System Test Plan

For

MASC

Version 1.0.0

Walter Hernandez

Shawn Merrick Miller

Luke Newcomb

Clay Pate

Version/Author	Date
1.0 / Luke Newcomb	10/16/2023
1.1 / Luke Newcomb	10/18/2023
1.2 / Walter Hernandez	10/18/2023
1.3 / Shawn Merrick Miller	10/18/2023
1.4 / Clay Pate	10/18/2023
2.0 / Clay Pate	12/05/2023
2.1 / Shawn Merrick Miller	12/06/2023
3.0 / Clay Pate	1/20/2024
3.1 / Walter Hernandez	02/05/2024
3.2 / Shawn Merrick Miller	02/07/2024
3.3 / Luke Newcomb	02/24/2024
3.4 / Luke Newcomb	02/25/2024
3.5 / Clay Pate	02/25/2024
4.0 / Clay Pate	03/30/2024
4.1 / Clay Pate	04/07/2024
4.2 / Walter Hernadez	04/07/2024
4.3 / Luke Newcomb	04/07/2024
4.4 / Shawn Merrick Miller	04/07/2024
4.5 / Luke Newcomb	04/14/2024
4.6 / Clay Pate	04/14/2024
1.0.0 / Walter Hernandez	04/14/2024

Table of Contents

1. Introduction

1.1. Purpose

This document is a test plan for MASC MagicGrid Model Testing, produced by the MASC team. It describes the testing strategy and approach to testing the team will use to verify that the created model meets the established requirements of the customer prior to release.

1.2. Objective

- Meets the requirements, specifications, and customer rules.
- Supports the intended future use and engineering functions.
- Follows the MagicGrid framework guidelines.

2. Functional Scope

The model in the scope of testing for the MASC System Testing are mentioned in the documents attached in the following path:

1. The System Requirement Specification document: SRS
2. Section 3.1 of this document

3. Overall Strategy and Approach

3.1. Testing Strategy

MASC System Testing will include testing of all functionalities that are in the scope (Section 2) identified. System testing activities will include the testing of measurable model attributes and verifiable qualities of the model such as organization and adherence to MagicGrid best practices.

The testing types section following this one will be discussing what needs to be tested. However, this section will describe how the types will be tested.

3.1.1. Model Mission Alignment Testing

Test Objective: To test the model's alignment with the selected spacecraft mission for which it was designed, while also maintaining a generic perspective.

Technique: Open the software and view the model in order to visually verify that conditions are met by the model or not.

Completion Criteria: When all requirements have been met which have to do with the mission plan, and any non-adherence to the mission plan has been fixed, or an appropriate assumption made.

Special Consideration: Access to the MASC system model, the MagicDraw software, and the System Requirements Specification document.

3.1.2. System MagicGrid Compliance Testing

Test Objective: To test the model's adherence to the MagicGrid process based on specific requirements in the SRS which reference the MagicGrid Book of Knowledge.

Technique: Open the software and view the model in order to visually verify that conditions are met within the model or not. For some aspects, the MagicDraw software itself can provide automatic insight into properties of the model.

Completion Criteria: When all requirements have been met and verified against the MagicGrid framework, and any discrepancies or failure to follow the framework have been rectified.

Special Consideration: Access to the MASC system model, the MagicDraw software, and the System Requirements Specification document.

3.1.3. System FMEA Process Testing

Test Objective: To test the model's adherence to the standard process of FMEA item identification and usage.

Technique: Open the software and view the model in order to visually verify that conditions are met by the model or not.

Completion Criteria: When all requirements have been met which have to do with the standard FMEA process, and any non-adherence to the appropriate process has been fixed, or an appropriate assumption made.

Special Consideration: Access to the MASC system model, the MagicDraw software, and the System Requirements Specification document.

3.2. System Testing Entrance Criteria

In order to start system testing, certain requirements must be met for testing readiness. The readiness can be classified into usability testing and functional testing.

