IT Project Guidance

Glossary of IT Specific Terms:   
Quality Assurance

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

Included are the meanings of acronyms and industry terms used to describe aspects of system quality assurance.

## Contents

[Description 1](#_Toc146637638)

[Contents 2](#_Toc146637639)

[Introduction 3](#_Toc146637640)

[Objective 3](#_Toc146637641)

[Terms & Acronyms 4](#_Toc146637642)

[System Quality Assurance Terms & Acronyms 4](#_Toc146637643)

[Quality Assurance as Code (QAaC) 4](#_Toc146637644)

[Recovery Point Objective (RPO) 4](#_Toc146637645)

[Recover Time Objective (RTO) 4](#_Toc146637646)

[RPO 4](#_Toc146637647)

[RTO 4](#_Toc146637648)

[Test Summary Report (TSR) 4](#_Toc146637649)

[TSR 4](#_Toc146637650)

[Appendices 5](#_Toc146637651)

[Appendix A - Document Information 5](#_Toc146637652)

[Images 5](#_Toc146637653)

[Tables 5](#_Toc146637654)

[References 5](#_Toc146637655)

[Review Distribution 5](#_Toc146637656)

[Audience 5](#_Toc146637657)

[Diagrams 5](#_Toc146637658)

## Introduction

## Objective

To develop a common understanding of terms used to deliver services with an IT component.

# Terms & Acronyms

## System Quality Assurance Acronyms

#### AAA

: Arrange, Act, Assert.

#### BDD

: Behaviour Driven Design

#### C&A

: *Certification and Assurance*.

#### DDD

: *Domain Driven Design*.

#### FR

: *Functional Requirements*.

#### NFR

: *Non-Functional Requirements*.

#### QAaC

; *Quality Assurance as Code.*

#### RPO

: *Recovery Point Objective*.

#### RTO

: *Recovery Time Objective*.

#### TSR

: *Test Summary Report*.

## System Quality Assurance Terms

#### Accreditation

: the formal recognition by a governing body or authority that the software system complies with specific standards or regulations. Accreditation is given after the system has been evaluated and assurance has been given that it is approvable to be used in a particular environment.

#### Arrange, Act-Assert (AAA)

: a pattern for arranging tests scenarios. See *Given, When, Then*.

#### Assurance

: the confidence or guarantee that a system or service meets certain standards or requirements to meet expected qualities. Assurance is about proving that the system is reliable, secure, and performs as expected. This quality assurance is accomplished through testing, validation, and reviews. Assurance may finish with Certification.

#### Behaviour Driven Development (BDD)

: an approach to automated software testing using natural language constructs using domain language to describe the behaviour of code. BDD borrows the concept of ubiquitous language from Domain Driven Design (DDD) See *Gherkin*.

#### Certification

: is an optional subset of the Assurance process. It is the process of obtaining written assurance from an independent body as to the conformity of a service or system to external international or industry specific requirements (ISO, PCI, DSS, etc.). Without diminishing its importance, certification is sought less often than assurance. Examples include obtaining ISO 27001 Certification of an information system intended for government use, or PCI DSS Certification of a payment system for use by an financial institution, or ISO-13485 Certification for a System with Patient Data for a Healthcare agency, etc.

#### Certification and Assurance

: For the most part, the process is really Certification that a system has met minimum degrees of Qualities and optionally may be completed with a check that the (purchased) system or vendor has obtained Certification from a trusted 3rd certification body.

#### Domain Driven Design (DDD)

: a major software design approach focusing on modelling large complexity as smaller distinct domains.

#### Domain Specific Language

: a computer language specialised to a particular application domain (see *Gherkin*).

#### Functional Requirements

: one half of the logical *System Requirements*.

#### Gherkin

: a *Domain Specific Language (DSL)* for defining acceptance tests of behaviours (which can be used to describe both functional or non-functional behaviours). See *BDD*. See *Given, When, Then*.

#### Given, When, Then

: the 3 core aspects of a Scenario as described within Gherkin. See *Arrange, Act, Assert*.

#### ISO-9126

: *deprecated* standard of qualities desired of an IT system.

#### ISO-25010

: International standard for defining and measuring desirable system qualities, composed of agreed topics and subtopics. An improvement over earlier ISO-9126 and defunct FURPS+. Covers:

* Functional Suitability[[1]](#footnote-2),
* Performance Efficiency,
* Compatibility,
* Interaction Capability,
* Reliability,
* Security,
* Maintainability,
* Flexibility,
* Safety.

#### ISO-25012

: International standard for qualities of the data managed by a system (see ISO-25010).

#### Non-Functional [Quality] Requirements

: one part of the logical set of *System Requirements*, these are the system’s quality requirements, as defined and organised by ISO-25010, ISO-25012 and ISO-25022.

#### Penetration Testing

: a black box test of a system’s availability, confidentiality, authentication authorisation and auditability, optionally supplemented by a white box analysis of source code, deployment, networking, configuration and hardening. Preferably done regularly as *QAaC* within a *DevOps* based *deployment pipeline*, but still done too often rarely and manually using *ClickOps*.

#### Pen Test

: a vernacular shorthand name for a Penetration Test.

#### Performance Testing

: a subset of Performance testing, testing the number of concurrent users and operations can be responded to within acceptable times and resource consumption constraints. Preferably done regularly as *QAaC* within a *DevOps* based *deployment pipeline*, but still done too often rarely and manually using *ClickOps*.

#### Perf Test

: a vernacular shorthand name for *Performance Test*.

#### Quality Assurance as Code (QAaC)

: Current best practice approach to testing the qualities of services by developing a set of tests that can be automated – rather than developing manual processes that can take weeks to months to perform.

#### Recovery Point Objective (RPO)

: the agreed definition of the maximum acceptable amount of data loss after an unplanned data-loss incident, expressed as a duration.

#### Recover Time Objective (RTO)

: the agreed duration of time before a system in available again after an unplanned incident.

#### System Requirements

: the logical set of the system’s *Functional Requirements* and *Non-Functional Quality Requirements*. A third category of requirements to assess to provide Assurance are the delivery Transitional Requirements.

#### Test Summary Report (TSR)

: the summary of the Quality Assurance performed by *Test Analysts*.

#### Transitional Requirements

: separate from the *System Requirements*, these are project *delivery* requirements to define how to move from current state to delivery state. Can involves workflow management, stakeholder reporting, dashboards, automation pipelines, resource training, data extraction, quality checking, transformation, migration, and importing, process development for provisioning, onboarding & supporting users, etc.

#### Triple-A

: an alternative to AAA to refer to the Arrange, Act, Assert test development pattern.

# Appendices

Appendix A - Document Information

### Authors & Collaborators

Author: Sky Sigal, Solution Architect

### Versions

0.1 Initial Draft

### Images

**No table of figures entries found.**

### Tables

**No table of figures entries found.**

### References

**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 Architect |  |
| Roger Govind, Security Architect |  |
| Archana Sahani, Business Analyst |  |
| Dijana Sneath, Business Analyst |  |
| Vincent Weirdsma, Lead Developer |  |

### Audience

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

### 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.

1. Noting that Functional suitability is not the same as Functionality requirements. It is the measure of Appropriateness, Correctness and Completeness, and not what would be specified in Functional Requirements. [↑](#footnote-ref-2)