Office of the Chief
Information Officer

Architecture, Standards and Planning Branch
Office of the CIO | Province of BC
People | Collaboration | Innovation

Development Standards for Information Systems and Services

Architecture, Standards and Planning Branch
Office of the CIO ● Province of BC
People ● Collaboration ● Innovation

Document Version 3.0
Published: May 2015

Replaces: Version 2.4

Development Standards for Information Systems and Services

Table of Contents

Document Control

Date	Author	Version	Change Reference
April 27, 2015	Walker/Wilson	FINAL 3.0	GCIO Approved & Published
April 10, 2015	Walker/Wilson	DRAFT 3.0	Released for ASRB Review.
April 13, 2010	Robert Walker	DRAFT 2.2	Sections 1.1, 1.2, 1.3 in development
May 25, 2010	Robert Walker	FINAL 2.4	ASRB endorsed

Version 3.0 (May 2015) highlights:
- NEW CHAPTER: 2.0 Open Source Development.
- NEW SECTION: 2.1 Source Code Management

Final Version 2.4 (May 2010) highlights:
- Endorsed by ASRB May 20, 2010.

Draft Version 2.2 (Mar 2010) highlights:
- The DRAFT is now issued as a document.
- An overview to Requirements Management has been added.
- Notes to reviewers have been added.
- Small wording changes for improved clarity.
- Section 1.2 a statement concerning assumptions was added.
- Section 1.2 "approval process" changed from SHOULD to MUST
- Section 1.2 "design decisions" changed from SHOULD to MUST

INTRODUCTION

This document contains a family of standards governing the development of information systems and services.

APPLICABILITY

This family of standards applies when an information system or service is being developed, modified or procured by, or on behalf of, the Province of British Columbia.

COMPLIANCE

There is no compliance schedule for these standards. The GCIO exemption process applies.

NOTES TO USERS AND REVIEWERS

<u>Feedback</u>

Feedback can be sent to robert.walker@gov.bcv.ca or Todd.Wilson@gov.bc.ca

Please identify the relevant section or paragraph, as appropriate.

<u>Terminology</u>

The term "MUST" is defined as an absolute requirement of the specification.

"SHOULD" means that valid reasons to use alternate methods may exist in particular circumstances, but the full implications must be understood and carefully weighed before choosing a different course.

1. REQUIREMENTS MANAGEMENT	Effective: 2010-07-01
	Reviewed: 2015-04-01
1.0 Introduction	Changed: 2015-04-01

Overview

The Requirements Management standards have been kept intentionally simple to maximize their usefulness in the widest of circumstances.

During review of the state of requirements management with architects and business analysts serving the province, many of the issues identified were quite basic. Therefore, as a first step, these standards focus on setting foundations for good practice.

The above approach has some important advantages:

1. These standards can be implemented without a high degree of organizational maturity.

2. These standards can be adapted for projects large and small.

3. Where mature development processes exist, these standards will fit without disruption.

4. These standards will not over-burden program areas with excess process.

Each standard is brief, easy to understand and is designed to be relatively self-contained.

These standards are not a comprehensive framework. They are intended as a simple tool for helping organizations achieve successful outcomes on IM/IT initiatives.

They represent enough governance to address some known concerns and provide a basis for tangible improvements.

INTENTIONALLY BLANK.

1. REQUIREMENTS MANAGEMENT	Effective: 2010-07-01
	Reviewed: 2015-04-01
1.1 Definitions	**Changed: 2015-04-01**

Purpose

This section specifies a government standard for the use of requirements in the development of information systems and services.

Context

To help ensure the fitness-for-purpose of IM/IT systems and services the OCIO is promoting a requirements-centric approach to development. Good requirements help align IM/IT with program area needs and objectives.

This standard provides specific guidance on the use and management of requirements.

In this standard the term "requirements" encompasses all types of requirements: business requirements, user requirements, user stories, performance and scalability requirements, security requirements, etc.

This standard applies when an information system or service is being developed, modified or procured by, or on behalf of, the Province of British Columbia.

