Appendix F

Software Requirement Specification Template

ANDS Software Requirements Specification

for

<Project Title>

ANDS Project Code <#>

Document Version <0.1>

Prepared by <author>

<organisation>

<date created>

Software Requirements Specification (SRS)

The purpose of this template is to provide the subcontractor with a framework to capture the requirements for an ANDS funded software development project.

This template is based closely on a version of the IEEE 830 Software Requirements Specification prepared by Karl E. Wiegers.

Aspects of the SRS that ANDS (a) requires, (b) recommends or (c) considers optional are indicated throughout the document.

If your organisation has used an alternative Software Requirements document format, please map/reference the required information from your existing document onto this template and attach the original Software Requirements document. If this is done, please ensure that all information required by ANDS (as indicated throughout this document) is included. *This will expedite the ANDS review processs.*

After this template has been completed, it will be reviewed by ANDS by the criteria listed at the end of this document.

Text in italics is explanatory and should be deleted in completed documents.

If you wish to discuss any aspects of your proposed work or this Software Requirements Specification, please contact your ANDS Client Liaison Officer: <contact name>, <email address> <telephone number>

ANDS Client Liaison Officer to pre-fill the table as fully as possible based on the approved Project Description.

ANDS Project Code	
Project Title	
ANDS Program	***Seeding The Commons*** ***Data Capture*** ***Public Sector Data*** ***Metadata Stores*** ***(ANDS Client Liaison Officer to delete as appropriate)***
Organisation responsible for the project (Subcontractor)	*The organisation that has entered into the subcontract agreement with ANDS/Monash University*
Organisation that will undertake the work (Sub-Subcontractor)	*If this is different from the organisation who is a party to the contract*
Name of Contact Person	*The contact person who is responsible for negotiations in respect of the project design with ANDS.*
Address and contact details of Contact Person	*Address, telephone numbers and email*
Names and affiliations of collaborators if any	*Include all those organisations that are participating in the project*

Table of Contents

Document Revision History

Name	Date	Reason For Changes	Version

1 Introduction

1. Purpose - Required

<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.>

2. Document Conventions - Optional

<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>

3. Intended Audience and Reading Suggestions - Optional

<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>

4. Product Scope - Required

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here.>

5. References - Recommended

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

2 Overall Description

1. Product Perspective - Required

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>

2. Product Functions - Required

<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, is often effective.>

3. User Classes and Characteristics - Required

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>

4. Operating Environment - Recommended

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

5. Design and Implementation Constraints - Recommended

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer's organization will be responsible for maintaining the delivered software).>

6. User Documentation - Required

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

2.7 Assumptions and Dependencies - Recommended

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

3 External Interface Requirements

1. User Interfaces - Optional

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

2. Hardware Interfaces - Recommended

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

3. Software Interfaces - Recommended

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

4. Communications Interfaces - Optional

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

4 System Features - Optional

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

1. System Feature 1

<Don't really say "System Feature 1." State the feature name in just a few words.>

1. Description and Priority

   <Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as

benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

2. Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

3. Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use "TBD" as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:
REQ-2:

4.2 System Feature 2 (and so on)

5 Other Nonfunctional Requirements

1. Performance Requirements - Recommended

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

2. Safety Requirements - Optional

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied.>

3. Security Requirements - Recommended

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

4. Software Quality Attributes - Required

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

5. Business Rules – Optional

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

6 Other Requirements - Optional

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary - Recommended

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models - Optional

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List - Optional

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>

FOR ANDS INTERNAL USE ONLY

To be completed by ANDS Client Liaison Officer

Institution:

ANDS Project Code:

Your Name:

Date of Assessment:

	Assessment Score. *Indicate your assessment on the scale using an 'X', where 1 = no/0% and 5 = yes/100%.*							Comments
	1 (no)	2	3	4	5 (yes)	unsure	n/a	
Has the subcontractor supplied their own format Software Requirements Specification?								
If yes, has all information indicated as "required" by ANDS been correctly referenced in this ANDS-format document?								
If no, has all information indicated as "required" by ANDS been entered in this ANDS-format document?								
In your opinion, does the SRS outline the same software package and software functions that were outlined in the agreed Project Description (Schedule B of the contract)? If not, give a detailed list of the differences and explain why these have been introduced.								

