

A picture containing outdoor, sky

Description automatically generatedA picture containing outdoor, sky

Description automatically generated

Improve | Innovate | Digitise

Improve | Innovate | Digitise

Status Draft / Issued

Monday, 2 March 2020

Business Analysis (BAPL)

Requirements Management Plan (Full)

<<Project Name>>

**Document Controls**

**Project Information**

|  |  |
| --- | --- |
| Organisation | << Organisation Name >> |
| **Project Sponsor** | << Project Sponsor >> |
| **Project Name** | << Project Name >> |
| **Project Number** | << Project ID >> |

**Contact for Enquires and Proposed Changes**

If you have any questions regarding the information in this document or suggestions for improving the document, please forward details to Business Analysis (BAPL) at the following contact point:

Email: [info@business-analysis.com.au](mailto:info@business-analysis.com.au)

Website: [www.business-analysis.com.au](http://www.business-analysis.com.au)

Phone: 1300 33 11 64

**Version History**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Changed By | Nature of Amendment |
|  |  |  |  |
|  |  |  |  |

**Document Approval**

The content of this document is hereby agreed to by:

|  |  |
| --- | --- |
| Name | Position |
|  |  |

**Stakeholder Review**

The content of this document has been reviewed by:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Position | Initial | Date |
|  |  |  |  |
|  |  |  |  |

**Copyright © Business Analysts Pty Ltd 2012**

Copyright © Business Analysts Pty Ltd 2012. Copyright protects this publication. Except for purposes permitted by the Copyright Act, reproduction by whatever means is prohibited without the prior registration and written permission of Business Analysts Pty Ltd. Inquiries should be addressed to info@business-analysis.com.au. Do not remove this copyright statement.

Copyright © This document has been licensed to <<Organisation Name>>. It must not be copied or reproduced in any way whatsoever outside of <<Organisation Name>> without the authority of Business Analysts Pty Ltd. This document is uncontrolled when printed. An electronic database manages and stores the controlled version.

**Disclaimer**

Business Analysis (BAPL) does not guarantee or warrants the results obtained while using this document and conducting business analysis activities. Usage of this document is strictly under the proviso that the user takes all responsibility for the document outcomes.

Table of Contents

[1. Executive Summary 1](#_Toc18333158)

[2. Introduction 2](#_Toc18333159)

[2.1. Purpose 2](#_Toc18333160)

[2.2. Intended audience 2](#_Toc18333161)

[2.3. Definitions, Acronyms and Abbreviations 2](#_Toc18333162)

[2.4. References and Related Documents 3](#_Toc18333163)

[2.5. Stakeholder Responsibility Matrix 3](#_Toc18333164)

[2.6. Scope 4](#_Toc18333165)

[2.6.1. Inclusions 4](#_Toc18333166)

[2.6.2. Exclusions 4](#_Toc18333167)

[3. Requirements Artefacts 5](#_Toc18333168)

[3.1. Requirement Types 6](#_Toc18333169)

[3.1.1. Requirement Artefact / Requirement Type Matrix 8](#_Toc18333170)

[4. Waterfall Requirements Management Best Practice 10](#_Toc18333171)

[4.1. Quality 10](#_Toc18333172)

[4.2. Requirement Attributes 10](#_Toc18333173)

[4.3. Requirement Traceability 11](#_Toc18333174)

[4.3.1. Requirement Trace Diagram 11](#_Toc18333175)

[4.3.2. Requirement Trace Relationships 12](#_Toc18333176)

[4.4. Requirement Baselines 12](#_Toc18333177)

[4.4.1. Baseline Strategy 13](#_Toc18333178)

[4.5. Requirement Change Management 13](#_Toc18333179)

[5. Requirement Process 14](#_Toc18333180)

[5.1. Process 14](#_Toc18333181)

[5.1.1. Evaluate Project Opportunities 14](#_Toc18333182)

[5.1.2. Select and Approve Project 14](#_Toc18333183)

[5.1.3. Update Program Roadmap 15](#_Toc18333184)

[5.1.4. Initiate Project 15](#_Toc18333185)

[5.1.5. Plan Analysis Activities 15](#_Toc18333186)

[5.1.6. Map Business Processes 15](#_Toc18333187)

[5.1.7. Specify Business Scenarios 15](#_Toc18333188)

[5.1.8. Elicit Business Requirements 15](#_Toc18333189)

[5.1.9. Derive Functional and Non-Functional Requirements 15](#_Toc18333190)

[5.1.10. Model Information Requirements 16](#_Toc18333191)

[5.1.11. Define Technical Architecture 16](#_Toc18333192)

[6. Requirements Change Management Process 17](#_Toc18333193)

[6.1. Requirement Change Management Process Diagram 17](#_Toc18333194)

[6.2. Requirement Change Management Process Description 17](#_Toc18333195)

[6.2.1. Submit Change Request 17](#_Toc18333196)

[6.2.2. Log Change Request 17](#_Toc18333197)

[6.2.3. Identify Requirement Owner 18](#_Toc18333198)

[6.2.4. Assign Change Request 18](#_Toc18333199)

[6.2.5. Assess Change Impact 18](#_Toc18333200)

[6.2.6. Review Recommendation 18](#_Toc18333201)

[6.2.7. Escalate Change Request 18](#_Toc18333202)

[6.2.8. Log Decision 18](#_Toc18333203)

[6.2.9. Notify Requestor of Outcome 19](#_Toc18333204)

[7. Agile Approach to Requirements Management 20](#_Toc18333205)

[7.1. Agile Requirement Attributes 20](#_Toc18333206)

[7.2. Agile Approaches & Techniques 20](#_Toc18333207)

[7.3. Quality in User Stories 22](#_Toc18333208)

[7.4. Agile Requirement Traceability 23](#_Toc18333209)

[7.4.1. Product Roadmap 23](#_Toc18333210)

[7.4.2. User Stories 24](#_Toc18333211)

[7.5. Requirement Backlog 26](#_Toc18333212)

[7.6. Requirement Change Management 26](#_Toc18333213)

[8. Hybrid Approach to Requirements Management 27](#_Toc18333214)

[9. Requirement Management Toolset 28](#_Toc18333215)

[9.1. Toolset 28](#_Toc18333216)

[9.2. Owner 28](#_Toc18333217)

[9.3. Administration 28](#_Toc18333218)

[9.4. Licensing Arrangements 28](#_Toc18333219)

[9.5. Installation and user guide 28](#_Toc18333220)

[10. Appendix A – Agile Technique Short Descriptions 30](#_Toc18333221)

[11. Appendix B – 32](#_Toc18333222)

Table of Figures

Figure 1 - Requirement Traceability Diagram 12

Figure 2 - Requirement Trace Relationships 12

Figure 3 - Requirement Process 14

Figure 4 - Requirement Change Management 17

Figure 5 – Selecting the Right Technique 19

Figure 6 – Story Decomposition Diagram 21

Figure 7 – Agile Traceability Diagram – High Level 22

Figure 8 – Product Roadmap 23

Table of Tables

Table 1 - Acronyms 2

Table 2 - Terminology 2

Table 3 - References 3

Table 4 - Stakeholder Responsibility Matrix (RACI Matrix) 3

Table 5 - Requirements Artefacts 5

Table 6 - Requirement Types 8

Table 7 - Requirement Artefact / Requirement Type & Technique Matrix 8

1. Executive Summary

*The executive summary provides an overview of the content contained in the Requirements Management Plan document. The executive summary should not say anything that has not been included in the body of the template. Many people write this section last i.e. after the rest of the template has been completed. Items that typically need to be in this summary include:*

* *A brief description of the problem or opportunity to be addressed and the intended audience,*
* *An outline of the scope,*
* *An outline of the project objectives and benefits sought,*
* *Key risks and issues,*
* *A summary of the time, costs and resources to complete the next stage,*
* *A summary of the recommended course of action.*

1. Introduction
   1. Purpose

The purpose of this requirements management plan (RMP) is to specify the process, information, people and tools that have been agreed by the various requirements stakeholders within <<Organisation Name>> to manage requirements for the <<Organisation Name>> program(s) of work.