3.3. Testing Types

3.3.1. Model Mission Alignment Testing

The structure of the model is tested using the System MagicGrid Compliance Testing, but actual coherence of the model to achieve a particular goal is outside of the scope of such testing. The goal of Model Mission Alignment Testing is to ensure that the model is cohesive and follows the NASA Gateway mission that was determined as the model purpose.

System Requirements Specification, 4.1.1 “The model shall capture the need for manual control of flight dynamics.”

System Requirements Specification, 4.1.2 “The model shall capture the need for automatic maintenance of station orbit.”

System Requirements Specification, 4.1.3 “The model shall capture the need for the station to produce, store, and regulate its own power.”

System Requirements Specification, 4.1.4 “The model shall capture the need to keep the crew alive and safe.”

System Requirements Specification, 4.1.5 “The model shall capture the need to accommodate extended crew mission durations.”

System Requirements Specification, 4.1.6 “The model shall capture the need to allow the crew to perform extra-vehicular activity.”

System Requirements Specification, 4.1.7 “The model shall capture the need to allow for visiting vehicles to dock.”

System Requirements Specification, 4.1.8 “The model shall capture the need to accept the transferring of crew and cargo.”

System Requirements Specification, 4.1.9 “The model shall capture the need to provide communications to the lunar surface.”

System Requirements Specification, 4.1.10 “The model shall capture the need to have the ability to support multiple self, commercial, and international partner objectives.”

System Requirements Specification, 4.1.11 “The model shall capture the need to accommodate up to and including 4 crew members..”

System Requirements Specification, 4.1.12 “The model shall capture the need to enable 30 to 90 days of a single crew mission duration.”

System Requirements Specification, 4.1.13 “The model shall capture the need to provide easy access from Earth with current launch vehicles.”

System Requirements Specification, 4.1.14 “The model shall capture the need to have continuous communication with Earth.”

3.3.2. System MagicGrid Compliance Testing

Requirements, structure, behavior, and parameters are categories within the MagicGrid framework that will be tested for both accuracy and general usability. The goal of the System MagicGrid Compliance Testing is to ensure that the stakeholder’s needs were captured and refined within the various SysML model elements on both the Black Box and White Box levels according to the MagicGrid framework.

System Requirements Specification, 4.2.1 “The system shall capture the stakeholder needs using a table which holds Requirement items.”

System Requirements Specification, 4.2.2 “The system shall represent the stakeholder needs at the context level within the System Context model as a BDD.”

System Requirements Specification, 4.2.3 “The system shall refine the functional stakeholder needs with the Use Case diagram and Use Case scenarios.”

System Requirements Specification, 4.2.4 “The system shall refine the non-functional stakeholder needs in the system-level Measures of Effectiveness (MoEs) diagram.”

System Requirements Specification, 4.2.5 “The system shall identify the expected behavior of the main use case with Functional Analysis diagrams.”

System Requirements Specification, 4.2.6 “The system shall decompose the system into conceptual subsystems using a BDD with directed composition relationships from the system to the subsystems.”

System Requirements Specification, 4.2.7 “The system shall specify subsystem-level MoEs for one or more conceptual subsystems within the model.”

System Requirements Specification, 4.2.8 “The system shall capture generated requirements in a table which contains Requirement items.”

System Requirements Specification, 4.2.9 “The system shall capture traceability of requirements through the creation of diagrams which use the Trace relationship between system requirements and stakeholder needs.”

System Requirements Specification, 4.2.10 “The system shall capture traceability of requirements through the creation of diagrams which use the Derive relationship between system requirements and use cases.”

System Requirements Specification, 4.2.11 “The system shall decompose the system into major distinct subsystems and represent them on the conceptual subsystems BDD.”

System Requirements Specification, 4.2.12 “The system shall decompose the avionics conceptual subsystem into conceptual components using a BDD with directed composition relationships from the subsystem to the components.”

System Requirements Specification, 4.2.13 “The system shall decompose the avionics subsystem into major distinct components and represent them on the avionics conceptual components BDD.”