Standard Definitions

Constraint[1]:

A **constraint** is a restriction on the degree of freedom. Constraints can be economic, political, technical, or environmental and may pertain to your project resources, schedule, target environment, or to the system itself. A constraint can overlap with business rules and requirements. Whether something is thought of as a constraint or a requirement is usually not important.

---

[1] Adapted from Agile Modeling

Requirement[2]:

    A. *A condition or capability needed by a stakeholder to solve a problem or achieve an objective.*

    B. *A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed document.*

    C. *A documented representation of a condition or capability as in definition (A) or (B).*

Stakeholder:

    A **stakeholder** is a person, group, organization, or system who affects or can be affected by the actions or outcomes of a project. For example, as it applies to a system or service the stakeholders may be any or all of: the owners, the sponsor, the operators, the users, the testers, the implementers, developers and customers.

Additional Guidance

- Requirements can be elicited through: interviews, facilitated sessions, mock-ups, prototypes, surveys and questionnaires.

- Requirements can be documented in various ways: declarative statements, diagrams, pictures, lists and tables.

- Gathering and documenting requirements is a collaborative process of discovery and validation.

References

1. SOFTWARE REQUIREMENTS, Second Edition, Karl E. Wiegers

2. IEEE Guide for Developing System Requirements Specifications
   a. IEEE Std 1233, 1998 Edition;

3. IEEE Recommended Practice for Software Requirements Specifications
   a. IEEE Std 830-1998;

4. OCIO – Information Security Policy 2.1.3 - Information security responsibilities must be documented.

---

[2] Adopted from the IEEE

1. REQUIREMENTS MANAGEMENT	Effective: 2010-07-01
	Reviewed: 2015-04-01
1.2 Requirements Practices	Changed: 2015-04-01

Purpose

This section specifies a government standard for the use of requirements in the development of information systems and services.

Context

To help ensure the fitness-for-purpose of IM/IT systems and services the OCIO is promoting a requirements-centric approach to development. Good requirements help align IM/IT with program areas needs and objectives.

This standard provides specific guidance on the use and management of requirements.

In this standard the term "requirements" encompasses all types of requirements: business requirements, user requirements, system requirements, security requirements, etc.

This standard applies when an information system or service is being developed, modified or procured by, or on behalf of, the Province of British Columbia.

Standard

1. Known requirements and constraints MUST be documented.

2. Known assumptions MUST be documented.

3. Known uncertainties MUST be documented.

4. The documentation from items 1- 3 above SHOULD be peer reviewed for completeness and clarity of purpose.

5. Requirements and constraints MUST be subjected to an approval process.

6. Approved requirements MUST have a sponsor and/or owner.

7. Final design and implementation choices MUST be based on approved requirements/constraints.[3]

---

[3] A "final design choice" is generally one that is irreversible or involves commitment; please see additional guidance section.

Additional Guidance

Items 1 - 3 do not prescribe how things are to be documented; therefore the methods and tools used for documentation can be adapted to the size and needs of each project.

Item 3: In addition to the above, the principal intent is to ensure that proof of concept projects have focused, tangible objectives (i.e. resolving specific documented uncertainties).

Item 4: A peer review is a review by one or more people qualified to provide constructive feedback on completeness and clarity of purpose, but who have not directly participated in producing the documents.

Item 6: Often the rationale for a requirement needs to be further understood. For that reason the source of a requirement needs to be known. If the actual source of a requirement is not the sponsor/owner then the source should be documented.

Item 7: The principle intent is to restrain unapproved, unsponsored requirements/constraints from driving the final design.

References

1. SOFTWARE REQUIREMENTS, Second Edition, Karl E. Wiegers

2. IEEE Guide for Developing System Requirements Specifications
    a. IEEE Std 1233, 1998 Edition;

3. IEEE Recommended Practice for Software Requirements Specifications
    a. IEEE Std 830-1998;

4. OCIO – Information Security Policy 2.1.3 - Information security responsibilities must be documented.

1. REQUIREMENTS MANAGEMENT	**Effective: 2010-07-01**
	Reviewed: 2015-04-01
1.3 Properties of Requirements	**Changed: 2015-04-01**

Purpose

This section specifies a government standard for the use of requirements in the development of information systems and services.