Note:

It is important that this plan be maintained with the same urgency as other project planning since it is the single “source of truth” for the requirements stakeholders within <<Organisation Name>>.

* 1. Intended audience

This is an internal document and is relevant to the following requirement stakeholders:

Business Representative (BR);

Program Director (PD);

Project Manager (PM);

Agile Product Owner (PO);

Business Analysts (BA);

Technical Architects (TA);

Subject Matter Experts (SME);

Solution Developers (SD)

Testers (TS);

Change Management Analysts (CM);

Quality Assurance (QA).

* 1. Definitions, Acronyms and Abbreviations

Table of terms and associated descriptions used within this document are contained in the following table.

Table 1 - Acronyms

|  |  |
| --- | --- |
| Acronym | Definition |
| RMP | Requirements Management Plan |
| PMO | Project Management Office |
| BC | Business Case |
| BS | Business Scenario |
| SRS | Stakeholder Requirements Specification |
| FRS | Functional Requirements Specification |

Table 2 - Terminology

|  |  |
| --- | --- |
| Terminology | Definition |
| Traceability | The ability to trace project elements especially those related to requirements to provide the ability to manage project element associations and dependencies. |
| Use Case | Describes the interaction between one or more actors (people and other computer systems) and the system. |
| Business Scenario | A description of the activities performed by one or more people to achieve a specific business outcome (e.g. Major plant maintenance) |
| Story Board | A technique / artefact for understanding how people will actually use the solution. Used to detail visually and textually the sequence of activities summing up different user interactions with the solution. |

* 1. References and Related Documents

Table of references and related documents used within this document are contained in the following table.

Table 3 - References

|  |  |  |
| --- | --- | --- |
| Document Type | Document Name | Location |
|  |  |  |

* 1. Stakeholder Responsibility Matrix

Stakeholder responsibilities can be represented in terms of the artefacts (products) that they have an involvement or interest in. The representation used in this document is a RACI Matrix, which stands for Responsible, Accountable, Contributed and Informed.

The following table lists the primary project stakeholders and their responsibilities in relation to the project artefacts that they have an interest in.

Table 4 - Stakeholder Responsibility Matrix (RACI Matrix)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Stakeholders | Products | | | | |
| **BC** | **BS** | **RMP** | **BRS** | **DRS** |
| Business Representative (BR) | I | I | I | I | I |
| Program Director (PD) | I | I | I | I | I |
| Project Manager (PM) | A | A | A | A | A |
| Agile Product Owner (PO) | A | A | A | A | A |
| Business Analysts (BA) | R | R | R | R | R |
| Technical Architects (TA) | I | I | I | I | C |
| Subject Matter Experts (SME) | C | C | I | C | C |
| Solution Developers (SD) | I | I | I | I | I |
| Testers (TS) | I | I | C | I | I |
| Change Management Analysts (CM) | I | I | I | I | I |
| Quality Assurance (QA) | I | I | C | I | I |

* 1. Scope
     1. Inclusions

The scope of this RMP includes the processes and related artefacts that the business analyst and subject matter experts produce.

* + 1. Exclusions

The following processes and products are not in scope for this version of the RMP:

Technical and Solution Architecture specifications;

Implementation activities and specifications;

Testing processes and specifications.

1. Requirements Artefacts

The artefacts to be produced by the business analysts for <<Project Name>> are listed in the table below. This table includes the name, description and the reason for producing the artefact; a full list of artefacts is also featured in section 3.4 - Requirement artefacts of this document.

*Please remove artefacts from the table as appropriate, note full list repeated in later section*

Table 5 - Requirements Artefacts

|  |  |  |
| --- | --- | --- |
| Artefact Name | Description | Purpose |
| Business Case (BC) | A decision support and planning tool that quantifies the expected financial results and other business consequences and impacts based on a particular action. | Inform the enterprise decision makers about a proposed project and provide sufficient information to facilitate a decision on whether or not to invest. |
| Stakeholder Requirements Specification (StRS)  Also commonly called Business Requirements Specification (BRS) | Details the roles, policies and key business processes for an initiative. The document first describes the “as is” process highlighting the issues with this current state and the opportunities for improvement. The “to be” processes include the suggested process changes required to realise these opportunities. These “to be” processes provide the context for the business requirements.  Documents the business needs as defined by the agreed business stakeholders (i.e. managers, policy makers and reviewers) | To provide the team with an understanding of the business processes that provides the context for the business requirements.  To specify the high-level requirements that allow:  The stakeholder to agree the minimum set of business requirements;  The team to create functional and non-functional requirements. |
| Solution Requirement Specification (FRS)  Also commonly called Functional Requirements Specification (FRS) | Describes the functional, non-functional and information requirements for an initiative. NB these requirements are a lower level of requirements geared towards a technical audience. | To specify the detailed requirements that allow:  The team to design and build their service components;  A software vendor to specify the detailed design required to build / modify a solution;  To evaluate whether or not a software vendor’s product meets our requirements. |
| Business Scenarios (BS) | Describe a “real-life” sequence of events and activities for a business. Normally evokes multiple business processes. | To validate the business processes from a business perspective. |
| Report Specification (RS) | Describes the report layouts for both operational and strategic reporting. | To specify the detailed requirements that allow:  The team to design and build their service components;  A software vendor to specify the detailed design required to build / modify a solution;  To evaluate whether or not a software vendor’s product meets our requirements |
| User Stories (US) | User Stories capture the needs of a specific stakeholder and enable teams to define features of value to a stakeholder using short, simple documentation. | To specify the High-Level requirements that allow:  The team to determine what functionality is required to meet the need;  What acceptance criteria is associated with the requirement;  A software vendor to specify the detailed design required to build / modify a solution;  A Product Owner to determine what pieces of functionality can be implemented within a sprint. |
| Feature | A Feature is a distinctive and prominent attribute of a solution. It is a piece of functionality that delivers business value and sits in the work item hierarchy between Goals and Epics, eg: Objective -> Goal -> Feature -> Epic -> Story -> Task | Used to distinguish it from other features within the solution;  Important; something that “generally” the solution cannot do without;  A technical or non-technical part that gives value to stakeholders or users;  Something that solves, or is a method for solving, a problem or achieves a goal. |

* 1. Requirement Types

The following requirement types are used by <<Organisation Name>>:

Business Requirements;

Stakeholder Requirements;

Solution Requirements;

* + Functional Requirements
  + Non-Functional Requirements

Transition Requirements.

