 75% within 12 months of go-live, without increasing error rates.” “Enable compliance with the new Education Records Privacy Standard before Q3 2026.”

## User Requirements

User requirements specify the needs of specific roles within the stakeholder group. These are work-related goals expressed in the user's language, not technical terms. Each should identify the role, the task, and the outcome, along with a clear success condition. This may include access to information, the ability to take actions, or the visibility of outcomes. These requirements are often elicited through interviews, journey mapping, and scenarios. They should be free of implementation assumptions. For example: “A school office manager needs to view and update a student’s address and emergency contact details within two screens, without needing to switch applications.” “An authorised Ministry analyst must be able to query anonymised attendance data by region, school type, and age group.”

## Qualities of Service Requirements

Qualities define how well the system must perform. They should align to international standards (e.g. ISO/IEC 25010, 25012, 25022). These cover usability, reliability, performance, security, maintainability, and portability. Requirements should be structured into four levels:

1. Industry standard baseline (e.g. NZ government or education sector)
2. Agency or organisation-specific needs
3. Programme or initiative-specific conditions
4. Quantitative project expectations (e.g. RTO, user count, bandwidth, expected load growth) All levels should be expressed quantitatively where possible. Missing values should default to industry standard baselines.

## Capability Requirements

Capabilities are the organisational-level functions the system must support. They describe what the organisation must be able to do, such as manage enrolments, issue credentials, or allocate resources. Capabilities are based on the domains in which the system operates—System, Social, Aspirational, Work, Organisational. This domain-led framing ensures traceability across government programmes. Capabilities allow reuse across projects. It is the vendor’s role to define the Functions and features that fulfil them.

## System Requirements

System Requirements act as the logical grouping of both Functional Requirements (FRs) and Non-Functional Requirements (NFRs). This section provides the structured framework that allows FRs and NFRs to be understood and validated as a whole. System Requirements represent the expected behaviour and qualities of the system under defined conditions. FRs define what the system does; NFRs define how well it must do them.

### Functional Requirements

Functional Requirements (FRs) are expectations that describe observable behaviours, processing logic, workflows, inputs, and outputs. They must be testable, unambiguous, and traceable to User Requirements or Capabilities. Example: "The system must prevent duplicate enrolments by checking NEN+DOB against all active learner profiles." "The service must expose a read-only endpoint that returns the full Education Record history for a given StudentID."

### Non-Functional Requirements (NFRs)

NFRs implement the qualities in the form of system-level conditions. These include maximum downtime, latency under load, response to error conditions, failover time, or maintainability. All NFRs should correspond to previously stated Qualities. If a vendor or team cannot meet a given quality, the gap must be acknowledged and, if accepted, an alternative mitigation strategy proposed. All NFRs must be subject to Acceptance Criteria and, where possible, verified through operational testing.

## Transitional Requirements

These describe what must exist at deployment time to make the solution operational. They include delivery of documentation, user training, workstation configuration, environment provisioning, licence acquisition, DNS updates, firewall reconfiguration, user support readiness, and data migration. They are not part of the product but are essential for the product to be usable. Transitional Requirements are often neglected but are one of the most common failure points in projects.

## User Stories

User Stories are a way of expressing emerging or unfinalised requirements during delivery. They follow the format: As a [role], I want [function], so that [value]. Each must include Acceptance Tests which verify that the function is complete. For example: "As a timetabling coordinator, I want to import a set of term dates from a spreadsheet, so that I don’t need to rekey the information for each school." Acceptance Tests should verify error handling, file format tolerance, and audit logging.

## Technical Requirements

These are the constraints and conditions imposed by the delivery environment. They may include supported browsers, deployment models, security tools, coding standards, monitoring obligations, or preferred frameworks. While these may be defined by the technical leads or infrastructure team, they must be captured alongside the broader requirement set to ensure they are included in scope, scheduling, and cost estimation.

## Conclusion

Requirements must be structured to serve their purpose: to guide analysis, inform design, validate delivery, and measure success. The BUQCS[F&NFR]TUT model provides a repeatable, extendable, and procurement-friendly format that enables clarity, reuse, and accountability. By framing requirements around purpose, outcomes, standards, and system context, public sector analysts can produce more effective, defensible, and enduring solutions.

## Next Steps

Further appendices will provide examples of requirements at each level, suggested formats, and checklists for analysts.

Appendices

Appendix A - Document Information

Authors & Collaborators

* Sky Sigal, Solution Architect

### Versions

* 1. Initial Draft

### Images

[Figure 1: TODO Image 2](#_Toc144995112)

### Tables

[Table 1: TODO Table 3](#_Toc145048484)

[Table 2: TODO Table 2 3](#_Toc145048485)

### References

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

### Review Distribution

The document was distributed for review as below:

|  |  |
| --- | --- |
| Identity | Notes |
|  |  |
|  |  |
|  |  |

### 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.

### Acronyms

API

: [Application Programming Interface](#Term_ApplicationProgrammingInterface).

GUI

: [Graphical User Interface](#Term_ApplicationProgrammingInterface). A form of [UI](#Acronym_UI).

ICT

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

IT

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

UI

: User Interface. Contrast with [API](#Acronym_API).

### Terms

Refer to the project’s Glossary.

Application Programming Interface

: an Interface provided for other systems to invoke (as opposed to User Interfaces).

User

: a human user of a system via its UIs.

User Interface

: a system interface intended for use by system users. Most computer system UIs are Graphics User Interfaces ([GUI](#Acronym_GUI)) or Text/Console User Interfaces (TUI).

