Software Verification and Validation Plan

SVVP

EOX Team

Version 1.1, 19/09/2024

DESIDE - Software Verification and Validation Plan

Destination Earth DESP Use Cases: DestinE Sea Ice Decision Enhancement (DESIDE) *Software Verification and Validation Plan* SVVP

COMMENTS and ISSUES If you would like to raise comments or issues on this document, send an email to <office@eox.at>.	**PDF** This document is available in PDF format here.
EUROPEAN SPACE AGENCY CONTRACT REPORT The work described in this report was done under ESA contract. Responsibility for the contents resides in the author or organization that prepared it.	**EOX IT Services GmbH** Thurngasse 8/4, 1090 Vienna, Austria. eox.at

AMENDMENT HISTORY

This document shall be amended by releasing a new edition of the document in its entirety. The Amendment Record Sheet below records the history and issue status of this document.

Table 1. Amendment Record Sheet

ISSUE	DATE	REASON
0.1	11/12/2023	Initial in-progress draft
0.2	22/04/2024	Update for Review 1
1.0	22/04/2024	First released version
1.1	19/09/2024	Second released version

Chapter 1. Introduction

1.1. Purpose and Scope

This document represents the Software Verification and Validation Plan (SVVP) for the DESIDE project 8482 with ESA contract 4000140320/23/I-NS. This document describes generic regression/unit tests that are run on the software when new commits are performed to ensure the software is still functioning as expected.

1.2. Structure of the Document

Chapter 2

This section provides an overview of the Destination Earth DESP Use Cases: DestinE Sea Ice Decision Enhancement (DESIDE).

Chapter 3

This section provides the software verification and validation plans, activities, resources, acceptance criteria, schedule and change control.

1.3. Reference Documents

The following is a list of Applicable and Reference Documents with a direct bearing on the content of this document.

Reference	Document Details	Version
[SOW]	Statement of Work Destination Earth DESP Use Cases selection - Round 1 Reference: CS301353.Docref.0002	1.0
[Proposal]	Proposal No. 8482: DestinE Sea Ice Decision Enhancement (DESIDE)	1.1 06/06/2023

1.4. Terminology

The following terms have been used in this document.

Term	Meaning
Admin	User with administrative capabilities on a platform.
Code	The codification of an algorithm performed with a given programming language - compiled to Software or directly executed (interpreted) within the platform.
Discovery	User finds products/services of interest to them based upon search criteria.

Term	Meaning
Interactive Web Application	An Interactive Application for analysis provided as a rich user interface through the user's web browser.
Key-Value Pair	A key-value pair (KVP) is an abstract data type that includes a group of key identifiers and a set of associated values. Key-value pairs are frequently used in lookup tables, hash tables and configuration files.
Object Store	A computer data storage architecture that manages data as objects. Each object typically includes the data itself, a variable amount of metadata, and a globally unique identifier.
Products	EO data (commercial and non-commercial) and Value-added products.
Software	The compilation of code into a binary program to be executed within the platform on-line computing environment.
User	An individual using the services.
Visualization	To obtain a visual representation of any data/products held within the platform - presented to the user within their web browser session.
Web Coverage Service (WCS)	OGC standard that provides an open specification for sharing raster datasets on the web.
Web Feature Service (WFS)	OGC standard that makes geographic feature data (vector geospatial datasets) available on the web.
Web Map Service (WMS)	OGC standard that provides a simple HTTP interface for requesting geo-registered map images from one or more distributed geospatial databases.
Web Map Tile Service (WMTS)	OGC standard that provides a simple HTTP interface for requesting map tiles of spatially referenced data using the images with predefined content, extent, and resolution.
Web Processing Services (WPS)	OGC standard that defines how a client can request the execution of a process, and how the output from the process is handled.

1.5. Glossary

The following acronyms and abbreviations have been used in this document.

Term	Definition
ADD	Architecture Design Document
AOI	Area of Interest
API	Application Programming Interface
COG	Cloud optimized GeoTiff
EO	Earth Observation
EOX	EOX IT Services GmbH

Term	Definition
ESA	European Space Agency
FUSE	Filesystem in Userspace
ICD	Interface Control Document
JSON	JavaScript Object Notation
KVP	Key-value Pair
M2M	Machine-to-machine
OGC	Open Geospatial Consortium
PMP	Project Management Plan
REST	Representational State Transfer
SDD	Software Design Document
SFTP	Secure File Transfer Protocol
SRF	Software Reuse File
SRN	Software Release Note
SRP	Software Release Plan
SRS	Software Requirements Specification
SSH	Secure Shell
STAC	Spatio-Temporal Asset Catalog
SUM	Software User Manual
SVVP	Software Verification and Validation Plan
SVVR	Software Verification and Validation Report
TOI	Time of Interest
UMA	User-Managed Access
US	User Story
WCS	Web Coverage Service
WFS	Web Feature Service
WMS	Web Map Service
WMTS	Web Map Tile Service
WPS	Web Processing Service
WPS-T	Transactional Web Processing Service

Chapter 2. Overview

Polar View Earth Observation Limited is working in collaboration with EOX IT Services, Drift+Noise Polar Services, the Danish Meteorological Institute, the Norwegian Meteorological Institute, and the Finnish Meteorological Institute to develop a fully functional Use Case that utilizes the DESP/DestinE system capabilities and data and adds value to meet the needs of policy and decision makers who require information on the past, current, and forecasted sea ice and other relevant conditions for operational purposes in the Baltic Sea, European Arctic Ocean, and the rest of the polar regions.