In addition to these requirement types the following specification and modelling techniques are also used (among others) in the requirements modelling process:

Business Capability Analysis;

Business Model Canvas;

Data Flow / Data Modelling;

Functional Decomposition;

Glossary;

Interface Analysis;

Process Modelling;

Sequence Diagrams;

Stakeholder List, Map or Personas;

Use Case;

User Stories.

Instances of each of the requirement types can be created for one or more projects. This provides for reuse and consistency of requirements definition across multiple <<Organisation Name>> programs and projects.

Table 6 - Requirement Types

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement Type | Description | Identifier Code | Additional Attribute(s) |
| Business Requirements | Statements of goals, objectives, and outcomes that describe why a change has been initiated. They can apply to the whole of an enterprise, a business area, or a specific initiative. | BR | None |
| Stakeholder Requirements | Describe the needs of stakeholders that must be met in order to achieve the business requirements. They may serve as a bridge between business and solution requirements. | StR | None |
| Functional Requirement (Solution Requirement) | Describe the capabilities that a solution must have in terms of the behaviour and information that the solution will manage. | FR | None |
| Non-Functional Requirement (Solution Requirement) | Do not relate directly to the behaviours of functionality of the solution, but rather describe conditions under which a solution must remain effective or qualities that a solution must have. | NFR | None |
| Transition Requirements | Describe the capabilities that the solution must have and the conditions the solution must meet to facilitate transition from the current state to the future state, but which are not needed once the change is complete. They are differentiated from other requirements types because they are of a temporary nature. Transition requirements address topics such as data conversion, training, and business continuity. | TR | None |

* + 1. Requirement Artefact / Requirement Type Matrix

The Requirement artefact / requirement type matrix shows which requirement types are contained in each of the artefacts produced by the business analysts.

Table 7 - Requirement Artefact / Requirement Type & Technique Matrix

|  |  |  |
| --- | --- | --- |
| Artefact | Requirement Type(s) | Techniques |
| Business Case (BC) | Business Requirement BR) |  |
| Stakeholder Requirement Specification (StRS) | Business Process (BP)  Stakeholder Requirements (StR) |  |
| Solution Requirement Specification (SRS) | Functional Requirement (FR)  Non-Functional Requirement (NR) | Use Case |
| Business Scenarios (BS) | Stakeholder Requirements (StR) | Business Scenarios  Job Stories |
| Story Board | User Stories (US) | User Stories  Story Cards |
| Feature | User Stories (US) | Wireframing |

1. Waterfall Requirements Management Best Practice

This section describes a number of activities associated with requirements management “Best Practice”.

* 1. Quality

Using a number of different requirements types allows an organisation to classify requirements for different requirement stakeholder groups (e.g. business users, developers, testers). No matter what the requirement type, the requirement must be able to be validated to ensure quality and agreement with the requirement stakeholders. To promote quality and facilitate this stakeholder agreement a requirement must be:

**Atomic** – self-contained and capable of being understood independently of other requirements or designs.

**Complete** – enough to guide further work and at the appropriate level of detail for work to continue. The level of completeness required differs based on perspective or methodology, as well as the point in the life cycle where the requirement is being examined or represented.

**Consistent** – aligned with the identified needs of the stakeholders and not conflicting with other requirements.

**Concise** – contains no extraneous and unnecessary content.

**Feasible** – reasonable and possible within the agreed-upon risk, schedule, and budget, or considered feasible enough to investigate further through experiments or prototypes.

**Unambiguous** – the requirement must be clearly stated in such a way to make it clear whether a solution does or does not meet the associated need.

**Testable** – able to verify that the requirement or design has been fulfilled. Acceptable levels of verifying fulfillment depend on the level of abstraction of the requirement or design.

**Prioritized** – ranked, grouped, or negotiated in terms of importance and value against all other requirements.

**Understandable** – represented using common terminology of the audience.

All requirements documents should be kept under version control remembering that there is a version number for the requirements document (collection of requirements) and an identifier and a version number for each requirement. When a requirements document is used to baseline a set of requirements, it is important to follow a consistent naming convention.

* 1. Requirement Attributes

Best practices related to requirement types, state that although each requirement type has specific information that needs to be captured (requirement type attributes). The following attributes will exist for each of the requirement types and model elements listed in the previous section:

* **Requirement Identifier** – A unique identifier for a requirement. The identifier is an alphanumeric code comprising of a two-character alpha abbreviation of the requirement type and a sequence number commencing at 1 (e.g. BR001, UC001)
* **Requirement Name** – The actual wording of a requirement
* **Requirement Version Number** – A number that indicates the current version of a requirement
* **Requirement Type** – A valid requirement type. For a list of valid requirement types refer to Table 1 – Requirement types.
* **Requirement Description** – Text describing additional information related to a requirement
* **Requirement Change History** – A list of the changes made to a requirement. Includes:
  + A brief description of the change
  + The identity of the user that made the change
  + The date and time the change was made
* **Requirement Status** – The current status of a requirement within the context of the lifecycle of a requirement. NB this requirement lifecycle is aligned with the requirement change management process. The valid status values for a requirement are: Created; Peer Reviewed; Business Reviewed; Approved; and Rejected.
* **Requirement Priority** – The importance of a requirement within the context of a project. The valid priority values for a requirement are: Desirable; Highly Desirable; and Business Critical.
* **Requirement Owner** – The user-id or name of the person that is responsible for and maintains any changes to a requirement.
* **Requirement Trace Information** – Either a text-based or graphic view of requirements of either the same of different type that the requirement is traced to or from.

When a set of requirements is collected in a document, each requirement needs to be identified by at least:

* Requirement Identifier;
* Requirement Version Number.

All requirements documents should be kept under version control remembering that there is a version number for the requirements document (collection of requirements) and an identifier and a version number for each requirement. When a requirements document is used to baseline a set of requirements, it is important to follow a consistent naming convention.

* 1. Requirement Traceability

A requirement trace defines an association between two or more requirement types. In fact many relationships exist between requirements and between requirements and other system development artefacts. Traces are maintained between requirements during planning, analysis, development, testing and implementation. These trace relationships are critical to:

Manage the impact of change (e.g. identifying dependencies);

Ensure completeness;

Ensure there is justification for the development.

Terms used to define the traces are as follows:

Should Trace - a relationship is supposed to exist, but may not (NB if the trace does not exist the requirement is said to be “suspect”);

Must Trace - a relationship has to exist for the requirement to be valid;

May Trace - a relationship can exist but, if one doesn’t, the requirement is still valid.

The final consideration when applying a trace between requirement types is the direction of the trace. The two options available are “trace from” and “trace to”. It is important to ensure that the direction of a trace between requirements types is specified so that analysts can consistently apply a trace between requirements in the same direction.

This section describes the traceability between the different types of requirements. It includes a traceability diagram and a description of each of the trace relationships

* + 1. Requirement Trace Diagram

Figure 1 - Requirement Traceability shows the trace relationships between the different requirement types.