Appendix B – Example Stakeholder Map

A stakeholder map identifies roles, responsibilities, and influences across a project. For clarity, stakeholders are grouped by function and relation to the system. Use this structure to initiate stakeholder identification at project outset:

* Sponsor: Authorises and funds the initiative. Sets strategic direction.
* Business Owner: Accountable for business outcomes. May be a Directorate or Group Manager.
* Subject Matter Experts (SMEs): Deep knowledge of current processes or legal obligations.
* Operational Owner: Responsible for day-to-day operation of the solution post-delivery.
* End Users: Interact directly with the system to perform tasks.
* External Users: Recipients of service output or participants in workflows (e.g. parents, other agencies).
* System Owner: Oversees integration, maintenance, support, and hosting.
* Support Staff: Provide helpdesk, admin, and operational support.
* Regulatory Authorities: Impose legal or compliance obligations.
* Integration Partners: Systems or teams responsible for APIs, shared data, or workflows.

Appendix C – Example Business Requirements

Business Requirements should always answer why the project exists. These statements:

* Must relate to a strategic threat or opportunity
* Must include a definition of success
* Should be agnostic of solution or implementation

Format: Each statement should follow: [Business Driver] + [Organisational Goal] + [Definition of Success]

Example:

1. Due to rising administrative costs in enrolments, the Ministry seeks to reduce manual enrolment processing by 75% within two years, while maintaining existing data accuracy.
2. A new regulatory standard requires all learning assessments to be traceable to national curriculum codes. The system must allow tagging and retrieval of assessments by curriculum strand and level.
3. The project aims to reduce risk exposure from manual record handling by enabling secure digital record management across all providers.

Appendix D: Example User Requirements

User Requirements define what specific users need to achieve in their roles. These requirements:

* Should use language familiar to the user
* Must describe tasks, data access, or workflow expectations
* Must include a definition of success or completion
* Should avoid prescribing how the need is met

Format: [Role] + [Need or Task] + [Context] + [Definition of Success]

Example:

1. A regional administrator needs to submit aggregated attendance data monthly to the Ministry, using a validated export format.
2. A school office manager requires visibility of all students flagged as vulnerable, including last contact date, within a single dashboard.
3. A parent must be able to update their child’s medical information and receive confirmation that the school has acknowledged it.

Appendix E: Example Qualities

(ISO-Aligned) Qualities describe how well the system must perform, operate, and endure under expected conditions. They must:

* Be traceable to ISO 25010/12/22 categories
* Be measurable
* Be stated independently of the technology used

Structure: Category + Metric + Threshold + Justification (if applicable)

Example:

1. Availability: The system must provide 99.95% uptime during business hours (7:30 AM – 6:00 PM NZST), excluding scheduled maintenance.
2. Performance: The system must return search results within 2 seconds for 95% of queries under a load of 2,000 concurrent users.
3. Security: Role-based access must be enforced for all data access. All login activity must be logged and stored for a minimum of 12 months.

Appendix F: Example Capability Requirements

Capability Requirements define the business actions the system must support. These are domain-level and often reused across programmes:

* Must describe what the organisation needs to do
* Should not assume how it will be done
* Should be stable across time, systems, and delivery models

Format: [Capability Domain] + [Action] + [Purpose or Impact]

Example:

1. Social Domain – Maintain Learner Profiles: The organisation must be able to create, update, view, and archive individual learner profiles over time.
2. Organisation Domain – Assign Staff Roles: The system must support the assignment of staff to multiple schools and roles, with effective dates.
3. System Domain – Manage Access Credentials: The organisation must issue, revoke, and audit access credentials across all user types.

Appendix G: Example Functional Requirements (FRs)

Functional Requirements describe what the system must do in response to inputs, workflows, or data events.

* Must be observable
* Must be verifiable by test
* Must support user or capability needs

Format: [Trigger/Event] + [Expected Behaviour] + [Constraints or Rules]

Example:

1. When a student’s address is updated, the system must validate the new address using the NZ Post API before saving.
2. The calendar import feature must detect duplicate entries and prompt for resolution before applying changes.
3. The system must send a notification to guardians whenever a learner is marked absent for three consecutive days.

Appendix H: Example Non-Functional Requirements (NFRs)

NFRs implement system-wide constraints and performance targets:

* Must correspond to stated Qualities
* Must be measurable
* Must include monitoring or verification method

Example:

1. The system must respond to all data retrieval requests in under 3 seconds for 95% of operations.
2. Error recovery must complete within 60 seconds of a failed transaction in normal conditions.
3. All audit logs must be retained and queryable for at least 7 years.

Appendix I: Example Transitional Requirements

These enable successful go-live and use of the system:

* Not part of product functionality
* Must be resourced and costed
* Often require collaboration with multiple parties

Example:

1. All users must be trained on the new interface and administrative workflows at least two weeks prior to go-live.
2. Legacy attendance data must be migrated into the new system with a 98% record integrity threshold.
3. Network firewall rules must allow bidirectional traffic between the platform and external cloud service X.

Appendix J: Example User Stories

User Stories guide agile delivery and capture in-progress or evolving needs:

* Must follow: As a [role], I want [action], so that [value]
* Must include acceptance tests or validation steps

Example:

1. As a teacher, I want to mark students present from my tablet, so that I can complete attendance during outdoor activities.
2. As a helpdesk agent, I want to resend activation emails, so that users can complete login setup.
3. As a principal, I want to generate weekly absenteeism reports by year level, so that I can detect emerging trends.

Appendix K: Example Technical Requirements

These define constraints or preferences imposed by the delivery environment or policy:

* Must be relevant to hosting, tooling, or integration
* May affect timeline or cost

Example:

1. The system must support Chrome, Edge, and Safari browsers at latest minus one version.
2. Hosting must occur within a New Zealand-based, IRAP-assessed data centre.
3. CI/CD pipelines must include automated static code analysis and vulnerability scanning.