ANDS Software Requirements Specification

for

Online decision support toolkit for climate resilient seaports

ANDS Project Code: AP35

Document Version <0.1>

Prepared by <Ian Thomas>

<RMIT Unversity>

<1st August 2012>

**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:***

**MingfangWu,** [**mingfang.wu@ands.org.au**](mailto:mingfang.wu@ands.org.au)**, 03 9902 4646**

|  |  |
| --- | --- |
| **ANDS Project Code** | AP35 |
| **Project Title** | Online decision support toolkit for climate resilient seaports |
| **ANDS Program** | Applications |
| **Organisation responsible for the project (Subcontractor)** | RMIT University |
| **Organisation that will undertake the work (Sub-Subcontractor)** | RMIT University |
| **Name of Contact Person** | Professor Darryn McEvoy |
| **Address and contact details of Contact Person** | Address: GPO Box 2476, Melbourne VIC 3001  Telephone: 03 99251943  Email: [Darryn.mcevoy@rmit.edu.au](mailto:Darryn.mcevoy@rmit.edu.au) |
| **Names and affiliations of collaborators if any** | Climate Change Adaptation Program and eResearch at RMIT University, Ports Australia, National Transport Commission |

Table of Contents

Table of Contents iii

Document Revision History iv

1 Introduction 5

1.1 Purpose - Required 5

1.2 Document Conventions - Optional 5

1.3 Intended Audience and Reading Suggestions - Optional 5

1.4 Product Scope - Required 5

1.5 References - Recommended 5

2 Overall Description 5

2.1 Product Perspective - Required 5

2.2 Product Functions - Required 6

2.3 User Classes and Characteristics - Required 6

2.4 Operating Environment - Recommended 6

2.5 Design and Implementation Constraints - Recommended 6

2.6 User Documentation - Required 6

2.7 Assumptions and Dependencies - Recommended 6

3 External Interface Requirements 7

3.1 User Interfaces - Optional 7

3.2 Hardware Interfaces - Recommended 7

3.3 Software Interfaces - Recommended 7

3.4 Communications Interfaces - Optional 7

4 System Features - Optional 7

4.1 System Feature 1 7

4.2 System Feature 2 (and so on) 8

5 Other Nonfunctional Requirements 8

5.1 Performance Requirements - Recommended 8

5.2 Safety Requirements - Optional 8

5.3 Security Requirements - Recommended 8

5.4 Software Quality Attributes - Required 9

5.5 Business Rules – Optional 9

6 Other Requirements - Optional 9

Appendix A: Glossary - Recommended 9

Appendix B: Analysis Models - Optional 9

Appendix C: To Be Determined List - Optional 9

Document Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## 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.>

This document describes the software SrS: Smart Resilient Seaports version 1.0. This product is an decision support tool that will assist robust climate change adaptation planning for the Australian seaports sector; composing, (i) a data management framework encapsulating existing publicly available primary data (refined for context specific adaptation decision-making), and linking extensible models which are being developed (out of scope of this tool) to identify engineering and logistic vulnerabilities; and (ii) an innovative software interface designed in close consultation with end users to enable an interactive and explorative approach to adaptation management.

## 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.>

## 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.>

## 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.>

The software must meet all the requirements spelt out in the Project Description and Project Management Plan. The software is a data management framework, connectors to publically available primary data and connectors to climate models, and a user interface for port authorities and researchers to create narratives that describe climate adaptation scenarios.

## 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.>

# Overall Description

## 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.>

This product is a self-contained product that provides decision support and data management for climate change adaptation, and provides connectors to publically available datasets and climate models and allows this information to consolidated and further manipulated. It is designed to provide consolidation of data from CSIRO, BITRE, GA, BoM and also to leverage models under development as part of the NCCARF project.

### CSIRO, BITRE, GA, BoM datasets

Each of these services provides some interface for its data sets, either in the form of a downloadable dataset, a scrape-able web site, APIS or web services. The SrS system will create appropriate connectors for each provider to aggregate that data for further manipulation within the system.

### NCCARF models

The NCCARF project provides one or more executable models for climate adaptation. The SrS will either integrate these models in with its own functionality to allow users to execute them; or more likely have a connector (analogous to the dataset connectors) to allow users to query these models kept in remote servers and return results for further analysis.

### Australian National Data Service (ANDS)

ANDS harvest all collections in RIF-CS format, as agreed in the contract; the harvesting of RIF-CS from ANDS with SrS will be done periodically. ANDS collects all the generated RIF-CS files from the metadata collection module, by using the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) protocol.

### Data Repository

The ingested data from datasets and models (and associated metadata) will be serialized for further analysis within an external data repository that is managed by a data management framework that is part of the SrS system.

### Visualization and Analytics

Although the SrS will contain built-in functions for data visualization and standard analytics, it is expected that data may be need more specialized analysis by external visualization and analytics tools (both within RMIT and in third parties), especially if the datasets are large. This includes connections to high-performance analysis facilities if users identify a need for this functionality.

