DRAFT - IT Project Guidance

System Quality Requirements –   
Custom Development

Version:

0.1

## Description

This glossary contributes to the organisation’s internal Body of Knowledge (BOK) by providing working definitions of terms, acronyms, and key phrases used within a specific domain or aspect of IT. Its intent is to reduce misinterpretation, support shared understanding, and assist in the consistent framing of discovery, definition, and design activities. While terms may evolve over time or vary across contexts, this glossary serves as a common reference to improve clarity and alignment within and across projects.

## Synopsis

Requirements are developed to meet ISO-25010 defined categories for Quality Requirements, organised in a tiered, pyramidal manner to diminish the effort of RFx respondents to determine project specifics versus default quality expectations.

## Contents

[Description 1](#_Toc149732228)

[Synopsis 1](#_Toc149732229)

[Contents 2](#_Toc149732230)

[Introduction 6](#_Toc149732231)

[Purpose 6](#_Toc149732232)

[Context 6](#_Toc149732233)

[Structure 6](#_Toc149732234)

[Tiers 6](#_Toc149732235)

[Grouping By Qualities 7](#_Toc149732236)

[Qualities 7](#_Toc149732237)

[Content 8](#_Toc149732238)

[Prioritisation 8](#_Toc149732239)

[Exemptions 8](#_Toc149732240)

[Acceptance Criteria 8](#_Toc149732241)

[Quantitative Settings 9](#_Toc149732242)

[Quality Requirements 10](#_Toc149732243)

[ISO Defined Qualities 10](#_Toc149732244)

[Terminology Used 10](#_Toc149732245)

[Prioritisation 10](#_Toc149732246)

[Headings 10](#_Toc149732247)

[Project Specific System Qualities 11](#_Toc149732248)

[Scope 11](#_Toc149732249)

[Custom Developed Solution Qualities 13](#_Toc149732250)

[Scope 13](#_Toc149732251)

[Security 14](#_Toc149732252)

[Functionality 23](#_Toc149732253)

[Reliability 26](#_Toc149732254)

[Usability 32](#_Toc149732255)

[Performance 35](#_Toc149732256)

[Maintainability 36](#_Toc149732257)

[Portability 39](#_Toc149732258)

[Installed Solution Qualities 41](#_Toc149732259)

[Scope 41](#_Toc149732260)

[Security 42](#_Toc149732261)

[Reliability 45](#_Toc149732262)

[Maintainability 46](#_Toc149732263)

[Ministry Specific System Qualities 47](#_Toc149732264)

[Scope 47](#_Toc149732265)

[Processes 48](#_Toc149732266)

[Security 49](#_Toc149732267)

[Compatibility 51](#_Toc149732268)

[Usability 52](#_Toc149732269)

[NZ Education Sector System Qualities 54](#_Toc149732270)

[Scope 54](#_Toc149732271)

[Processes 55](#_Toc149732272)

[Security 56](#_Toc149732273)

[Compatibility 56](#_Toc149732274)

[Maintainability 56](#_Toc149732275)

[NZ Government Sector System Qualities 58](#_Toc149732276)

[Scope 58](#_Toc149732277)

[Processes 59](#_Toc149732278)

[Security 59](#_Toc149732279)

[Functionality 65](#_Toc149732280)

[Compatibility 66](#_Toc149732281)

[Usability 66](#_Toc149732282)

[Maintainability 67](#_Toc149732283)

[Portability 68](#_Toc149732284)

[NZ specific Default System Qualities 69](#_Toc149732285)

[Scope 69](#_Toc149732286)

[Privacy 70](#_Toc149732287)

[Default System Qualities 71](#_Toc149732288)

[Scope 71](#_Toc149732289)

[Security 72](#_Toc149732290)

[General 72](#_Toc149732291)

[Confidentiality 75](#_Toc149732292)

[Privacy 85](#_Toc149732293)

[General 85](#_Toc149732294)

[Functionality 87](#_Toc149732295)

[Completeness 87](#_Toc149732296)

[Compatibility 89](#_Toc149732297)

[Interoperability 90](#_Toc149732298)

[Usability 93](#_Toc149732299)

[Operability 93](#_Toc149732300)

[User Protection Error 96](#_Toc149732301)

[User Interface Aesthetics 96](#_Toc149732302)

[Accessibility 96](#_Toc149732303)

[TODO 98](#_Toc149732304)

[Reliability 109](#_Toc149732305)

[Performance 118](#_Toc149732306)

[Maintainability 124](#_Toc149732307)

[Portability 138](#_Toc149732308)

[Default Internal Requirements 144](#_Toc149732309)

[Conclusion 146](#_Toc149732310)

[Appendices 147](#_Toc149732311)

[Appendix A - Document Information 147](#_Toc149732312)

[Images 147](#_Toc149732313)

[Tables 147](#_Toc149732314)

[References 147](#_Toc149732315)

[Review Distribution 147](#_Toc149732316)

[Audience 148](#_Toc149732317)

[Structure 148](#_Toc149732318)

[Diagrams 148](#_Toc149732319)

[Terms 148](#_Toc149732320)

[Appendices B – ISO-25010 Qualities 149](#_Toc149732321)

[Delivery 149](#_Toc149732322)

[Security 149](#_Toc149732323)

[Confidentiality 149](#_Toc149732324)

[Integrity 149](#_Toc149732325)

[Non-Repudiation (of Action) and Accountability (by Whom) 149](#_Toc149732326)

[Authenticity 149](#_Toc149732327)

[Availability 149](#_Toc149732328)

[Privacy 150](#_Toc149732329)

[Functionality 150](#_Toc149732330)

[Functional Completeness 150](#_Toc149732331)

[Functional Correctness 150](#_Toc149732332)

[Functional Appropriateness 150](#_Toc149732333)

[Performance 150](#_Toc149732334)

[Time behaviour 150](#_Toc149732335)

