UNIVERSITÀ DEGLI STUDI
DI SALERNO

PROGRAMMABLE CALCULATOR

Software Engineering Project

20th NOVEMBER 2023

DE LUCA A. - D'AGOSTINO M. - DE LUCA D. - FESTA R.
- GROUP 21 -

Index

1 About Team

1.1 Team Information

Name	ID
D'Agostino Marco	*0612705817*
De Luca Aniello *(group leader)*	*0612705805*
De Luca Daniele	*0612705654*
Festa Raffaele	*0612705355*

2 Planning Basis

2.1 Scope

The goal is the realization of a programmable calculator with RPN syntax which can perform computations in the set of complex numbers (ℂ), following stack-based logic. The implementation consists of a GUI and CLI user interface in which the required functions will be implemented.

2.2 Phases

Phase	Description	Sequence
Project planning	Teamwork scheduling and Gantt chart creation	Phase #1
Requirements Engineering	Requirements elicitation for Software Development	Phase #2
Design	Definition of the major software class, of their properties and interactions between them	Phase #3
Testing	Testing planification and Test Cases definition	Phase #4
Implementation	Source Code writing and testing	Phase #5

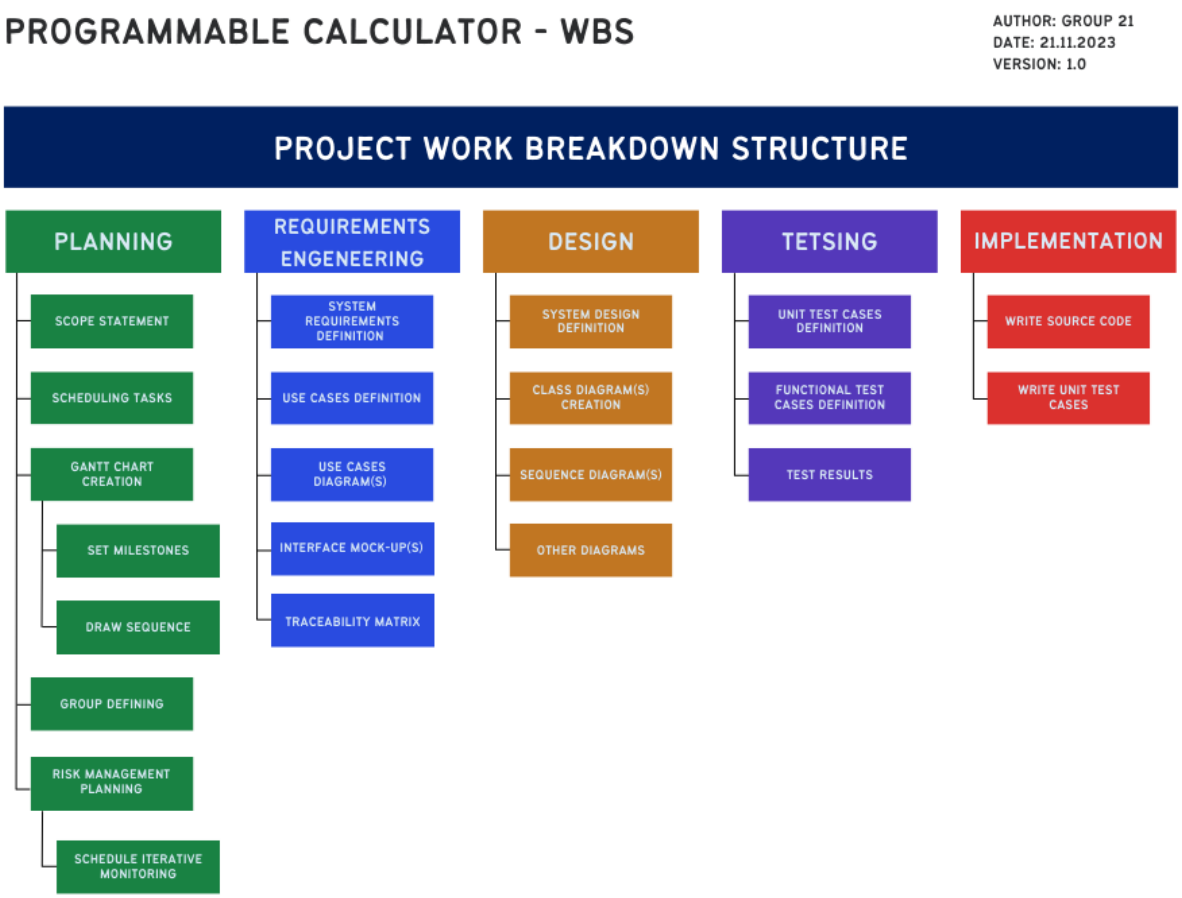
2.3 Milestones

Milestone	Description	Due Date
Project planning Done & Documented	The Project Planning has been documented and delivered.	23/11/2023
Requirements Engineering Done & Documented	The use cases have been defined and their diagrams completed. The interface mock-up is finished, and the traceability matrix has been developed.	30/11/2023
Design Done & Documented	The classes to be implemented and their interactions are defined.	7/12/2023
Testing Done & Documented	The scheduled tests for the source code have been defined.	13/12/2023
Implementation Completed	The source code is complete, and the tests programmed during the testing phase are implemented and produce the expected results.	15/11/2023