The Use Case will build on and complement existing operational and climate sea ice products and services including those provided by the Copernicus Marine Service, the national Ice Services, the ESA Polar Thematic Exploitation Platform (Polar TEP), and the commercial Drift+Noise IcySea app. The Use Case will augment and improve on the current offerings by:

- Aggregating information of different types and from different sources to provide common products that span jurisdictional boundaries.
- Producing new products that will improve the ability of users to make good decisions.
- Making the products available in ways and means that are appropriate for the skills and requirements of different user communities.

One driver for the project is the regulation of the International Maritime Organization (IMO) of the United Nations mandating that ships operating in the polar regions meet certain requirements (the Polar Code). Among other things, the Polar Code specifies a range of information that ships traveling in polar waters are required to access for planning and operations. The Use Case will demonstrate the value of short and medium-term forecasts of sea ice, meteorological, and ocean conditions suitable for strategic and tactical decision making by ships and their owners.

A second driver for the project is the effect of climate change on polar conditions that will impact long-term planning and policy development for polar operations such as fishing, tourism, scientific research campaigns, oil and gas development, and supplying northern communities. The Use Case will deliver long-term forecasts of how changing sea ice and other conditions will affect where different types of ships will be able to travel in the polar regions compared to historical averages.

Benefits to polar operations and the rest of society will include increased safety of life and property, decreased pollution, and protection of sensitive environmental areas.

Chapter 3. Verification

The verification approach for the DESIDE system consists of a combination of unit testing, integration testing, and system testing.

Unit Testing: Each component of the data pipeline undergoes thorough unit testing. Unit tests are designed to verify the individual functionality of each component in isolation. The unit tests ensure that the components perform as expected and adhere to the defined requirements.

Integration Testing: Integration testing is conducted to verify the interactions and compatibility between the components of the data pipeline. Integration tests are executed to ensure proper data flow and integration points between the components. These tests focus on verifying the overall functionality and communication of the integrated components.

Server Testing: The server responsible for data sharing is subjected to a comprehensive set of tests. These tests cover various aspects, including data input/output verification, data storage and retrieval, error handling, and performance under different load conditions. The server tests are designed to ensure the reliability, stability, and efficiency of the data sharing functionality.

System Testing: The main repository, which contains the deployment and bundling system, is verified through system testing. System tests are designed to evaluate the end-to-end functionality and behavior of the software system as a whole. These tests cover various scenarios and use cases to ensure that the system operates as intended and meets the specified requirements.

To facilitate the testing process, the `pytest` framework has been selected as the primary testing tool. `pytest` offers ease of use, is widely adopted within the industry, and provides comprehensive documentation. Its rich set of features enables efficient test development, execution, and result analysis.

The rationale behind this approach is to ensure that each component of the software system is thoroughly verified in isolation, as well as in conjunction with other components to verify their integration. By adopting a combination of unit testing, integration testing, and system testing, we aim to identify and address any issues early in the development cycle, ensuring the delivery of a high-quality and reliable software system.

3.1. Verification activities

This section outlines the verification activities for key DESIDE components.

Requirement	System Requirement ID	Description	Verification method
Interactive dashboard	REQ01	Data relevant for the Use Cases are available in online Polar Dashboard	Demonstration

Requirement	System Requirement ID	Description	Verification method
Data access	REQ02	Data relevant for the Use Cases are available through Polar TEP Interactive development environment	Demonstration
Connection between IcySea and Polar TEP	REQ03	Ensuring data transfer and connection between IcySea and Polar TEP	Testing
Available DESP Data	REQ04	The available DESP data is accessible via the provided Jupyter notebook.	Demonstration
Available Data	REQ05	The available fallback data is accessible via the provided Jupyter notebook.	Demonstration
Available DESP Data	REQ06	The available DESP data is accessible via the provided Jupyter notebook.	Demonstration
Available Data	REQ07	The available fallback data is accessible via the provided Jupyter notebook.	Demonstration
Available DESP Data	REQ08	The available DESP data is accessible via the provided Jupyter notebook.	Demonstration
Available Data	REQ09	The available fallback data is accessible via the provided Jupyter notebook.	Demonstration
Available Data	REQ10	The available RCM data is accessible via the dashboard visualisation.	Demonstration

Requirement	System Requirement ID	Description	Verification method
Available Data	REQ11	The available sea ice chart data are accessible via the dashboard and usable by polaris algorithm.	Demonstration

3.2. Verification criteria and acceptance

Types of verifications performed:

- Tests
- Demonstration

3.2.1. Test

Test Execution: The primary acceptance criteria for verification is that all tests, including unit tests, integration tests, and system tests, pass successfully without any critical failures or errors.

Test Results: The verification process will consider the test results generated from the execution of the test suite. The results should indicate a high percentage of passed tests, demonstrating that the software system meets the expected functionality and behavior.

Error Handling: The software system should exhibit appropriate error handling mechanisms. Verification will verify that error messages are displayed accurately, and the system recovers gracefully from errors without causing any data loss or instability.

3.2.2. Demonstration

In addition to testing, demonstration will be conducted as part of the verification process. The demonstration aims to showcase the functionality, features, and capabilities of the software system in a real or simulated environment.

By including a demonstration as part of the verification process, we aim to provide stakeholders with a tangible and visual representation of the software system's capabilities. The demonstration serves as an effective means to validate the software against the specified requirements and ensure that it meets the expectations of the end-users and stakeholders.

<< End of Document >>