[Resource Utilisation 150](#_Toc149732336)

[Capacity 150](#_Toc149732337)

[Compatibility 150](#_Toc149732338)

[Co-existence 150](#_Toc149732339)

[Interoperability 150](#_Toc149732340)

[Usability 150](#_Toc149732341)

[Appropriateness recognisability 151](#_Toc149732342)

[Learnability 151](#_Toc149732343)

[Operability 151](#_Toc149732344)

[User Error Protection 151](#_Toc149732345)

[User Interface Aesthetics 151](#_Toc149732346)

[Accessibility 151](#_Toc149732347)

[Reliability 151](#_Toc149732348)

[Maturity 152](#_Toc149732349)

[Availability 152](#_Toc149732350)

[Fault tolerance 152](#_Toc149732351)

[Recoverability 152](#_Toc149732352)

[Maintainability 152](#_Toc149732353)

[Modularity 152](#_Toc149732354)

[Reusability 152](#_Toc149732355)

[Analysability 152](#_Toc149732356)

[Modifiability 152](#_Toc149732357)

[Testability 152](#_Toc149732358)

[Portability 152](#_Toc149732359)

[Adaptability 152](#_Toc149732360)

[Installability 152](#_Toc149732361)

[Replaceability 152](#_Toc149732362)

## Introduction

BOSSCARD/ RAID: Background [], Objective, Options, Scope[In/Out], Stakeholders [Users], Constraints, Assumptions, Risks, Dependencies, Decisions, Deliverables.

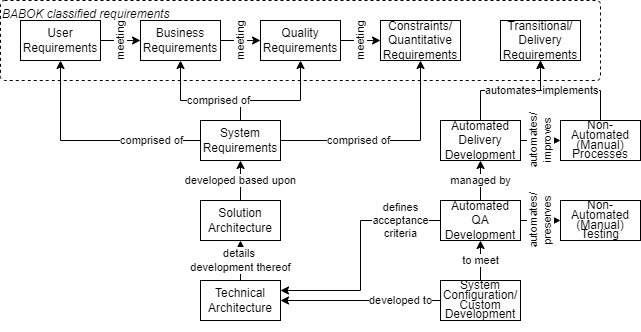
## Purpose

This document summarises the project’s key Qualitative Quality Requirements.

## Context

This set of requirements fits within the Business Analysis Body of Knowledge (BABOK) requirements classification system (Business, User, Functional, Quality, Constraints and Transition requirements.

The totality of the Functional Requirements, Quality Requirements and Quantitative Requirements are referred to as the “System’s Requirements”, the basis of the Solution Architecture Description document.



## Structure

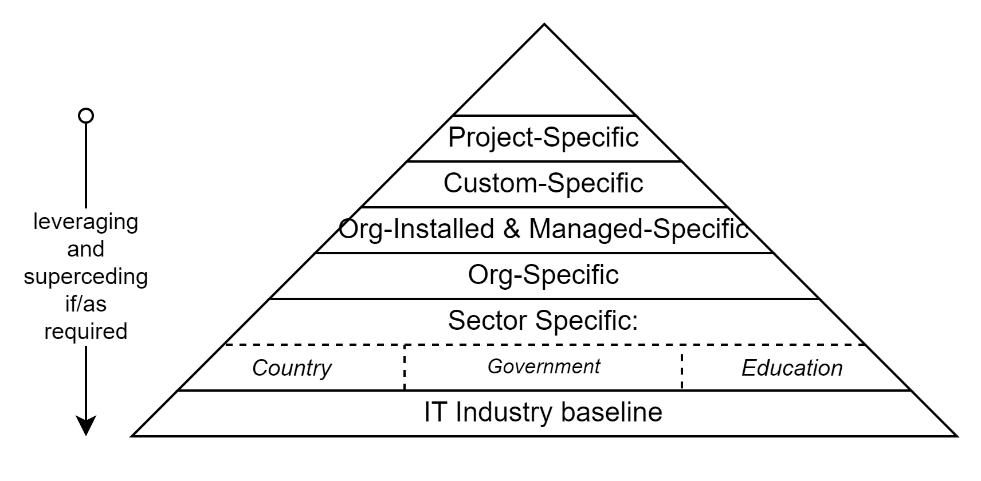
To minimise miscomprehensions as to expectations, the document’s requirements are organised and presented in a specific order, as described below.

### Tiers

Requirements are organised according to layered Scopes -- each section expanding or superseding one or more requirements in a more general scope.

The requirements start with the smallest set of requirements, the Project-Specific, that build on top of more general Organisation requirements, which in turn are built upon Sectors-Specific requirements.

At the bottom are Baseline/Default industry standard requirements.



**Scoping as above is purposeful:** the delivery of projects is de-risked by avoiding developing requirements to define novel approaches to previously solved IT solutions, while being mindful of choosing ones that do not negatively impact the delivery of uniquely valuable business service features.

## Grouping By Qualities

Within each tier, Requirements are organised according to ISO standards based headers where applicable:



### Qualities

The ISO-\* quality model is the cornerstone of a Service quality evaluation system.

The quality model determines which quality characteristics will be considered when evaluating the properties of a software service.

The quality of a system is the degree to which the system satisfies the stated and implied needs of its various stakeholders, and thus provides value. Those stakeholders' needs (functionality, performance, security, maintainability, etc.) are precisely what is represented in the quality model.

## Content

Good practice dictates that requirements are SMART (Specific, Measurable, (pragmatically) Achievable, Relevant, Timely), developed in a CLEAR manner (Collaboratively, Limited-scope, Evaluated, Appropriate, Resource Conscience) manner. Refer to the Appendices.

### Prioritisation

Requirement prioritisation is according to RFC-8174 defined terms. Their implication in the context of procurement is as below:

1. **MUST** - requirements expected to be deliver as defined.   
   If technically *unable* to (e.g.: because the service is a **SaaS**), propose a work-around.
2. **SHOULD** - requirements expected to be deliver as defined, but alternative means to obtaining an equivalent outcome are acceptable.
3. **MAY** - represents requirements that -- time and resources permitting – would be ‘nice to have’ but are not considered essential functionality for the final solution.