System Requirements Specification, 4.2.14 “The system shall organize Functional Analysis diagrams using Vertical Swimlanes which correspond to each conceptual subsystem involved in the function.”

System Requirements Specification, 4.2.15 “The system shall organize Use Case Scenario diagrams using Vertical Swimlanes which correspond to the system and each external system involved in the use case.”

System Requirements Specification, 4.2.16 “The system shall organize non top-level Functional Analysis diagrams using Vertical Swimlanes which correspond to each conceptual subsystem and conceptual component involved in the function.”

3.3.3. System FMEA Process Testing

FMEA Items, Failure Modes, Causes of Failure, and Effects of Failure are items managed by and created through the System FMEA Process which has been defined by Dassault Systemes. The goal of the System FMEA Process Testing is to ensure that the well-defined process for generating and later using FMEA Items is followed accurately. This is important as it can drive later requirements in the model.

System Requirements Specification, 4.3.1 “The system shall capture possible failure items by creating unique FMEA Item objects.”

System Requirements Specification, 4.3.2 “The system shall classify FMEA Items as either electrical, software, or mechanical failures.”

System Requirements Specification, 4.3.3 “The system shall identify which model item (block, subsystem, etc) is identified as the item that fails due to an FMEA Item.”

System Requirements Specification, 4.3.4 “The system shall identify the failure mode associated with an FMEA Item by creating Failure Mode objects.”

System Requirements Specification, 4.3.5 “The system shall identify the local effect of a failure associated with an FMEA Item by creating Local Effect of Failure objects.”

System Requirements Specification, 4.3.6 “The system shall categorize local effects of a failure as effects which impact the system item identified by the model item attached to the FMEA Item.”

System Requirements Specification, 4.3.7 “The system shall identify the final effect of a failure associated with an FMEA Item by creating Final Effect of Failure objects.”

System Requirements Specification, 4.3.8 “The system shall categorize final effects of a failure as effects on the customer(s) of a system.”

System Requirements Specification, 4.3.9 “The system shall identify the cause of failure associated with an FMEA Item by creating Cause of Failure objects.”

4. Execution Plan

4.1. Plan Overview

The execution plan will detail the test cases to be executed. The Execution Plan will be put together to ensure that all the requirements are covered. The execution plan will be designed to accommodate some changes if necessary if testing is incomplete on any day. All the test cases of the projects under test in this release are arranged in a logical order depending upon their interdependency.

The test plan for MASC is as follows:

4.1.1 Model Mission Alignment

Requirement (From SRS)	Test Case Identifier (TC-X.X.X)	Input	Expected Behavior	Pass / Fail
REQ-4.1.1	TC-4.1.1	Open system model where manual control of flight dynamics would be captured.	The model captures the need for manual control of flight dynamics	Pass
REQ-4.1.2	TC-4.1.2	Open system model where automatic maintenance of station orbit would be captured.	The model captures the need for automatic maintenance of station orbit.	Pass
REQ-4.1.3	TC-4.1.3	Open system model where station power management would be captured.	The model captures the need for the station to produce, store, and regulate its own power.	Pass
REQ-4.1.4	TC-4.1.4	Open system model where crew safety	The model captures the need to keep the crew alive and safe.	Pass

		would be captured.		
REQ-4.1.5	TC-4.1.5	Open system model where mission duration would be captured.	The model captures the need to accommodate extended crew mission durations.	Pass
REQ-4.1.6	TC-4.1.6	Open system model where extra-vehicular activity would be captured.	The model captures the need to allow the crew to perform extra-vehicular activity.	Pass
REQ-4.1.7	TC-4.1.7	Open system model where visiting vehicle docking would be captured.	The model captures the need to allow for visiting vehicles to dock.	Pass
REQ-4.1.8	TC-4.1.8	Open system model where crew and cargo transfer would be captured.	The model captures the need to accept the transferring of crew and cargo.	Pass
REQ-4.1.9	TC-4.1.9	Open system model where lunar surface communications would be captured.	The model captures the need to provide communications to the lunar surface.	Pass
REQ-4.1.10	TC-4.1.10	Open system model where multiple partner interactions would be captured.	The model captures the need to have the ability to support multiple self, commercial, and international partner objectives.	Pass