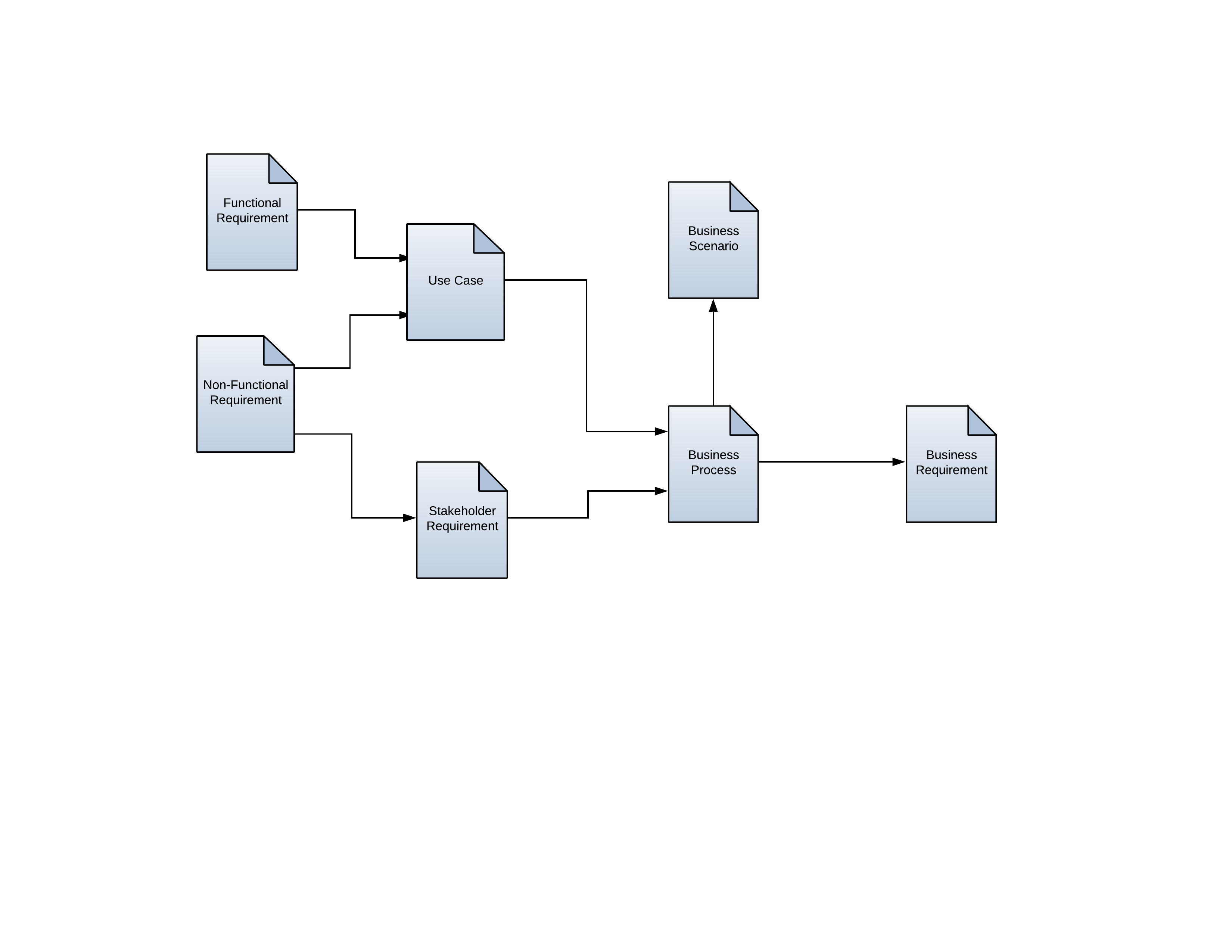


Figure 1 - Requirement Traceability Diagram

* + 1. Requirement Trace Relationships

The following table lists each requirement type and the trace relationship(s) with other types of requirements as depicted in the requirement trace diagram above.

Figure 2 - Requirement Trace Relationships

|  |  |  |
| --- | --- | --- |
| Requirement Type | Tag | Trace Relationship(s) |
| Stakeholder Requirement | StR | Each Stakeholder Requirement *should trace to* one or more Business Process |
| Functional Requirement | FR | Each Functional Requirement *must trace to* one or more Use Case |
| Non-Functional Requirement | NR | Each Non-Functional Requirement *should trace to* one or more Stakeholder Requirement  Each Non-Functional Requirement *should trace to* one or more Use Case |
| Business Requirement | BR | Each Business Process *should trace to* one or more Business Requirement |
| Transition Requirement | TR | Each Transition Requirement *should trace to* one or more Business Process. |

* 1. Requirement Baselines

A requirements baseline consists of taking a 'snapshot' of the requirements (either the complete set of requirements or just a subset) as they exist at a particular moment in time. To be included in a ‘snapshot’, a requirement should have the minimum set of attributes defined. This “snapshot” can establish an informal or formal baseline. An informal baseline is an agreement by the project team that the requirements in the ‘snapshot’ are correct as of that point in time. This does not require an official sign-off but from this point on any changes to a requirement in the informal baseline need to be documented in the requirement change history log. It is recommended that informal baselines be created at several points in the requirements process.

A ‘snapshot’ to create a formal baseline should occur at three points in the requirements process. The first point is prior to development of the software to implement the requirements, the second point prior to the requirements and software going to validation (testing) and the third point when validation is complete. Once a formal baseline is created any changes to a requirement must be handled according to the formal project’s procedure for managing change.

* + 1. Baseline Strategy

The proposed strategy is to baseline requirements at the point each artefact is approved and signed-off. The three major artefacts produced in the requirements process presented in section 8 are:

Business Case;

Business Requirements Specification;

Functional Requirements Specification.

As each of these artefacts is approved the version of the requirements contained in these artefacts should be baselined. NB:

Each artefact may contain more than one type of requirement (e.g. DRS contains functional and non-functional requirements);

The traceability between the different requirements types should also be baselined. If the trace between two different requirements changes this is considered a change to a requirement (i.e. the version of a requirement changes).

* 1. Requirement Change Management

Even a well-defined set of requirements tasks with effective elicitation/analysis techniques will not produce satisfactory results (or any result at all) if changes are not managed. This change management must consider the types of requests for change;

Need to handle ‘demands’ for change (critical/urgent issues which will often step outside the defined procedures);

Need to find root cause of a defect so that the link back to the source requirements can be maintained;

Fact that a ‘Request for Change’ does not equal a Requirement.

Therefore, it is essential that a formal change management procedure is followed when managing changes to requirements that have been ‘baselined’.

Change management process is explained in detail in section 6 - Requirements Change Management Process incorporating process diagram and description.

1. Requirement Process
   1. Process

The process map shown below depicts the activities performed and artefacts delivered by the different roles of the participants in the requirements space of the solution delivery life cycle. This process can be modified to focus on the roles and activities relevant for the <<Organisation Name>> requirements process.

