

System Test Plan

For

MASC

Version 2.0

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

Table of Contents

1. Introduction	4
1.1. Purpose	4
1.2. Objective	4
2. Functional Scope	4
3. Overall Strategy and Approach	4
3.1. Testing Strategy	4
3.1.1. Organizational Testing	4
3.1.2. MagicGrid Adherence Testing	5
3.1.3. Mission Alignment Testing	5
3.2. System Testing Entrance Criteria	5
3.3. Testing Types	6
3.3.1. Usability Testing	6
4. Execution Plan	6
4.1. Plan Overview	6
4.1.1 Usability Testing (See 3.1.1)	7
4.1.2 Functional Testing (See 3.1.2)	7
4.2. Traceability Matrix	9
5. Environment	10
6. Assumptions	11

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:
<https://github.com/MBSE-MASC/mbse-masc/blob/main/SRS%20MASC%20V0.1.0.docx>
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. Organizational Testing

Test Objective: To test the model's internal organization and correctness based on specific requirements in the SRS.

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 the high-level customer needs are tested, and any mistakes are fixed or otherwise mitigated.

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

3.1.2. MagicGrid Adherence 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. 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.

3.2. 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. Usability 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 Usability 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.

System Requirements Specification, 4.2.1 "The system shall capture the stakeholder needs"

System Requirements Specification, 4.2.2 "The system shall represent the stakeholder needs at the context level within the System Context model"

System Requirements Specification, 4.2.3 "The system shall refine the functional stakeholder needs with the use case model and use case scenarios"

System Requirements Specification, 4.2.4 "The system shall refine the non-functional stakeholder needs in the Measures of Effectiveness(MoEs) model"

System Requirements Specification, 4.2.5 "The system shall identify the expected behavior of every function (Req. 4.2.3) of the system with the Functional Analysis model"

System Requirements Specification, 4.2.6 "The system shall capture the conceptual subsystems"

System Requirements Specification, 4.2.7 "The system shall specify MoEs for one or more conceptual subsystems"

3.3.2. Functional Testing

This section will have the requirements that makes sure that the system being modeled is following the hypothetical mission statement.

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 Usability Testing (See 3.1.1)

4.1.2 Functional Testing (See 3.1.2)

Requirement (From SRS)	Test Case Identifier (TC-X.X)	Input	Expected Behavior	Pass / Fail
4.2.1	1.1	Running the stakeholder needs model	Stakeholder needs matrix is structured in compliance with the MagicGrid book of knowledge and contains references to each matrix entry.	Pass
4.2.2	2.1	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
4.2.3	3.1	Running the use case model	Use case model is structured in compliance with the MagicGrid book of knowledge and uses appropriate SysML	Pass

			modeling to refine functional stakeholder needs.	
4.2.4	4.1	Running the measures of effectiveness model	Measures of effectiveness model is structured in compliance with the MagicGrid book of knowledge and uses appropriate SysML modeling to refine nonfunctional stakeholder needs.	Pass
4.2.5	5.1	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
4.2.6	6.1	Running the conceptual subsystems model	Conceptual subsystem model is structured in compliance with the MagicGrid book of knowledge and uses appropriate SysML	Pass

			modeling to capture all the functions identified in the functional analysis model and capture them with a conceptual subsystem.	
4.2.7	7.1	Running the measures of effectiveness for the subsystems model	Measures of effectiveness for the subsystems model is 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.	Pass

4.2. Traceability Matrix

Requirement CRITICAL: System requirements Specification, 4.2.1: “The system shall capture the stakeholder needs”

Test Cases: TC-1.1, identify whether stakeholder needs exist and capture intended requirements.

Requirement CRITICAL: System requirements Specification, 4.2.2: “The system shall represent the stakeholder needs at the context level within the System Context model”

Test Cases: TC-1.2, determine if the system represents the stakeholder needs contextually with external interfaces and each need is accounted for with an interface relationship.

Requirement CRITICAL: System requirements Specification, 4.2.3: “The system shall refine the functional stakeholder needs with the use case model and use case scenarios”

Test Cases: TC-1.3, evaluate the use case model to ensure each functional stakeholder need was evaluated with the model.

Requirement MEDIUM: System requirements Specification, 4.2.4: “The system shall refine the non-functional stakeholder needs in the Measures of Effectiveness (MoEs) model”

Test Cases: TC-1.4, evaluate the measures of effectiveness model to ensure all of the nonfunctional requirements were allocated within the model.

Requirement CRITICAL: System requirements Specification, 4.2.5: “The system shall identify the expected behavior of every function of the system with the Functional Analysis model”

Test Cases: TC-1.5, identify if all the functions determined in the use case model were captured in the functional analysis model, and have a clear traceability to the stakeholder needs.

Requirement CRITICAL: System requirements Specification, 4.2.6: “The system shall capture the conceptual subsystems”

Test Cases: TC-1.6, determine if all the functions determined in the functional analysis model were captured within one or more conceptual subsystems, and have a clear traceability to the stakeholder needs.

Requirements MEDIUM: System requirements Specification, 4.2.7: “The system shall specify MoEs for one or more conceptual subsystems”

Test Cases: TC-1.7, evaluate the MoEs for conceptual subsystems to ensure an MoE is allocated to each system, and each nonfunctional stakeholder need is represented by at least one subsystem.

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 list 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