REQ-4.1.11	TC-4.1.11	Open system model where crew size would be captured.	The model captures the need to accommodate up to and including 4 crew members.	Pass
REQ-4.1.12	TC-4.1.12	Open system model where mission duration would be captured.	The model captures the need to enable 30 to 90 days of a single crew mission duration.	Pass
REQ-4.1.13	TC-4.1.13	Open system model where launch costs would be captured.	The model captures the need to provide easy access from Earth with current launch vehicles.	Pass
REQ-4.1.14	TC-4.1.14	Open system model where Earth communications would be captured.	The model captures the need to have continuous communication with Earth.	Pass
REQ-5.1.1	TC-5.1.1	Delivery of all problem domain model views within 8 months.	The project is stated to be completed within the given deadline.	Pass
REQ-5.1.2	TC-5.1.2	2 External entities are identified to interact with the system of interest.	System context contains 2 or more external entities.	Pass
REQ-5.1.3	TC-5.1.3	2 conceptual subsystems are identified within	System contains 2 or more subsystems.	Pass

		the system of interest.		
REQ-5.1.4	TC-5.1.4	2 conceptual components are identified within the system of interest.	System contains 2 or more components.	Pass
REQ-5.1.5	TC-5.1.5	1 use case scenario is identified within the system of interest.	System contains a use case scenario.	Pass
REQ-5.1.6	TC-5.1.6	1 functional analysis diagram is identified within the system of interest.	System contains a functional analysis diagram.	Pass
REQ-5.1.7	TC-5.1.7	2 sub level functional analysis diagrams are identified within the system of interest.	System contains 2 sub level functional analysis diagrams.	Pass

4.1.2 System MagicGrid Compliance

Requirement (From SRS)	Test Case Identifier (TC-X.X)	Input	Expected Behavior	Pass / Fail
REQ-4.2.1	TC-4.2.1	Running the stakeholder needs model	Stakeholder needs matrix is structured in compliance with the MagicGrid book of	Pass

			knowledge and contains references to each matrix entry.	
REQ-4.2.2	TC-4.2.2	Running the system context model	System context model is structured in compliance with the MagicGrid book of knowledge and uses appropriate SysML modeling to introduce all external entities to the system context.	Pass
REQ-4.2.3	TC-4.2.3	Running the use case model	Use case model is structured in compliance with the MagicGrid book of knowledge and uses appropriate SysML modeling to refine functional stakeholder needs.	Pass
REQ-4.2.4	TC-4.2.4	Running the measures of effectiveness model	Measures of effectiveness model is structured in compliance with the MagicGrid book of knowledge and uses appropriate SysML	Pass

			modeling to refine nonfunctional stakeholder needs.	
REQ-4.2.5	TC-4.2.5	Running the functional analysis model	Functional analysis model is structured in compliance with the MagicGrid book of knowledge and uses appropriate SysML to model the functions identified in the use case model.	Pass
REQ-4.2.6	TC-4.2.6	Running the conceptual subsystems model	Conceptual subsystem model is structured in compliance with the MagicGrid book of knowledge and uses appropriate SysML modeling to capture all the functions identified in the functional analysis model and capture them with a conceptual subsystem.	Pass
REQ-4.2.7	TC-4.2.7	Running the measures of effectiveness for	Measures of effectiveness for the subsystems model is	Pass

		the subsystems model	structured in compliance with the MagicGrid book of knowledge and uses appropriate SysML modeling the nonfunctional stakeholder needs to a level which is defined within the subsystems.	
REQ-4.2.8	TC-4.2.8	Logging all generated requirements needed for model verification.	Created requirements allow for distinct tracing throughout the model.	Pass
REQ-4.2.9	TC-4.2.9	Traces between all models, requirements, and stakeholder needs.	The model provides continuous verification of the stakeholder needs.	Pass
REQ-4.2.10	TC-4.2.10	‘Derive’ relationships between specific models and stakeholder needs.	Allows tracing to the origin of parts of the model.	Pass
REQ-4.2.11	TC-4.2.11	Full system decomposition into model subsystems.	Full system detailed as parts of a BDD.	Pass
REQ-4.2.12	TC-4.2.12	Avionics decomposition	Full avionics system detailed	Pass