## 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.>

The approved Project Description lists these functions:

Deployed, tested and documented software system that:

1. Allows climate science, geospatial science and relevant social science data related to sea ports to be incorporated form separate data sources such as BoM, CSIRO, Geoscience Australia and BITRE, ABS into one management framework
2. Allows data be analysed using accepted risk management principles (ISO31000:2009)
3. Allows for the integration of vulnerability models (being developed via the NCCARF-funded project) related to deterioration of structural materials.
4. Allows for the inclusion of supply chain modeling information (being developed via the NCCARF-funded project).
5. Enables a significant demonstration of value in the transformation of this previously disparate information as specified in (1) and enables a significant demonstration of value in the transformation of this previously disparate information as specified in (3) and (4)

Add detailed architecture and description here

## 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.>

### Social Researchers

Need to identify these users more clearly.

### Port Managers

Need to identify these users more clearly.

### Port Engineers

Need to identify these users more clearly.

### Local Authorities

Need to identify these users more clearly.

## 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.>

The SrS system will be deployed on an infrastructure-as-a-service cloud platform to allow flexibility of deployment and improve sustainability. The eventual deployment sites can then include both RMIT-provided cloud infrastructure and third-party cloud providers if needed to allow for future development.

It will be deployed using a set of automated scripts to allow improve repeatability and stability of the system.

## 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).>  
  
Only appropriate if deploying on ITS infrastructure

## 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.>

A user manual will be produced to meet ANDS’ requirements.

## 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).>

# External Interface Requirements

The SrS system contains four major components: a set of connectors to external data and model providers, a data management framework for holding and manipulating datasets, a workflow engine and tool chain for manipulating derived datasets, and a user interface to allow decision support functions for the users.

## 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.>

The SrS will contain a web-based portal that allows users to browse publically available user stories, login to see their own created user stories and stories shared by others and alter and create their own user stories.

The workspace for information about climate resilient is a scenario, which is spactial report or grid within which are embedded data elements, in the form of tables, graphs, maps and text boxes. The data within these elements are source from datasets which are in ingesetion of external data and model results.

These scenarios can be shared with other authorized users, and when ready, can be converted into a user stories, which is summary report which can be publically accessed and be ingested by RDA, etc.

There will be an admin interface that allows authorized users to create, retrieve, update and delete elements within the system, as well as perform general maintenance of the site.

## 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.>

There are no additional hardware requirements beyond the basic linux software infrastructure as realized in hardware, and internet-based connection protocols.

## 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.>

The SrS system will include a database for holding metadata and a bulk storage system (e.g., file system) for holding datasets and access to these stores by user interfaces and connectors will be mediated through a data management framework.

## 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.>

Preferred communication protocols to dataset providers will be using HTTP in most cases, either directly through request-response REST-style web interfaces, or through web-scraping if needed. In some cases, more specialized connectors may be offered by data providers (such as SOAP web services, MySQL connectors etc.), and these will be implemented as required.

# 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.>

## System Feature 1

<Don’t really say “System Feature 1.” State the feature name in just a few words.>

4.1.1 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).>

4.1.2 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.>

4.1.3 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:

## System Feature 2 (and so on)

# Other Nonfunctional Requirements

## 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.>

The performance of the software using assessed feedback on by users. It is thought that metrics will be determined by users during the course of the project.

## 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.>

Are there any appropriate certification standards we need to look into?

## 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.>

Authentication to the web application will be done initially using internal user/password type authentication, though options will be left open to use other external authentication schemes in the future.

Each connector will communicate with its provider using the appropriate authentication mechanisms. Where possible all external communication will be encrypted using SSL.

It is expected that no personal information will be kept about users beyond name and email address, unless the user opts to add additional information to their user profile.

## 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.>

In general, the major attributes of software quality are availability, efficiency, flexibility, integrity, interoperability, maintainability, portability, reliability, reusability, robustness, safety, testability and usability. Since, the software is under development phase not all the attributes are addressed in this document. However, some of the major attributes are addressed as follows.

The developed software will be installed in the designated data center managed by RMIT, which accounts for the availability and the efficiency of the built software. Since, RMIT have successfully deployed and maintained complex software for their day-to-day operations. The access level availability of the deployed software for the researchers and public access will be as per the contract.

The built software will use open source for all its components, which provide researchers and developers not only to use our application but also can enhance its features to suit their requirement, this accounts for the reusability and interoperability of the application.

The reliability of the system is achieved by performing various testing in the deployed system.

The development team at eResearch, RMIT have set up the test framework for undergoing unit testing during the development phase. Apart from this, a regression testing at the demo server as well as in the actual test will be done before the final deployment. Since the development team works closely with the users and the product owner, a set of user acceptance testing may also be done as part of the development.

## 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.>

# 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”).