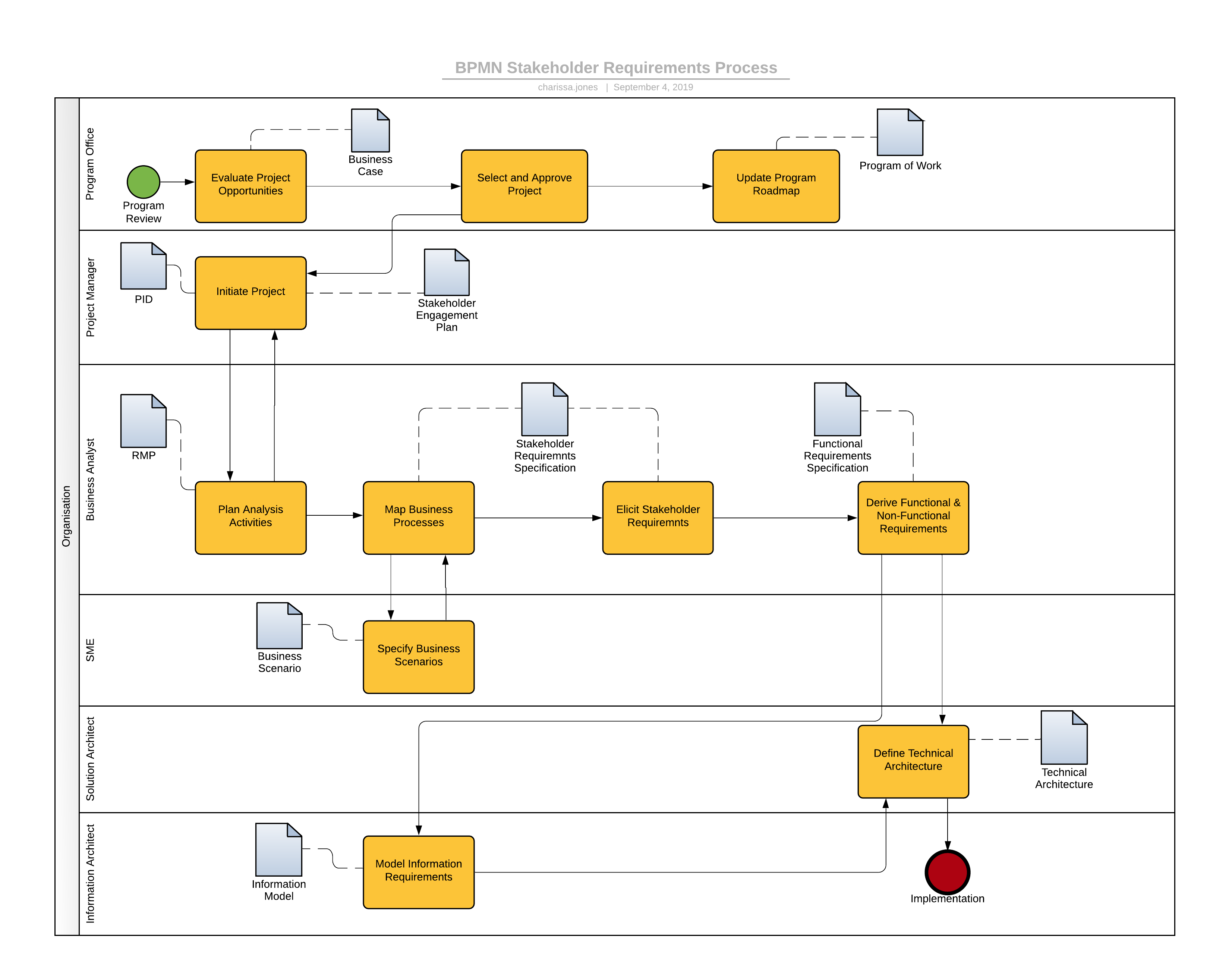


Figure 3 - Requirement Process

* + 1. Evaluate Project Opportunities

The organisation should require each program and the projects in the programs to identify and document the Business Requirements that are to be achieved. These Business Requirements are typically documented in a business case along with the proposed benefits. These requirements are recorded so that full requirements validation and verification can be completed.

* + 1. Select and Approve Project

The process of reviewing the business case and subsequent selection of a project can vary from organisation to organisation depending on the methodology and formality adopted by the organisation, but is shown here as a prompt for further consideration when completing an RMP for an organisation.

* + 1. Update Program Roadmap

Once the project has been approved the program of work will need to be updated to reflect this. Note the completion of this activity is not mandatory for the project to be initiated

* + 1. Initiate Project

The Project Manager performs a number of project initiation and planning activities, which can result in several artefacts being produced. The exact type of artefacts will again depend on the project and program methodologies being used. The Project Initiation Document (PID), a Prince2 product is shown in this process example. The project manager will also work closely with the Business Analyst to understand the business analysis tasks required for the project as well as the stakeholders. These stakeholders and the method of communicating with them should be documented.

* + 1. Plan Analysis Activities

The Business Analyst will identify the analysis activities (including requirement artefacts, requirement stakeholders, requirements management and communication processes and tools) required for the project. This information is recorded in the RMP and is done in close collaboration with the Project Manager.

* + 1. Map Business Processes

The Business Analyst models the business processes for several reasons. Firstly, to understand how the business stakeholders currently operate and what issues and opportunities for improvement exist. Secondly to provide a starting point for the business processes that will be required to support the business with the implementation of a new IT solution or business process transition initiative. In addition to the process models developed, the business analyst can also identify the Stakeholder Requirements for specific activities documented within a process. These Stakeholder Requirements describe the stakeholder needs that the solution should satisfy.

* + 1. Specify Business Scenarios

The SMEs develop scenarios, which describe the major business workflows performed. These scenarios are written from a business not Information Technology (IT) perspective. The scenarios may make reference to IT systems but their focus is to describe what a business stakeholder does in a business situation to derive a specific business outcome. The purpose of these scenarios is to provide a mechanism to validate the business process models, Stakeholder Requirements and Functional Requirements.

* + 1. Elicit Business Requirements

The Business Analyst uses the business scenarios to map across the business processes to ensure that they are valid from a business perspective. This is important because these processes will be used as basis for change management activities, which will involve the business stakeholders and support staff.

* + 1. Derive Functional and Non-Functional Requirements

The Business Analyst further investigates and analyses the issues and opportunities identified in the Stakeholder Requirements Specification. The purpose of this activity is to specify the Functional and Non-Functional Requirements required to design and build or procure an appropriate solution for the business. In the case of a procurement project these requirements could be used to go to the market with a tender (e.g. RFP).

* + 1. Model Information Requirements

Depending on the complexity of the information requirements for a project, this activity may be performed by a Business Analyst with information modelling experience or require the expertise of an information architect. Typically, these requirements will be modelled using notations such as the Unified Modelling Language (UML).

* + 1. Define Technical Architecture

The Technical Architect defines the technological constraints and aspects for the solution based on the business processes, Functional, Non-functional, and information requirements.

1. Requirements Change Management Process

This section describes how to manage changes to requirements included in a requirements baseline. Changes to ‘baselined’ requirements are managed using the <<Organisation Name>> Requirements Change process.