When completed, the ANDS Client Liaison Officer is to upload this document to JIRA, and if ready for Independent Assessment, use JIRA to both "Request Assessment" and assign it to the Lead of the appropriate Assessment Group.

To be completed by ANDS Independent Assessor

Name:

Date of Assessment:

Note: Numbers in the following descriptions refer to sections in the IEEE-based SRS template. Partners are not required to use the template, but they should provide an index using its numbering system, to simplify cross-referencing.

	Assessment Score. *Indicate your assessment on the scale using an 'X', where 1 = no/0% and 5 = yes/100%.*							**Comments.**
	1 (no)	2	3	4	5 (yes)	unsure	n/a	
1.1 Purpose								
Does this document describe all the software components that will be developed? If not, where can this other information be found?								
1.4 Product Scope								
Is the scope of the software being developed clearly defined?								
Does the purpose of the product align with ANDS objectives?								
Does the purpose of the product fit within the funding guidelines?								
1.5 References								
Can we access the referenced material?								
2.1 Product Perspective								
Does the document explain how the product is related to other existing software?								
Does the document explain how the product is related to								

other systems within the institution?								
Does the document adequately explain how the product is related to other national infrastructure?								
2.2 Product Functions								
Is there a range of functions covering the complete product scope, explained in sufficient detail to understand what the function does? (Note for projects using Agile, a plausible range of functions is expected, even if the final software varies from this.)								
Are the functions consistent with the agreed project description?								
2.3 User Classes and Characteristics								
Are specific classes of user (eg, primary investigator, project team member, lab technician, system administrator) clearly identified and described?								
Are these end users consistent with what was agreed in the project description?								
2.5 Design and Implementation Constraints								
Does the SRS demonstrate that the developers have thought about constraints imposed by: • Institution/project policies • Security • Technologies								
Are any of these constraints a cause for concern, requiring closer investigation?								
Is a lack of identified constraints itself a cause for concern, requiring closer investigation?								
2.6 User Documentation								
Does the SRS list the documentation that will be produced, including at a minimum: user manual, system administrator/installation instructions, developer documentation?								
5.4 Software Quality Attributes								
Does the SRS indicate clearly the preference priorities for the different quality attributes?								
In the assessor's opinion, are the defined levels of quality								

in each of the chosen measures appropriate, given properties of the project such as budget, scope, kind of end user, and technology constraints?								
General								
In the assessor's opinion, if software was delivered that conformed to these requirements, would this be a success for ANDS, achieving objectives such as improved data management, adding to Research Data Australia, and creating reusable software?								
In the assessor's opinion, are there any concerns that merit further investigation? If so, please provide a clear, itemised list outlining these concerns.								

When completed, the ANDS Independent Assessor is to upload the document to JIRA, and use JIRA to either "accept" and close this issue or to request that further work is undertaken ("not OK").

Bibliography

1 Bill Curtis, William E. Hefley, Sally A. Miller, *People Capability Maturity Model (P-CMM) Version 2.0*, Carnegie Mellon Software Engineering Institute, 2001. (https://resources.sei.cmu.edu/asset_files/MaturityModule/2001_008_001_435287.pdf)

2 Jeff Tian, *Software Quality Engineering*, Wiley-Interscience, 2005.

3 Michael Fagan, *Design and code inspections to reduce errors in program development*, 1976. (http://www.mfagan.com/pdfs/ibmfagan.pdf)

4 James Whittaker, Jason Arbon, Jeff Carollo, “How Google Tests Software”, Addison Wesley, 2012.