

Simple Graph Calculator and Plotter

Norbert Jahn

July 15, 2025

Document Number	Revision 1.0
Responsible	Norbert Jahn
Status	Draft
Confidentiality	Internal

Contents

1	Introduction	2
2	Purpose of this Document	2
3	Revision history	2
4	Functional Requirements	2
5	Non-Functional Requirements	3
6	Assumptions and Clarifications	3
7	Out of Scope	3
8	Requirement Prioritization	3
9	Acceptance Criteria	3
10	QA Test Considerations	4
11	Supporting Documents	4
12	Final Product Considerations	4

1 Introduction

This document contains the functional analysis and software design specifications of the "Simple Graph Calculator and Plotter" application. This application is used as prototype of a software module used to calculate and display several mathematical functions on a graphical user interface such as sin, cos and sinc.

2 Purpose of this Document

The purpose of this document is to provide a structured overview of the functional and non functional features of the "Simple Graph Calculator and Plotter" application. This document gives a detailed analysis of the given task and its subtasks and the corresponding software requirements.

3 Revision history

Table 1: Document history

Revision	Date	Author	Comment
1.0	12.07.2025	Norbert Jahn	Creation of initial document

4 Functional Requirements

Task	Description
F1	User can select one of the mathematical functions sin, cos, sinc
F2	User can alter the parameters amplitude, frequency, phase
F3	User can set x- and y-axis range values
F4	Selected function is displayed in a 2D chart (graph)
F5	On startup recently used graph settings are restored
F6	Result graph can be exported as a vector graphic (SVG images)
F7	Invalid user inputs are handled with specified error messages

5 Non-Functional Requirements

ID	Description
NF1	The program is a Windows desktop application (WPF using .NET 4.8)
NF2	The user interface shall be intuitive
NF3	Graph parameters are stored locally for each function
NF4	No commercially restricted third-party libraries may be used
NF5	No administrator rights required

6 Assumptions and Clarifications

- No other functions beyond sin, cos, and sinc are required
- Exporting graphs as SVG-File is sufficient
- Only one graph is displayed at once
- Use of the Oxyplot library to display graphs
- English is the only available UI language

7 Out of Scope

- No 3D visualization or interactive zoom
- No symbolic equation parsing

8 Requirement Prioritization

Priority	Function	Justification
High	F1–F4: Function selection and plotting	Core functionality
Medium	F5-F6: Restore previous state	Usability improvement

9 Acceptance Criteria

- Selecting a function updates the plot correctly
- Changing the axis range refreshes the graph with new values
- On restart, the last parameter settings are automatically loaded
- Exported SVG opens without rendering errors

10 QA Test Considerations

- Verify correct function rendering for edge values (e.g., large frequency)
- Ensure parameter are loaded correctly across multiple sessions
- Test export functionality with different file names and paths
- Validate user input: prevent empty or invalid numeric fields

11 Supporting Documents

The following additional documents are recommended or required to accompany the software module in a production setting:

- **User Guide:** Describes how to operate the application, including function selection, parameter input and export options
- **Installation Guide:** Steps to install and run the application on supported Windows systems
- **Test Plan:** Defines test cases, expected results and test data for validating functional and non-functional requirements
- **Change Log:** Tracks version history and feature modifications

12 Final Product Considerations

The prototype which is described here focuses on demonstrating the core functionality of selecting and visualizing mathematical functions. However, several aspects would require additional attention in a production-ready version:

- **Error handling:** The prototype may include basic validation, but robust error reporting, localization and logging would be required for a final product
- **User interface polish:** While the prototype UI demonstrates the flow, a final version should follow usability best practices and accessibility standards
- **Extensibility:** The final software may need to support additional functions, zooming or multi-plot views
- **Persistence and configuration:** A production version would implement more flexible, versioned storage of settings (JSON)
- **Testing and deployment:** The final product would require automated tests, packaging, and proper installation mechanisms