* 1. Requirement Change Management Process Diagram

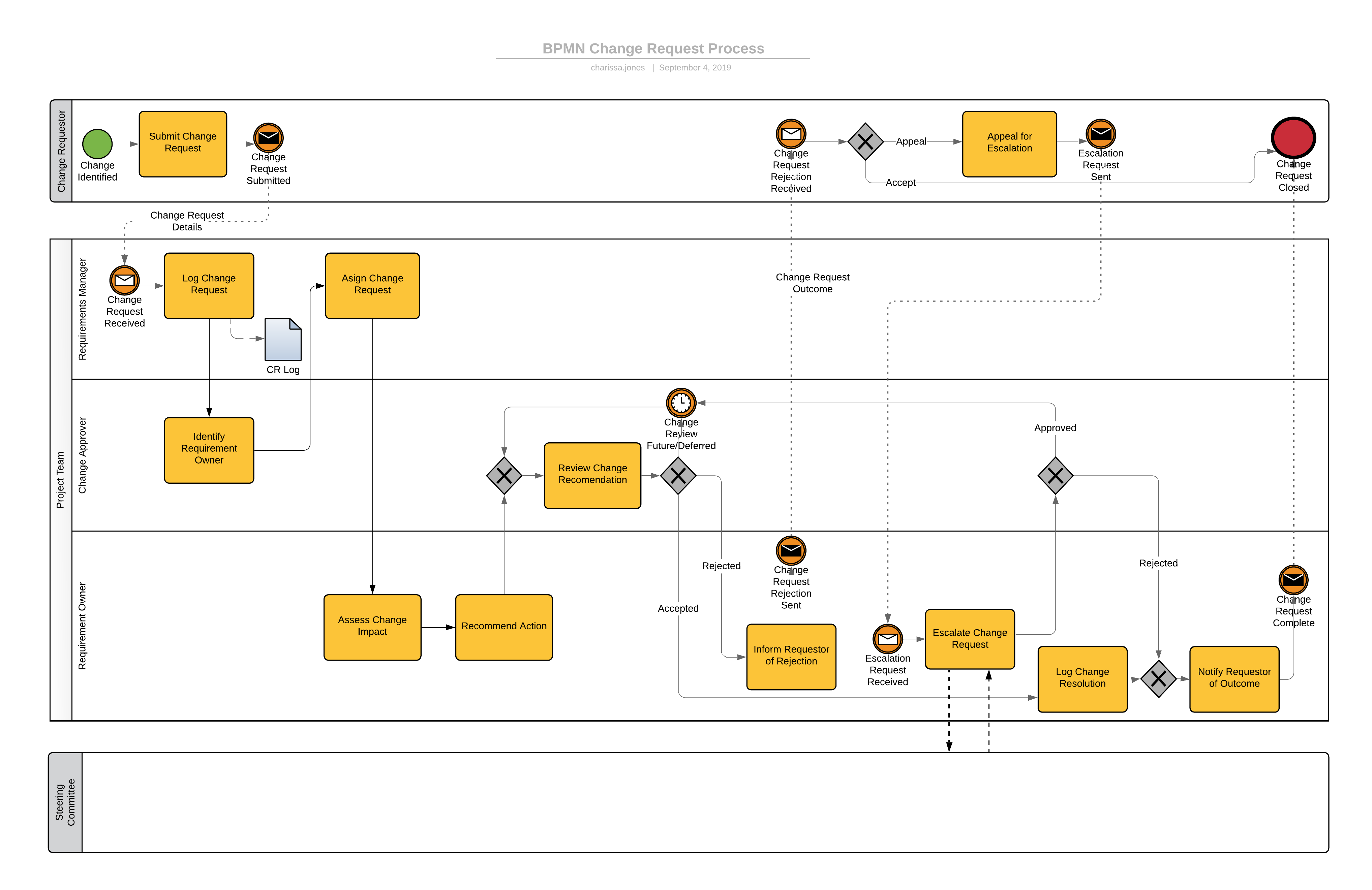


Figure 4 - Requirement Change Management

* 1. Requirement Change Management Process Description
     1. Submit Change Request

The person requesting the change completes the information required for the change request and submits this information to the Requirements Manager.

* + 1. Log Change Request

The Requirements Manager records the change request details and assigns a unique identifier. If there is any missing information the Requirements Manager will contact the person who lodged the change request and ask them for the missing details. The completed change request status is set to “Open” and is forwarded to the Change Approver.

* + 1. Identify Requirement Owner

The Change Approver will identify the owner of the requirement/business process and determine the resources required to conduct an assessment of the change request. NB the change request may involve more than one requirement.

* + 1. Assign Change Request

The Requirements Manager assigns the change request to the nominated Requirements Owner and set the change request status to “Evaluating”.

* + 1. Assess Change Impact

The Requirements Owner conducts an impact assessment of the change request.

* + 1. Recommend Action

The Requirements Owner considers possible solutions and makes a recommendation based on the impact assessment to the Change Approver.

* + 1. Review Change Recommendation

The Change Approver reviews the recommendation and makes one of the following decisions to resolve the change request:

Accept the recommendation;

Reject the recommendation;

Defer a decision until a future review cycle.

Recommendations that are deferred will be reviewed by the Change Approver at a future change request review cycle.

* + 1. Inform Requestor of Rejection

If the Change Approver has rejected the recommendation; the change request reverts to the Change Requestor to make one of the following decisions to resolve the change request:

Accept the rejection;

Appeal for Escalation.

* + 1. Escalate Change Request

In the situation where the Change Approver rejects a recommendation from the Requirements Owner the Requirements Owner can escalate the decision for review. This may require the Requirement Owner to provide additional information and justification. The result of this escalation step is that the change recommendation is either approved or rejected.

* + 1. Log Change Resolution

The Requirements Owner records the outcome of the change request review and sets the status of the change request to ‘Closed’.

* + 1. Notify Requestor of Outcome

The Requirements Owner notifies the change Requestor of the outcome of the change request.

1. Agile Approach to Requirements Management

Agile is an umbrella term for a variety of approaches. The primary characteristic of any agile approach is its alignment to the values and principle of the Agile Manifesto. An agile team may implement or evolve to use a combination of approaches which enables them to deliver value more effectively given their project type and work environment. This section describes a Disciplined Agile Delivery approach to project and initiative requirements management.

* 1. Agile Requirement Attributes

In an Agile environment, a Business Analyst is often presented with the desired end-product/solution first and then must determine what is required to achieve this solution. This differs from Waterfall requirements in that Waterfall Requirements begin as a representation of a business need and continue through elicitation and prioritisation until a solution is found that meets the total requirements whilst Agile Requirements begin as a desired solution and work backwards to determine how a system should behave in relation to specific stakeholders needs. These stakeholder needs are presented as User Stories.

* 1. Agile Approaches & Techniques

Whilst Agile projects generally have just enough to specify a high-level outline of the project at the outset, with details being added along the way; there is still a need for initial requirements capture and analysis to define the project vision and scope.

In the agile context, requirements will initially be gathered at the Strategy Horizon and will include User Stories and Epics. The User Stories and Epics are entered into a Product Backlog and then each Story goes through a process of Story Decomposition to define the various elements of each requirement and what acceptance criteria must be applied.

User Stories, Epics and Story Decomposition are all referred to as Techniques and are just a few of the many Agile Techniques available to a Business Analyst during the project lifecycle. The many Agile Techniques can be managed as requirements and will offer greater value and usability if they can be traced to related project artefacts.

Some techniques commonly used by agile business analysis practitioners and in which context they are applied is outlined in the table below:

Figure 5 – Selecting the Right Technique

|  |  |  |
| --- | --- | --- |
| Context | Works well with internal teams | Works well with stakeholders external to the team |
| Communications | * Backlog Refinement * Planning Workshops * Portfolio Kanban * Retrospectives * Visioning | * Reviews * Visioning |
| Process Analysis | * Value Stream Mapping | * Impact Mapping |
| Product Management Refinement | * Minimal Viable Product * Product Roadmap * Purpose Alignment Model * Real Options | * Kano Analysis * Minimal Viable Product |
| Requirements Management | * Behaviour Driven Development * Job Stories * Relative Estimation * Spikes * Story Decomposition * Story Elaboration * Story Mapping * User Stores |  |
| Understanding your Customer | * Personas * Storyboarding * Value Modelling |  |

The most commonly used tools and techniques by Agile Business Analysts are:

**Backlog Management** – used to track initiatives across the portfolio of work and also to understand critical risks, integration needs, and dependencies between initiatives. Used consistently during the Delivery Horizon; backlog management is a frequent if not constant activity;

**User Stories** – used to convey a customer requirement or need for the delivery team. They are small, concise statements of features need to deliver value. A User Story is captured on the backlog;

**Epics** – used to group user stories and requirements that have a common objective;

**Story Decomposition** – provides a structure for defining the various elements of requirements at progressively smaller levels of granularity, starting with the broad system context and drilling down in multiple levels to eventually define the detailed acceptance criteria for individual stories.