### Exemptions

Not all solutions can reasonably meet all requirements that are marked as MUSTs.   
For example, proposed solutions may rely on the use of 3rd party SaaS service providers -- whom it is improbable will agree to modifying their service to meet project specific requirements. In such cases, it is up to the presenter to submit an endorsable work around that can meet the same obligations and/or objectives.

### Acceptance Criteria

Quantitative requirements are Measurable in their own right, therefore do not include accompanying Fit statements.

# Quantitative Settings

TO FILL IN PER PROJECT – OR DO IT IN A SEPARATE DOCUMENT (I THINK THE SECOND OPTION IS MY PREFERENCE)

# Quality Requirements

The Business Analysts Book of Knowledge (BABOK) defines Quality Requirements as one of the 5 types of categories requiring defining.

## ISO Defined Qualities

ISO-25010 defines a list of system qualities considered valuable.

ISO-25012 defines a list of qualities defined valuable when assessing data within a system.

ISO-25022 defines a list of qualities that users expect to experience when using a system that contains information.

## Terminology Used

Specific terminology is used throughout this document, following guidance given in *IT Project Guidance – Requirement Development*.

### Prioritisation

As per the Appendices, Requirements are Obligations.

The quality requirements outline the required or preferred capabilities of the solution by priority (Must, Should and Could).

1. **Must** - the respondent must propose a solution that, as a minimum standard, meets the requirements of the core functionality identified as “Must”.
2. **Should** - functional requirements that have been identified as “Should” are highly desirable functionality that should be included, or workarounds may be available for this functionality.
3. **Could** - represents requirements that are ‘a nice to have’ (time and resources permitting), but not considered necessary functionality for the final solution.

### Headings

# Custom Developed Solution Qualities

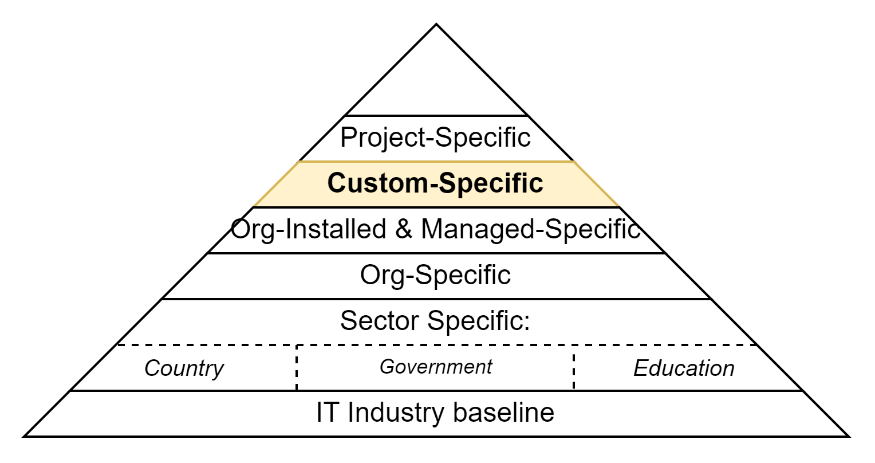


Figure 2: Custom Developed Specific Tier of Quality Requirements

This section lists qualities specific to custom developed components and services – as opposed to licensed, procured, and/or subscribed proprietary services.

**Important:**While most solution proposals will not contain custom developed services, it is highly recommended that proposals are evaluated against what *could* be delivered if custom built. The resulting gap analysis can then be used to evaluate the solution proposals.

## Scope

The following are the Requirements categories defined in this Tier:

|  |  |  |
| --- | --- | --- |
| Defined | Undefined (Inherited) | Details |
| * Security * Usability * Maintainability * Portability | * Privacy Deliverability * Functionality * Reliability * Performance | All systems, whether for government, education or other purposes, are required to abide by national laws and regulations that are applicable to all persons. |