		into components.	as components on BDD.	
REQ-4.2.13	TC-4.2.13	Decomposition of each component.	Full BDD breakdown of each component in the avionics system.	Pass
REQ-4.2.14	TC-4.2.14	Usage of vertical swimlanes in Functional Analysis.	Separation of each external system within a Functional Analysis.	Pass
REQ-4.2.15	TC-4.2.15	Usage of vertical swimlanes in Use Case Scenario.	Separation of each external system within a set use case.	Pass
REQ-4.2.16	TC-4.2.16	Usage of vertical swimlanes to identify components in Functional Analysis.	Separation of components within a Functional Analysis.	Pass

4.1.3 System FMEA Process

Requirement (From SRS)	Test Case Identifier (TC-X.X)	Input	Expected Behavior	Pass / Fail
REQ-4.3.1	TC-4.3.1	Open system model to the Safety and Reliability Analysis table FMEA Item column.	Unique failure modes are captured as FMEA items are captured in the model.	Pass

REQ-4.3.2	TC-4.3.2	Open system model to the Safety and Reliability Analysis table category column.	The category column of the FMEA table is set to either electrical, software, or mechanical in accordance with the cause of failure.	Pass
REQ-4.3.3	TC-4.3.3	Open system model to the Safety and Reliability Analysis table failure item column.	The failure mode associated with the FMEA item is linked in the model.	Pass
REQ-4.3.4	TC-4.3.4	Open system model to the Safety and Reliability Analysis table Failure Mode column.	The failure mode associated with the FMEA Item exists and is along the same row, but in the Failure Mode column.	Pass
REQ-4.3.5	TC-4.3.5	Open system model to the Safety and Reliability Analysis table Local Effect of Failure column.	The system contains Local Effect of Failure objects.	Pass
REQ-4.3.6	TC-4.3.6	Open system model to the Safety and Reliability Analysis table Local Effect of Failure column.	The Local Effect of Failure items are associated with an impact on the system itself, not the stakeholders.	Pass

REQ-4.3.7	TC-4.3.7	Open system model to the Safety and Reliability Analysis table Final Effect of Failure column.	The system contains Final Effect of Failure objects.	Pass
REQ-4.3.8	TC-4.3.8	Open system model to the Safety and Reliability Analysis table Final Effect of Failure column.	The Final Effect of Failure items are associated with an impact on the project stakeholders, not the system itself.	Pass
REQ-4.3.9	TC-4.3.9	Open system model to the Safety and Reliability Analysis table Cause of Failure column.	The system contains Cause of Failure objects.	Pass

4.2. Traceability Matrix

Test Case ID	Satisfies Requirement	Requirement Severity
TC-4.1.1	REQ-4.1.1	CRITICAL
TC-4.1.2	REQ-4.1.2	CRITICAL
TC-4.1.3	REQ-4.1.3	CRITICAL
TC-4.1.4	REQ-4.1.4	CRITICAL
TC-4.1.5	REQ-4.1.5	CRITICAL
TC-4.1.6	REQ-4.1.6	MEDIUM
TC-4.1.7	REQ-4.1.7	MEDIUM
TC-4.1.8	REQ-4.1.8	MEDIUM
TC-4.1.9	REQ-4.1.9	CRITICAL
TC-4.1.10	REQ-4.1.10	MEDIUM
TC-4.1.11	REQ-4.1.11	LOW
TC-4.1.12	REQ-4.1.12	LOW
TC-4.1.13	REQ-4.1.13	LOW
TC-4.1.14	REQ-4.1.14	LOW
TC-5.1.1	REQ-5.1.1	CRITICAL
TC-5.1.2	REQ-5.1.2	CRITICAL
TC-5.1.3	REQ-5.1.3	CRITICAL