**Features** – A piece of functionality that is considered essential to the proposed solution and meets one/more business goals.

Figure 4 – Story Decomposition Diagram shows the trace relationships between the different requirement types

A close up of a sign

Description automatically generated

Figure 6 – Story Decomposition Diagram

A full list of the techniques listed in Figure 3 and their short descriptions can be found in Appendix A.

* 1. Quality in User Stories

Using a number of different techniques to elicit requirements allows an analyst or organisation to classify requirements for different requirement stakeholder groups (e.g. business users, developers, testers). No matter what the requirement type, the requirement must be able to be validated to ensure quality and agreement with the requirement stakeholders.

A commonly used construct for ensuring quality in user stories is the **INVEST** criteria, which calls for user stories to be;

**(I) Independent** – represents a feature which can be delivered independent of other features.

* + Example: “ATM PIN entry” is independent from ‘Withdrawal Amount’

**(N) Negotiable** – the team can negotiate how to deliver;

**(V) Valuable** – expresses the value to the customer.

* + Example: “ATM PIN entry” allows only the correct person to access the account.;

**(E) Estimable** – team can estimate effort to deliver based on past experience;

**(S) Sized Appropriately** – for the team to complete in one iteration. In general, the smaller the better;

**(T) Testable** – can be validated objectively by a stakeholder.

Not all backlog items are to be written as user stories however; User Stories is a common technique used as they emphasize the customer value.

* 1. Agile Requirement Traceability

It is critical that requirement and traceability information can be captured at the same time as design artefacts are created. The engineering activities in an agile development are not linear; to deliver working increments of a system or software requires constant iteration across all lifecycle processes. Agile requirements management must therefore be tightly integrated with other engineering activities and their associated tools, including architecture, design, testing, change and configuration management, and workflow planning and management.

In constant and rapidly changing environments, organisations are required to be able to sense and respond to local opportunities and problems without the need to involve the whole organisation, while also looking forward at emerging threats and opportunities. These planning horizons provide a framework for the shift in focus that occurs when moving between understanding the long-term strategic needs of the organisation and the immediate needs of a customer.

There are a number of really good software products; such as Jira - which track traceability inherently however; by utilising a few different Techniques an Agile Business Analyst can also effectively document Requirement Traceability. An example of these are outlined below.

A close up of a logo

Description automatically generated

Figure 7 – Agile Traceability Diagram – High Level

* + 1. Product Roadmap

**Purpose:**

Product Roadmap is used to communicate direction and progress towards the vision for a solution or initiative, and it measures progress against that vision through achieving the stakeholders’ desired outcome

**Description:**

Product Roadmap is a strategic document and plan used to describe how a product is likely to grow, to align to stakeholders’ needs, and to acquire a budget for delivery. It shows features, requirements, or initiatives, and outlines a path to deliver them over time.

Agile values working solutions; product roadmaps focus on product/feature/value delivered, no milestones or check points.

Product Roadmap enables iterative delivery by expressing features in terms of now, next, and later. It defines what the solution is and what it is not.

A screenshot of a cell phone

Description automatically generated

Figure 8 – Product Roadmap

**Elements:**

1. **Defined Vision and Strategy** – Clearly defines the vision and strategy for the initiative. This vision clarifies what is included for the solution and the goal to be achieved. The roadmap also articulates how that vision will be achieved;
2. **Defined Desired Outcomes** – Clearly articulates organisation and stakeholder desired outcomes. Defining the desired outcomes helps the delivery team provide a working solution that adds value;
3. **Product Management Team** – Led by the product owner or customer representative. This is a small team focused on maintaining the Product Roadmap. The product management team ensures the roadmap reflects the most current priorities and goals, is accessible to those who need it, and tailors the view based on the audience as needed;
4. **Themes** – includes themes which represent a collection of requirements, features, or stories;
5. **High-level Requirements** – Comprised of high-level requirements or features which are expected to deliver value to achieve the vision and goals for the solution. These high-level items represent a group of requirements or stories.

**Traceability Elements**:

* Each Epic should align directly with a High-level Requirement component of the Product Roadmap
* When writing User Stories – the WHY component should be easily traceable to the Product Roadmap Defined Desired Outcomes and the detailed acceptance criteria.
* After each Scrum Sprint (development run) – the result should trace back to a Capability outlined in the Product Roadmap and achieve a component of the Defined Desired Outcomes.
  + 1. User Stories

**Purpose:**

User Stories are used to convey a customer requirement for the delivery team. A user story represents a small, concise statement of functionality or quality needed to deliver value to a specific stakeholder.

**Description:**

User stories capture the needs of a specific stakeholder and enable teams to define features of value to a stakeholder using short, simple documentation. They can serve as a basis for identifying needs and allow for the prioritising, estimating, and planning of solutions. A user story is typically a sentence or two that describes who has the need addressed by the story, the goal the user is trying to accomplish, and any additional information that may be critical to understanding the scope of the story. With a focus on stakeholder value, user stories invite exploration of the requirements by promoting additional conversations with stakeholders and grouping functional requirements for delivery.

User stories can be used:

* To capture stakeholder needs and prioritize development of solutions
* As a basis of estimating and planning solution delivery,
* As a basis for generating use acceptance tests,
* As a metric for measuring the delivery of value,
* As a unit for tracing related requirements,
* As a basis for additional analysis, and
* As a unit of project management and reporting.

**Elements:**

1. **Title** (optional) – The title of the story describes an activity the stakeholder wants to carry out with the system. Typically, it is an active-verb goal phrase similar to the way use cases are titled;
2. **Statement of Value** – There is no mandatory structure for user stories. The most popular format includes three components:
   1. **Who**: a user role or persona
   2. **What**: a necessary action, behaviour, feature, or quality
   3. **Why**: the benefit or value received by the user when the story is implemented.

For example, “As a <who>, I need to <what>, so that <why>.”

1. **Conversation** – User stories help teams to explore and understand the feature described in the story and the value it will deliver to the stakeholder. The story itself doesn’t capture everything there is to know about the stakeholder need and the information in the story is supplemented by further modelling as the story is delivered;
2. **Acceptance Criteria** – A user story may be supported through the development of detailed acceptance criteria. Acceptance criteria define the boundaries of a user story and help the team to understand what the solution needs to provide in order to deliver value for the stakeholders. The most popular format used to determine acceptance criteria is Behaviour Driven Development (BDD):
   1. **Given**: a pre-condition is met
   2. **When**: the trigger that starts the story
   3. **Then**: the result received by the user when the story is implemented.

For example, “Given <pre-condition>, When <trigger>, Then <result>.”