Context

To help ensure the fitness-for-purpose of IM/IT systems and services the OCIO is promoting a requirements-centric approach to development. Good requirements help align IM/IT with program areas needs and objectives.

This standard provides specific guidance on the use and management of requirements.

In this standard the term "requirements" encompasses all types of requirements: business requirements, user requirements, system requirements, security requirements, etc.

This standard applies when an information system or service is being developed, modified or procured by, or on behalf of, the Province of British Columbia.

Standard

Properties of Requirements

1. Each requirement MUST be testable against objective criteria.

2. Each requirement SHOULD be specific: free of ambiguity and not open to unconstrained interpretation.

3. A requirement SHOULD NOT arbitrarily restrict solution choices.

4. Each requirement MUST be uniquely identified.

5. A requirement statement SHOULD express only one requirement, no more.

Additional Guidance

Eliciting requirements is an iterative process. At the beginning of the process a requirement may be loose, open to interpretation, not fully explored. That is normal. As further iterations complete, the requirement becomes better defined, it matures. Finally the requirement is ready to graduate. It is ready to be approved and move to the next stage in the overall procurement process. It is these mature requirements that are the focus of this standard.

Item 1:
If a requirement is not testable or verifiable in any objective way, this suggests a problem. It could be a wording problem or there might be uncertainty that needs resolution. If a requirement is not verifiable by some objective criteria, then there is no way to validate that the requirement has been satisfied or otherwise delivered.

Item 4:
Once assigned, a unique identifier should never be reused or reassigned even if the requirement itself is dropped. Item 4 amounts to uniquely numbering: statements, diagrams, pictures, use cases, user stories or features.

Item 5:
Using compound sentences to describe requirements is a bad practice. A requirements statement, i.e. a sentence, should express one coherent requirement, not a multitude of requirements.

References

1. SOFTWARE REQUIREMENTS, Second Edition, Karl E. Wiegers

2. IEEE Guide for Developing System Requirements Specifications
   a. IEEE Std 1233, 1998 Edition;

3. IEEE Recommended Practice for Software Requirements Specifications
   a. IEEE Std 830-1998;

4. OCIO – Information Security Policy 2.1.3 - Information security responsibilities must be documented.

2. OPEN SOURCE DEVELOPMENT	Effective: 2015-05-01
	Reviewed: 2015-04-01
2.1 Source Code Management	**Changed: 2015-04-01**

Purpose

This section specifies a government standard for source code management and version control of open source projects.

Context

Open source development opens new frontiers of possibilities for faster, cheaper, better ways of addressing public needs. The purpose of this standard is to ensure that open source development undertaken across the many diverse initiatives of the Province remains governable as it becomes ubiquitous.

This standard provides specific guidance on the use of Open Source software repositories. It provides a single coherent set of policies for BCGov Open Development to achieve a collaborative model of software development whereby the human-readable source code used to produce a given piece of software is made freely available for others to adapt or improve upon as they see fit.

This standard applies to software artefacts developed or shared as Open Source on behalf of the Province of British Columbia. In the narrative that follows open source software projects initiated on behalf of the Province of BC will be referred simply as projects.

Standard

1. Projects MUST use the github.com service for Open Source code management.

2. Projects MUST reside within the "BCGov" organization domain on github.com.

3. Projects MUST be approved by the appropriate Deputy Minister or designate.

4. Projects MUST adhere to the Content Approval Checklist.

5. Project components on github.com MUST consist only of Open Repositories – any content that must be protected must be managed in house.

6. All activity in the BCGov Open Source environment MUST follow the BC Open Source Development Employee Guide.

7. Repository Administrators MUST use 2-factor authentication.

8. All repository content MUST apply an approved license.

9. All repository content SHOULD apply one or more of the pre-approved [default licenses](#).

Additional Guidance

A software development project may elect to embrace an open source philosophy at any stage in the development lifecycle. However, the OCIO recommends that an "Open from Day One" strategy be taken to avoid the difficulties that naturally arise when committing later in the lifecycle.

The Risk Management Branch and Legal Services Branch have the authority to approve an open source license.

References

1. OCIO –[BC Policy Framework for Github](#).

---END