TC-5.1.4	REQ-5.1.4	CRITICAL
TC-5.1.5	REQ-5.1.5	CRITICAL
TC-5.1.6	REQ-5.1.6	CRITICAL
TC-5.1.7	REQ-5.1.7	CRITICAL
TC-4.2.1	REQ-4.2.1	CRITICAL
TC-4.2.2	REQ-4.2.2	CRITICAL
TC-4.2.3	REQ-4.2.3	CRITICAL
TC-4.2.4	REQ-4.2.4	CRITICAL
TC-4.2.5	REQ-4.2.5	CRITICAL
TC-4.2.6	REQ-4.2.6	CRITICAL
TC-4.2.7	REQ-4.2.7	CRITICAL
TC-4.2.8	REQ-4.2.8	CRITICAL
TC-4.2.9	REQ-4.2.9	MEDIUM
TC-4.2.10	REQ-4.2.10	MEDIUM
TC-4.2.11	REQ-4.2.11	CRITICAL
TC-4.2.12	REQ-4.2.12	CRITICAL
TC-4.2.13	REQ-4.2.13	CRITICAL
TC-4.2.14	REQ-4.2.14	CRITICAL
TC-4.2.15	REQ-4.2.15	CRITICAL

TC-4.2.16	REQ-4.2.16	MEDIUM
TC-4.3.1	REQ-4.3.1	MEDIUM
TC-4.3.2	REQ-4.3.2	MEDIUM
TC-4.3.3	REQ-4.3.3	MEDIUM
TC-4.3.4	REQ-4.3.4	MEDIUM
TC-4.3.5	REQ-4.3.5	MEDIUM
TC-4.3.6	REQ-4.3.6	LOW
TC-4.3.7	REQ-4.3.7	MEDIUM
TC-4.3.8	REQ-4.3.8	LOW
TC-4.3.9	REQ-4.3.9	MEDIUM

4.3. Traceability Matrix Defect Severity Definitions

CRITICAL	The defect causes a catastrophic or severe error that results in major problems and the functionality rendered is unavailable to the user. A manual procedure cannot be either implemented or a high effort is required to remedy the defect. Examples of a critical defect are as follows: ● System abends ● Data cannot flow through a business function/lifecycle ● Data is corrupted or cannot post to the database
MEDIUM	The defect does not seriously impair system function and can be categorized as a medium Defect. A manual procedure requiring medium effort can be implemented to remedy the defect. Examples of a medium defect are as follows: ● Form navigation is incorrect ● Field labels are not consistent with global terminology
LOW	The defect is cosmetic or has little to no impact on system functionality. A manual procedure requiring low effort can be implemented to remedy the defect. Examples of a low defect are as follows: ● Repositioning of fields on screens ● Text font on reports is incorrect

5. Environment

The System Testing Environment will be used for System Testing.

In order to conduct the testing the tester needs to have the following installed onto their computer:

- Windows 10
- Magic Systems of Systems Architect 2022x
- Magic Model Analyst 2022x
- The created model

6. Assumptions

This section lists assumptions specific to the MASC project

- The user has access to Magic Systems of Systems Architect 2022x
- The user is familiar with SysML and MagicGrid terminologies
- The user is familiar with avionics and space system terminologies
- The user has the most current MASC system model
- The user has access to Magic Model Analyzer

7. Risks

This section lists risks specific to the MASC project

- Decisions made in creating the model were derived from multiple reputable sources, and are not guaranteed to follow a specific or standardized “style”.
- The created model is intended to be general to the problem domain, and was not created to solve any singular problem or issue.
- The model was designed using only knowledge from open access or otherwise unrestricted resources, and therefore could be outdated or oversimplified.
- The model is designed to explore the “avionics” subsystem of the spacecraft, and therefore does not provide any additional information to any of the other subsystems beyond the top-level.
- The model is designed to showcase the “process” of creating a generalized spacecraft, and therefore is not intended to cover all possible requirements or layers of the system of interest.