## Security

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | # | ID | Statement | Rationale | Fit Criteria | Response | Comments | Details |  |
|  | o **General** o **Confidentiality** þ **Integrity** o **Non-Repudiation** þ **Accountability** o **Authenticity** | | | |  |  |  |  |  |
|  | **Integrity** | | | |  |  |  |  |  |
|  | QR-CUS-SEC- 00 | Custom/ Security/ Integrity/ Authorisation/ Obligations based | The solution SHOULD present Responsibilities & Obligations to Users for acceptance, before they are assigned a Roles Permissions. | Roles are a combination of accepted Responsibilities (SHOULDs) & Obligations (MUSTs) in return for Permissions (MAYs).  Adding them to Role definitions makes Invitation to Group or Resource Roles by Notification possible to be automated. |  |  |  | The solution allows assigning Responsivities & Obligations to Roles, which are in turn assigned to Users.  The solution allows fine tuning by adding or removing Obligations directly to Users.  *Note: These can be assigned to Users and Roles the same way as Permissions.*  Roles are a combination of accepted Responsibilities (SHOULDs) & Obligations (MUSTs) in return for Permissions (MAYs).  Adding them to Role definitions makes Invitation to Group or Resource Roles by Notification possible to be automated. |  |
|  | QR-CUS-SEC- 00 | Custom/ Security/ Integrity/ Authorisation/ Group specific Roles | The solution MUST be capable of assigning group specific roles. | A user may belong to different groups, in different capacities/roles (Owner in one, Member in another, etc.). |  |  |  | The solution provides the capability of inviting the same users to different groups, in different Roles.  For example, in one group, a user can be an Administrator, and in another, be a Member or Guest.  Consider the following for Group Roles:  - Accountable - Administrators - Approvers - Group Contact  Person - Members - Collaborators, - Guests observers, |  |
|  | QR-CUS-SEC- 00 | Custom/ Security/ Integrity/ Authorisation/ Resource specific Roles | The solution MUST be capable of associating Users in different Roles to Resource. | A resource can be commenced, collaborated on, approved, published and managed by different users. |  |  |  | The solution associates a user that starts a new resource the owner role.  Other users are invited to work on the resource in one capacity or another (collaborator, etc.)  The solution’s Inheritable Group configuration settings will determine if resources that belong to a group inherit default roles from the group.  Consider the following for Resource Roles:  - Creator - Contributor - Reviewer (may be several types: purpose, structure, accuracy according to SMEs, completeness, legal and/or copyright, etc.)  - Approver - Maintainer - Endorser (may be several types) - Accessor/User |  |
|  | QR-CUS-SEC- 00 | Custom/ Security/ Integrity/ Authorisation/ Permission based | The solution MUST be Permission based, using Roles to collect several Permissions together. | Roles are descriptions of a person's activities, but do not allow fine grain control. |  |  |  | Roles are descriptions of a person's activities, but do not allow fine grain control over permissions.  The solution allows Assigning or Removing Permissions to Roles.  The solution allows assigning Roles to Users.  The solution allows adding or removing Permissions directly to Users.  If a Permission has been assigned multiple times, either directly or via assigned Roles, and one of the assignments is a Removal of the Permission, then the Permission is not given to the target user. |  |
|  | QR-CUS-SEC- 00 | Custom/ Security/ Integrity/ Roles/ Duration | Role Assignments MUST have a Start and End DateTime. | Reduces the risk of Users having access to Roles after they have left their current contract. |  |  |  | This implies a reliance on a scheduled job to Notify members of subscribed roles that that the role is expiring soon (by some configurable delay) permitting them to extend the contract if so desired. |  |
|  | QR-CUS-SEC- 00 | Custom/ Security/ Integrity/ Information/ In Transit/ Cookies/ Secure &HttpOnly | HTTP Cookies MUST be marked ‘Secure’ and ‘HTTPOnly’ |  |  |  |  | The HttpOnly flag helps mitigate the risk of client-side script accessing and/or modifying the contents of protected cookie issued by a server (e.g.: session identifiers).  The Secure flag ensures cookies are not transmitted on encrypted HTTPS channels. |  |
|  | QR-CUS-SEC- 00 | Custom/ Security/ Integrity/ Information/ In Transit/ SQL | Communication to Data stores MUST be Encrypted. | Steps must be taken to protect communication between devices via comms established within 3rd party managed networks. |  |  |  | In SQL Server (other databases may be different), this means that the Connection String includes the following flags:  Encrypt=True; TrustServerCertificate=True |  |
|  | QR-CUS-SEC- 00 | Custom/ Security/ Integrity/ Information/ PII Storage | The solution MUST protect to a higher level at rest Persons and Users Person Information (PI). | Protection of Users Personal Information is protected by regulation.   The obligation contributes significantly to this desired outcome. |  |  |  | By removing PI information from a database, the information that remains is effectively anonymised users’ activity, safer to share as is with reporting services, data warehouses, etc.  The approach implies merging of information before caching and/or presentation. While at appearing daunting, this quickly becomes second nature when needed, which is relatively rarer than one would naively assume.  If using a second database is not an achievable option, consider encrypting the columns containing personal information (PI) of Users. |  |
|  | QR-CUS-SEC- 00 | Custom/ Security/ Integrity/ Information/ At Rest/ Keys | The solution design and operational processes MUST demonstrate appropriate encryption key management. |  |  |  |  | The solution uses a cloud based secure credentials/key storage service to persist keys.  Access to the key storage solution is limited to the deployment pipeline's service account, which uses this access to retrieve credentials and inject them into the deployed system's config file.  Access to the service is monitored and audited.  References: NZISM v2.7 - Section 17.9 Key Management |  |
|  | QR-CUS-SEC- 00 | Custom/ Security/ Integrity/ Source Code/ Contamination | If credentials and/or environment specific properties are checked in, steps must be taken to:  remove the credentials from the code repository and  rotate the credentials so the information is no longer a potential risk. | Environments, whether production or non-production environments, must be protected.  Publicly accessible source code must not become a means of discovering means to bypass a solution's security controls.  Note: Until the source code repository is cleansed the incident must be registered on the project's risk register. |  |  |  |  |  |
|  | **Accountability** | | | |  |  |  |  |  |
|  | QR-CUS-SEC- 00 | Custom/ Security/ Accountability/ Multiple Digital Identities | The solution SHOULD allow a user to associate multiple digital Identities to their system user's record. | Users belong to multiple organisations (schools, organisations, groups) each of which may have their own Identity Provider Service.  A user should be able to use either identity to sign in.  But also keep them separate in case they prefer to keep their system identities separate. |  |  |  | The solution uses external IdPs to persist user credentials.  The solution allows integration with multiple social IdPs (Microsoft Accounts, Google Accounts, etc.).  the solution allows easy integration with Azure AD. |  |