* 1. Requirement Backlog

A requirements backlog consists of taking a 'snapshot' of the requirements (generally displayed in User Stories) as they exist at a particular moment in time. To be included in a ‘snapshot’, a requirement should have the minimum set of attributes defined. This “snapshot” can establish an informal or formal baseline. An informal baseline is an agreement by the project team that the requirements in the ‘snapshot’ are correct as of that point in time. This does not require an official sign-off but from this point on any changes to a requirement in the informal baseline need to be documented in the requirement change history log. It is recommended that informal baselines be created at several points in the requirements process.

A ‘snapshot’ to create a formal baseline should occur at three points in the requirements process. The first point is prior to development of the software to implement the requirements, the second point prior to the requirements and software going to validation (testing) and the third point when validation is complete. Once a formal baseline is created any changes to a requirement must be handled according to the formal project’s procedure for managing change.

* 1. Requirement Change Management

Even a well-defined set of requirements tasks with effective elicitation/analysis techniques will not produce satisfactory results (or any result at all) if changes are not managed. This change management must consider the types of requests for change;

Need to handle ‘demands’ for change (critical/urgent issues which will often step outside the defined procedures);

Need to find root cause of a defect so that the link back to the source requirements can be maintained;

Fact that a ‘Request for Change’ does not equal a Requirement.

Therefore, it is essential that a formal change management procedure is followed when managing changes to requirements that have been ‘baselined’.

Change management process is explained in detail in section 6 - Requirements Change Management Process incorporating process diagram and description.

1. Hybrid Approach to Requirements Management

Another approach to Requirements Management is referred to as the Hybrid Approach. In hybrid, the planning is done using the waterfall approach. The execution and delivery are handled by the Agile method. This hybrid approach makes the planning and project estimation a lot more accurate. At the same time, the team can react to requirement changes and deliver what the stakeholder demands in place of what the team planned.

Hybrid can handle requirement changes and, due to its iterative nature, can deliver products in stages. As soon as the product reaches the minimum viable product, or MVP, it can be rolled-out and the development team can continue on future enhancements.

1. Requirement Management Toolset

This section needs to be completed based on the tool that has been allocated to the project. The following sections provide a guide on what information needs to be completed once the toolset decision has been made.

* 1. Toolset

The team is using <<specific tool>> for requirements modelling and requirements management. Specifically, the following models and requirement types are being captured and managed in <<specific tool>>:

Models

* + Business Process (using BPMN)
  + Use Case (using UML)
  + Information Models (using UML)

Requirements Types

* + Business Requirements
  + Stakeholder Requirements
  + Functional Requirements
  + Non-Functional Requirements
  + Transition Requirements

Requirements management including requirements traceability will be done in <<specific tool>>.

* 1. Owner

*List the name / organisational position and contact details of the nominated owner of the requirements management toolset.*

* 1. Administration

*List the name / organisational position and contact details of the nominated administrator of the requirements management toolset. Also provide relevant references to:*

*tool client, server and database administration documentation;*

*user, user group and project administration*

* 1. Licensing Arrangements

*Provide an overview of the licensing agreement including any product support contacts and process.*

* 1. Installation and user guide

*Provide relevant references to toolset:*

*Installation guide;*

*User guide;*

*Tutorial and training material*

1. Appendix A – Agile Technique Short Descriptions

Short descriptions of the types of techniques used to document and express requirements throughout Agile Projects:

**Backlog Refinement** – used to ensure there is enough detail and clarity for items in the backlog so that the delivery team can complete an iteration. Backlog Refinement uses analysis to get a reasonable gauge of the size, scope, and complexity of each backlog item;

**Planning Workshops** – used to determine what value can be delivered over an agreed time period. They enable customer collaboration and response to changes that result from feedback and learning;

**Portfolio Kanban** – used to manage the implementation of strategic initiatives by increasing visibility into the process, work-in-progress (WIP), decision making criteria, and feedback loops. A Portfolio is a collection of strategic initiatives for an organisation or department to execute in alignment with their business goals.

**Retrospectives** – used to continuously improve by reflecting on what went well, what could be better, and to improve the processes. The retrospective includes the entire team;

**Visioning** – used to determine the desired outcome for an initiative worded in a concise and approachable manner. Visioning creates aspirational guidance that is used to understand if efforts align to desired outcomes and add value;

**Value Stream Mapping** – used to understand the creation of value across the whole customer experience to prioritise, plan and integrate the creation of value and reduction of waste between initiatives across the portfolio. A value stream map is a graphical representation that captures a snapshot of the value stream;

**Minimum Viable Product** – used to prioritise the allocation of resources and to increase the speed of organisational learning. MVP identifies the smallest set of features or requirements to deliver value to stakeholders and satisfy early adopters in the shortest time possible;

**Product Roadmap** – used to communicate the expected future direction of the product and to improve collaboration among teams in different initiatives. It shows features, requirements, or initiatives, and outlines a path to deliver them over time. Also used to support decision making and prioritisation;

**Purpose Alignment Model** – used to assess ideas in the context of customer and business value. The purpose alignment model rates features, processes, products, r capabilities in two dimensions. 1) whether or not the feature creates market differentiation; 2) whether or not the feature is critical for the continued functioning of the organisation;

**Real Options** – used to help determine when to make decision. Real Options has three simple rules 1) options have value; 2) options expire and 3) never commit early unless you know why;

**Behaviour Driven Development** – used to increase value, decrease waste, and increase communication between stakeholders and delivery teams by focusing on the intended customer behaviours for the solution to satisfy customer needs;

**Job Stories** – used to represent a product backlog item (PBI) or requirement in terms of a job to be done by a stakeholder. They add contextual information that can affect how a stakeholder wants a desired feature to be and enables help with making the right implementation decisions;

**Relative Estimation** – used to make future predictions based on past experience, knowledge, complexity, size, and uncertainty required to complete backlog items. Estimates are not part of the solution and are used to guide internal decision making;

**Spikes** – used to time-box research, design, exploration, investigation or prototyping activities in order to understand the effort required to deliver a backlog item or an initiative;

**Story Decomposition** – used to support decisions about which features to delivery, in what order, and how much of the feature needs to be delivered in order to reach the desired outcome;

**Story Elaboration** – used to define the detailed design and acceptance criteria for a story as needed to deliver a working solution. Story Elaboration facilitates the elicitation and communication of the most detailed requirements;

**Story Mapping** – used to assist in creating understanding of product functionality, the flow of usage, and to assist with prioritising product delivery. It uses a two-dimensional grid structure to show sequence and grouping of key aspects of the product on the horizontal dimension, with detail and priority of stories on the vertical dimension.

**Stakeholder List, Map, or Personas** – used to identify the people who need to be involved in decisions.

**User Stories** – used to convey a customer requirement or need for the delivery team. They are small, concise statements of features need to deliver value. A User Story is captured on the backlog;

**Personas** – used to understand and empathize with an intended stakeholder in order to align the solution with the stakeholder need. They are often used in agile approaches to understand value from the perspective of a particular stakeholder and allow a team that may not have direct access to a customer representative to better understand their needs;

**Storyboarding** – used to describe a task, scenario, or story in terms of how stakeholders interact with the solution. Storyboarding is used in conjunction with other techniques such as use cases, user stories and prototyping to detail visually and textually the sequence of activities summing up different user interactions with the solution;

**Value Modelling** – used to focus solution development on value delivery by tracing decision to the value perspective of the stakeholder.

For a more complete description of each technique listed above please refer to the Agile Extension to the BABOK Guide V2

1. Appendix B –

*<< Attachments and supporting information as required>>*.