## Functionality

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | # | ID | | Statement | Rationale | Fit Criteria | Response | Comments | Details |  |
|  |  | | o **General** þ **Completeness**  o **Correctness** o **Appropriateness** | | |  |  |  |  |  |
|  |  | | **Completeness** | | |  |  |  |  |  |
|  | QR- CUS-FUNC- 00 | Custom/ Functionality/ Completeness/ Services | | The solution MUST be capable of managing the following core capabilities: - Diagnostics Logging & Error Mgmt  - Permissions Mgmt  - Session Mgmt  - Session Operation Auditing Mgmt  - User Mgmt  - User Digital Identity Mgmt  - Nestable Group Mgmt  - Organisations/ (ie Parent Groups) Mgmt  - Permission to Group Role Mgmt  - Role Request, Invitation, Acceptance, and Assignment Mgmt  - User to Group Role Assigment Mgmt  - Resource Mgmt  - Resource Medata Mgmt  - Resource Route Mgmt - Resource Role Allocation Mgmt | The solution is required to be capable of enabling: - maintenance specialists investigate unexpected behaviour - maintenance specialists report of rate of errors still being reported - audit all operations done by users in any session - provide fine grain control over the permissions provided directly to or inherited by users with roles within one or more nestable groups - permit users to invite others to accept roles within their groups, - permit users to collaborate in different roles (collaborator, reviewer, approver, etc.) on the development of resources (whether they be Records with or without associated uploaded Media) categorised with Metadata - operations specialists route users to replacement resources  - permit resource managers to merge, unmerge, replace, remove, restore resources. |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | QR-CUF-MAIN-00 | Group Invitations | The solution MUST allow appropriate users to invite by notification other users to a role in one of their groups. | Centrally controlled group and user provisioning requires unwieldy processes that do not scale efficiently.  Letting appropriate users organize groups as they see fit improves the speed with which users can begin to benefit from using a system. |  |  |  | the solution allows for a non-centralised process of inviting users, whether internal or external organisation users, to join a group within the solution.  The solution allows requiring approval in some cases of invitations.  *Note: Group configuration can configure that approval is required before issuing invitations.* |  |

## Reliability

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | # | ID | Statement | Rationale | Fit Criteria | Response | Comments | Details |  |
|  | o **General** o **Maturity** þ **Availability**  o **Fault Tolerance** o **Recoverability** | | | |  |  |  |  |  |
|  | **Availability** | | | |  |  |  |  |  |
|  | QR- CUS- REL- 00 | Custom/ Reliability/ Availability/ ??? | The solution MUST meet the Maximum Tolerable Downtime (MTD) |  |  |  |  | Strategies include:  Cloud Provider hosted.  Behind a WAF  Minimising Database connections by design first and local host caching second (for immutable data), and distributed caching last (for mutable data)  Avoiding Data Locking of records by reducing the length of open Transactions, performing updates at the end of Requests.   * Complex reporting queries are relegated to a reporting database * Backups are taken regularly * DR utilizes automated deployment and data restoration pipeline * Automated deployments utilize infrastructure as code and database schema as code to ensure infrastructure and DBs are current.   Automated deployments must take down the system for as short a time as possible. |  |
|  |  |  | Services **MUST** allow users to undo Operations, by using Logical versus Physical state changes. | Users make mistakes. Beyond being and embarrassing and frustrating experience, having support specialists on call to undo mistakes, and designing a system to provide the necessary permissions on behalf of users, is costly to create and provide over the whole service lifespan. |  |  |  | Use State [Active… Removed, Deleted] versus Physical changes to records. (consider indexing optimisations to adjust for larger record count).  Consider queuing & delaying by a configurable delay operations that change state outside of the system (e.g., send emails) permitting the operation to be aborted and returned to an edit state (see Google Mail approach as an example implementation). |  |

## Usability

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | # | ID | Statement | Rationale | Fit Criteria | Response | | Comments | Details |  |
|  | oGeneral o Appropriate Recognisability o Learnability  þ Operability o User Error Protection  o User Interface Aesthetics | | | |  | |  |  |  |  |
|  | **Operability** | | | |  | |  |  |  |  |
|  |  |  | The System MUST be usable without a Mouse | Blind and low vision users have difficulties using devices to select visual targets. |  |  | |  |  |  |
|  | QR- DEF- USA- 00 | Custom/ Portability/ Adaptability/ Configurability/ GUI | The solution **MUST** be configurable by API and GUI. | Operations specialists may be required to change a system setting before the next scheduled release (which will re-apply the setting via API). |  |  | |  |  |  |
|  | QR-DEF-USA-00 | Custom/ Portability/ Operability/ Search/ SearchSummaryItem | The solution MUST provide the means to search across one or more resource types. | Using a single Search entry point improves efficiency by lowering having to find different search entry points. |  |  | |  | Consider searching across various forms of records, returning the various types of records mapped to a common SearchSummaryItem.  Consider enabling the ability to define what types to search over. |  |
|  | QR- CUS- USA- 00 | Custom/ Usability/ Operability/ System Wide | The solution SHOULD be capable of assigning System Wide Roles. |  |  |  | |  | Role Management is the most basic form of authentication. When only applicable System Wide, it is only really usable in an small controlled environment (e.g. small organisation), generally failing to be adaptable to the increasing complexity of Medium-sized organisations or large Enterprises – and certainly not the management of external users grouped as distinct organisations (e.g.: schools). |  |
|  | QR- CUS- USA- 00 | Custom/ Usability/ Operability/ Roles/ Tenancy Wide | The solution SHOULD be capable of assigning Roles specific to Tenancies |  |  |  | |  | Permitting the assignment of Roles to users within different Tenancies improves minorly on System-Wide Role control, but does allow for limited assigning of Roles to users in external organisations (e.g. schools).   Note that the larger the number of external organisations, the larger the risk that there will be exceptions to a common pattern. |  |
|  | QR-CUF-USA- 00 | Custom/ Usability/ Operability/ Roles/ Group Based | The solution SHOULD be capable of assigning Roles specific to nestable Groups. |  |  |  | |  | As expressed above, System Wide Role Management is the most basic form of authentication that has few applicable scenarios. Providing for the assignment of inheritable Roles associated to Nestable Groups permits a wider range of use cases. |  |
|  | QR-CUS- USA- 00 | Custom/ Usability/ Operability/ Roles/ Resources | The solution SHOULD be capable of assigning Roles specific to  Resources |  |  |  | |  | Example Resource roles include:  Creator,  Collaborator, Reviewer,  Approver,  Maintainer, Consumer,  etc. |  |
|  | QR- CUS- USA- 00 | Custom/ Usability/ Operability/ Invite | Permitted Users SHOULD be capable of inviting external Persons to Accept a Role within a Group they belong to. | Support and Operation costs are reduced when Users who know best their own needs have agency and the capabilities to enable collaboration without reliance on a central authority. |  |  | |  | This requirement is associated to the requirement for JIT User table entry development.  This requirement is tied to the security requirement of only enabling Role allocations to specified future dates. |  |
|  | QR-CUS- USA- 00 | Custom/ Usability/ Operability/ Media Malware Detection | Media uploaded by Malware MUST be scanned by the system itself. | The reputation of the organisation is dependent on users trusting its services to do them and their group or organisation no harm. |  |  | |  | The use of a WAF in front of services is acceptable, but brutish in that it won’t return a rich explanatory message as to why the message won’t be accepted. |  |
|  | QR-CUS-USA- 00 | Custom/ Usability/ Operability/ Media/ Malware Detection/ Periodic | Stored Media MUST be periodically scanned. | The reputation of the organisation is dependent on users trusting its services to do them and their group or organisation no harm. |  |  | |  | Media may have been uploaded before malware detection services have been received an update capable of detecting the virus. |  |

## Performance

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | # | ID | Statement | Rationale | Fit Criteria | Response | Comments | Details |  |
|  | o**General** þ **Time Behaviour** þ **Resource Utilisation** | | | |  |  |  |  |  |
|  | **Time Behaviour** | | | |  |  |  |  |  |
|  | QR-CUS- PER- 00 | Custom/ Performance/ Time/ Operation Readiness | 90% of Web Views MUST be ready for next Operation within 0.6 seconds, the maximum permitted for the remaining % being 10 times as long, 6 seconds. | Users must be provided an efficient system to improve their perceptions their ISO-25022 defined qualities. |  |  |  | When developing custom solutions, where one has more control over the enforcement of design, development obligations, a higher expectation of performance can be met, therefore obligated. |  |
|  | **Resource Utilisation** | | | |  |  |  |  |  |
|  | QR-CUS- PER- 00 | Custom/ Performance/ Resource Utilisation/ CPU |  |  |  |  |  |  |  |

## Maintainability

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | # | ID | Statement | Rationale | Fit Criteria | | Response | Comments | | Details |  |
|  | þ**General** o**Modularity** þ**Reusability** o**Analysability** þ**Modifiability** þ**Testability** | | | |  | |  | |  |  |  |
|  | **General** | | | |  |  | | |  |  |  |
|  | QR-CUS-MNT- 00 | Custom/ Maintainability/ General/ PaaS | The solution MUST be designed as a PaaS service before requiring a Decision to permit a CaaS based solution. | PaaS based solutions require less operational responsibility and therefore Cost compared to CaaS based solutions. |  | |  | |  | Development using PaaS is more economical permitting a longer service lifespan and better Cost/Benefit.  Some implications exist regarding constraints on the ability to use subnets to control traffic between solution devices. |  |
|  | **Reusability** | | | |  | |  | |  |  |  |
|  | QR-CUS-MNT- 00 | Custom/ Maintainability/ Reusability/ UI Tier | Service Clients MUST be developed in a separate Tier from the Service APIs. | User Interfaces are the component quickest to become dated. Being able to update them, without fear of impacting other tiers, improves the longevity and value of the service. |  | |  | |  | This requirement aligns with an API-First strategy.   When the interface is developed as a server-rendered part of the service, APIs can be bypassed and therefore there is no impetus or guarantee that all necessary APIs will be developed and sufficiently tested. |  |
|  | **Modifiability** | | | |  | |  | |  |  |  |
|  | QR-CUS-MNT- 00 | Custom/ Maintainability/ Modifiability/ DDD | Domain Driven Design principles MUST be applied to the design of components. | The maintainability of systems and replated savings in effort and cost is improved when DDD is used to lay out components and how they are integrated together. |  | |  | |  | The component structure of the system is as a system of logical modules, each with the following physical assemblies/libraries:  - Presentation, for view models and API controllers, - Application, a UI technology agnostic assembly, orchestrating calls to either the Infrastructure or Business assemblies. - Infrastructure, orchestrating calls to underlying framework or 3rd party services APIs (caching, storage, IdPs, search, rule and workflow engines, etc.) - Business (services specific to the business case) - Common, an assembly of logical models and entities used in common by Application, Infrastructure and Domain assemblies. |  |
|  | QR-CUS-MNT- 00 | Custom/ Maintainability/ Modifiability/ OO | Custom Code MUST be developed according to SOLID & GRASP Object Oriented Patterns | The maintainability of systems and related savings in effort and cost is improved when SOLID principles are applied from the start. |  | |  | |  |  |  |
|  |  | Custom/ Maintainability/Modifiability/ Modules | The development and installation of another logical Module MUST NOT impact other Modules. | Implementing coding practices to adhere to separation of duties improves maintainability, therefore lowers maintenance cost. |  | |  | |  |  |  |
|  | QR-CUS-MNT- 00 | Custom/ Maintainability/ Dependencies/ Currency | The solution’s dependencies MUST be maintained to the latest major and minor -2 version or 6 months since released, which ever is shortest. | Systems must be delivered in a state that permits meeting user expectations for the longest service lifespan. |  | |  | |  | This implies both maintenance specialists and development specialist roles having practices in place to monitor and alert to new versions being available. |  |
|  | **Testability** | | | |  | |  | |  |  |  |
|  | QR-CUS-MNT- 00 | Custom/ Maintainability/ Testability/ Unit Tests | The solution MUST be delivered with a report of the coverage and passing of static Unit Tests associated to developed code. | Implementing coding practices to adhere to separation of duties improves maintainability, therefore lowers maintenance cost. |  | |  | |  | The unit tests are defined by Acceptance Test Statements in technical work items developed before a developer begins developing the system code (i.e., following best practice Test Driven Development).  The static tests are run as post-compilation but pre-deployment step within the automated deployment pipeline. |  |

## Portability

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | # | ID | Statement | Rationale | Fit Criteria | Response | Comments | Details |  |
|  | o**General** o**Adaptability** þ**Installability** o**Replaceability** | | | |  |  |  |  |  |
|  | **Installability** | | | |  |  |  |  |  |
|  | QR-CUS- POR- 00 | Custom/ Portability/ Installability/ NonDisruptive | Installations SHOULD be to a secondary environment, to which users are redirected after the deployment has been successfully tested. | Disruption to end users must be minimised. |  |  |  | If a secondary target environment is used, disruption to end users can be made imperceptible in most cases. |  |
|  | QR-CUS- POR- 00 | Custom/ Portability/ Installability/ Rollback | An aborted or failed installation of the solution MUST be capable of being rolled back without change to the system. | Disruption to end users must be minimised. |  |  |  | This implies (but does not commit to) a strategy involving installing to another environment which if successful is switched over to. |  |
|  | QR-CUS- POR- 00 | Custom/ Portability/ Installability/ Idempotent | Installation of the solution MUST be idempotent. | Re deployment over an existing deployment must not change the solution’s state. |  |  |  |  |  |
|  | QR-CUS-POR- 00 | Custom/ Portability/ Installability/ OS Independent | ‘Custom Code’ MUST be operating system agnostic. | The solution should be as portable to different environments as possible to take advantage of changes made possible over the whole service’s lifespan. |  |  |  | ‘Custom Code’ should be developed using cross-OS platform & frameworks (.NET Core, Python, etc.) preferably deployable to managed platforms (e.g.: PaaS) that remove the need to know the underlying OS. |  |

# Default Internal Requirements

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
| QR-DEF-MAIN-00 | Replaceability | The solution SHOULD allow the consolidation of the capabilities of one or more existing systems. | Consolidation of new solutions should be considered based on reducing complexity, improve maintainability of the organisation's system, decreasing cost. | NRN: this requirement cannot be answered by respondents. |  |  |

Maintainability

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| QR-DEF-MAIN-00 | Modifiability/Lockout | Any development or ongoing support contract prepared to deliver this solution MUST NOT exclude this organisation from actively participating in development and operations. | This organisation must not accept locking or lockout contractual conditions. | The solution’s automated pipeline, running unit tests, integration tests, behaviour tests are there to catch errors in logic before checkin branches are accepted and merged.  Any fixes required are the whole project team's responsibility, allocated to the group most knowledgeable as to the cause and resolution, dependent on availability. |  |  |
| QR-MT-B-06 | **Baseline** | **Maintainability/Accessible Code Repository** | Custom code, configuration or test scripts written for this solution **MUST** be maintained in an organisation accessible and clonable code repository. | Code written for this organisation must be analysable for code security and quality. |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| QR-DEF-MAIN-00 | Replaceability | The solution SHOULD allow the consolidation of the capabilities of one or more existing systems. | Consolidation of new solutions should be considered based on reducing complexity, improve maintainability of the organisation's system, decreasing cost. | NRN: this requirement cannot be answered by respondents. |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| QR-DEF-MAIN-00 | Reusability | The solution’s components MAY be reusable by other systems. | The cost benefits of reusing components and services must be weighed against the cost of complexity and cross-dependencies. | The solution’s components are re-using other solution's components.  This solution's components are envisioned as being reusable by other solutions. |  |  |

Training to stop checking in credentials.

# Conclusion

Appendices

Appendix A - Document Information

### Authors & Collaborators

Sky Sigal, Solution Architect

### Images

[Figure 1: Project Specific Tier of Quality Requirements 11](#_Toc149570784)

[Figure 2: Custom Developed Specific Tier of Quality Requirements 13](#_Toc149570785)

[Figure 3: Ministry Specific Tier of Quality Requirements 34](#_Toc149570786)

[Figure 4: Organisation Specific Tier of Quality Requirements 40](#_Toc149570787)

[Figure 5: Education Sector Specific Tier of Quality Requirements 47](#_Toc149570788)

[Figure 6: NZ Government Sector Specific Tier of Quality Requirements 51](#_Toc149570789)

[Figure 7: Default System Base Tier of Quality Requirements 62](#_Toc149570790)

### Tables

[Table 2: Table of Default System Security Quality Requirements 64](#_Toc149570774)

[Table 3: Default System Privacy Quality Requirements 77](#_Toc149570775)

[Table 4: Default System Functionality Quality Requirements 80](#_Toc149570776)

[Table 5: Default System Usability Quality Requirements 85](#_Toc149570777)

[Table 6: Default System Reliability Quality Requirements 100](#_Toc149570778)

[Table 7: Default System Performance Quality Requirements 109](#_Toc149570779)

[Table 8: Default System Maintainability Quality Requirements 115](#_Toc149570780)

[Table 9: Default System Portability Quality Requirements 129](#_Toc149570781)

### References

*IT Project Guidance – Requirement Development*

**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.

### Terms

Refer to the project’s Glossary.

##### ‘Custom Code’

: all information required to develop, deploy, configure, integrate, provision any aspect of the service, excluding proprietary information such as Software as a Service (SaaS).

##### Custom Deliverables

: Code & Documentation.

Prod Data Environments: environments that contain production data that includes confidential information.

##### 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.

Non-Prod Data Environment: *all* other environments (BT, DT, ST, UT, PP, TR, etc.) than the Prod environment.

##### Organisation Provided

: if the organisation that is procuring the service has a service, then it must be used, unless mutually decided otherwise. If the organisation does not have the service, governance can mutually agree to use an alternate service until the organisation provides one to which it can be moved.

##### Proprietary Code

: code that indirectly or directly belongs to and is controlled by a vendor distinct from the organisation purchasing the development of the solution (the Ministry). Does not include code used to develop a SaaS.

##### SaaS

: *Software as a Service*, a managed service rented via a subscription account.

Appendices B – ISO-25010 Qualities

ISO-25010 defines Headers and Subheaders with their individual descriptions. These are listed below.

### Delivery

While *Installability* is a quality defined under *ISO-25010/Portability* (see further down), the subject of Delivery wider, and called out first, with its own section.

### Security

Security is defined in ISO-25010 as the practice of maintaining the confidentiality, privacy, integrity and accountability of data changes by controlling authorised access, use and disclosure, while preventing unauthorised use, disruption, modification or destruction.

ISO-25010 recommends considering the following qualities and their descriptions when considering a solution:

#### Confidentiality

: the degree to which the solution ensures data is accessible only by those authorised to do so.

#### Integrity

: the degree to which the solution prevents unauthorised access, modification of systems and the information they manage.

#### Non-Repudiation (of Action) and Accountability (by Whom)

: are associated in that non-repudiation defines the degree to which the solution can prove that actions have been taken, and accountability is being able to associate the non-repudiatable (audited) activity to a specific user.

#### Authenticity

: defines the degree to which the identity of a user can be claimed.

#### Availability

: a prerequisite for Security but is treated separately, under Reliability.

Note:  
Security and Privacy risk assessments, and matching Statement of Applicability listing required controls for the solution will be conducted during the solution’s design phase.  That process will further augment the security requirements listed below.

### Privacy

Privacy is not a defined ISO-25010 concern, but since the last edition, has become a first-class quality.

### Functionality

#### Functional Completeness

: …

#### Functional Correctness

: …

#### Functional Appropriateness

: …

### Performance

Efficiency, throughput, capacity, response time and resource consumption for Online Transaction Processing (OLTP) and background processes.

The system's ability to complete defined volumes of computer assisted Business Transactions (online and background operations) within acceptable times and with an acceptable consumption of resource.

ISO-25010 recommends considering the following qualities and their descriptions when considering a solution:

#### Time behaviour

: is the degree to which the response and processing times and throughput rates of a solution, when performing its functions, meets requirements.

#### Resource Utilisation

: is the degree to which the amounts and types of resources used by a product or system, when performing its functions, meets requirements.

#### Capacity

: is the degree to which the maximum limits of the solution meet or exceed requirements.

### Compatibility

#### Co-existence

: …

#### Interoperability

: …

### Usability

Usability is how easy and efficient it is for an end user to correctly, accurately, and safely use the solution.

It also ensures constancy with other organisation applications.

It is sometimes called system ergonomics, accessibility and ease of use.

ISO-25010 recommends considering the following qualities and their descriptions when considering a solution:

#### Appropriateness recognisability

: is the degree to which users can recognize whether a solution is appropriate for their needs.

#### Learnability

: is the degree to which a solution enables the user to learn how to use it with effectiveness, efficiency and in the case of an emergency.

#### Operability

is the degree to which a product is easy to operate, control and appropriate to use.

#### User Error Protection

: is the degree to which a solution protects users against making errors.

#### User Interface Aesthetics

: is the degree to which a user interface enables pleasing and satisfying interaction for the user.

#### Accessibility

: is the degree to which a solution can be used by people with the widest range of characteristics and capabilities to achieve a specific goal in a specified context of use.

### Reliability

Reliability of a solution is - a request being processed by that system according to agreed business logic and a valid response being returned by the system in the time expected. The time expected is influenced and constrained by the performance requirements of the system.

Fault tolerance is defined as the immunity of the system in the event of an unexpected internal or external fault. This is evaluated in the context of the likelihood of the fault, the severity of the fault and resulting impact on end users.

*Recoverability* is a subset of *Reliability* andis the ability to re-establish the normal functioning of a system after a significant external event, such as a disaster. A disaster implies total loss of access or use of the system by end users (e.g., production) where the outage has, or will, extend beyond the Maximum Tolerable Downtime (MTD).

The MTD and ‘normal functionality’ are agreed between IT (or the vendor) and the business owner of the system in the Service Level Agreement (SLA).

*Recoverability* uses activities such as data protection, Disaster Recovery (DR), and Business Continuity Planning (BCP) to ensure the recoverability of systems

ISO-25010 recommends considering the following qualities and their descriptions when considering a solution:

#### Maturity

: is the degree to which a solution needs for reliability, under normal operation.

#### Availability

: is the degree to which a solution is operational and accessible when required for use.

#### Fault tolerance

: is the degree to which a solution operates as intended despite the presence of hardware, software or user faults.

#### Recoverability

: is the degree to which, in the event of an interruption or failure, a solution can recover the data directly affected and re-establish the desired system state.

### Maintainability

Maintainability is the ability to maintain the system efficiently: find and remove faults, improve performance, carry out modifications and infrastructure upgrades. It is also known as modify-ability, enhancement, fault detection, isolation and repair.

Traceability is the ability to retain details of specified business, transactional, and system activity and data changes, including normal, abnormal, and error conditions with selective control. It is also called auditability and audit trail.

ISO-25010 recommends considering the following qualities and their descriptions when considering a solution:

#### Modularity

: is the degree to which a solution is composed of discrete components such that a change to one has minimal impact on others.

#### Reusability

: is the degree to which an asset can be used in more than one system or reused to build other assets.

#### Analysability

: is the degree of effectiveness and efficiency with which it is possible to assess the impact of a solution, a change, a failure, or determine what requires change.

#### Modifiability

: is the degree to which a solution can be effectively and efficiently modified without introducing defects or degrading operational quality.

#### Testability

: is the degree to which test criteria can be established, and determination of whether the solution meets them.

### Portability

ISO-25010 recommends considering the following qualities and their descriptions when considering a solution:

#### Adaptability

: is the degree to which a solution can effectively and efficiently be adapted for different and evolving hardware, software, or environments.

#### Installability

: is the degree of effectiveness and efficiency in which a solution can be successfully installed/uninstalled in a specified environment.

#### Replaceability

: is the degree to which a solution can replace another system for the same purpose in the same environment.