2.4 Tasks

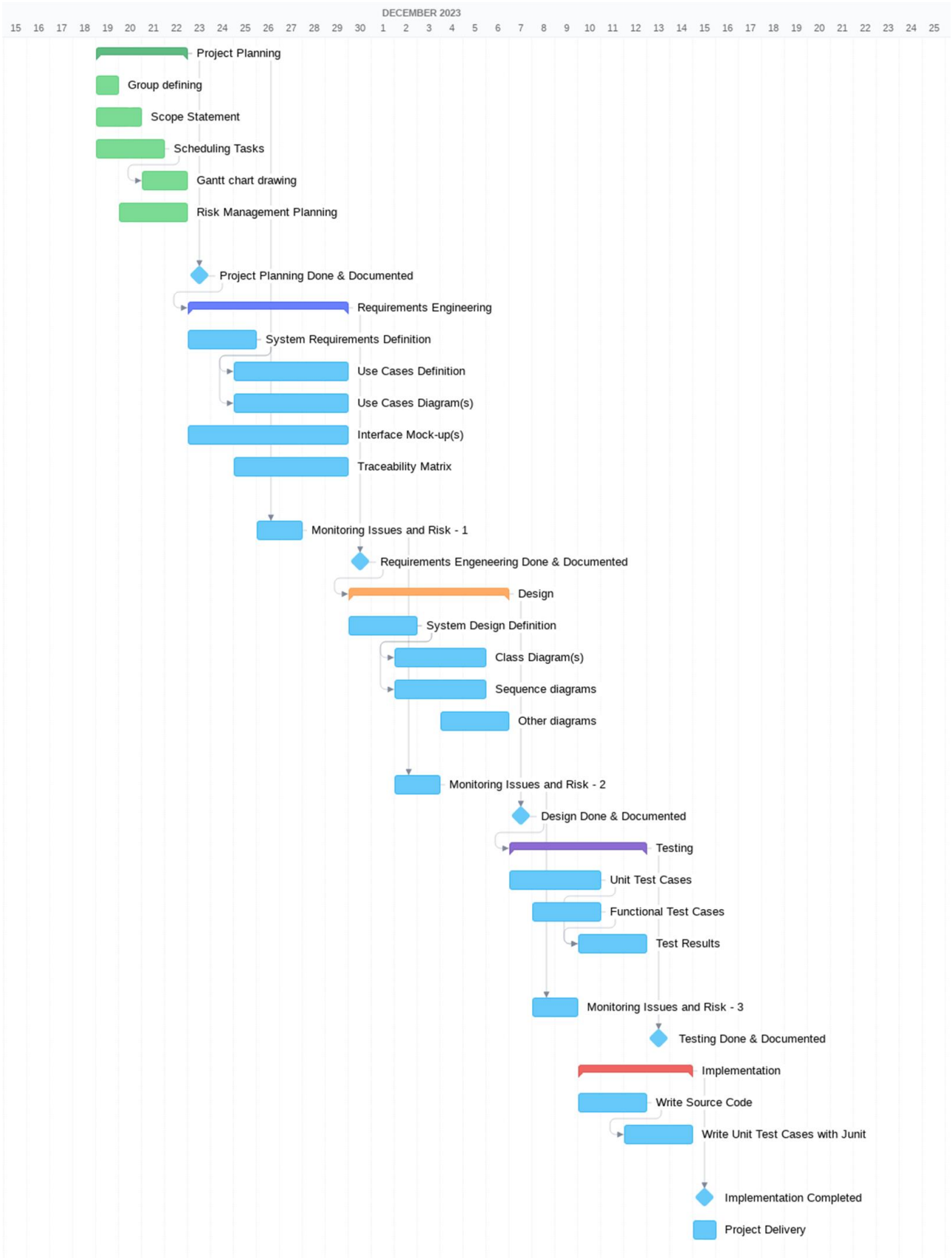
Phase	Task	Sequence
Project Planning	• Scope Statement • Group defining • Scheduling Tasks • Gantt chart creation • Risk management Planning	1st 2nd 3rd 4th 5th
Requirements Engineering	• System Requirements Definition • Interface Mock-up(s) • Use Cases Definition • Use Cases Diagram(s) • Traceability Matrix	1st 2nd 3rd 4th 5th
Design	• System Design Definition • Class Diagram(s) creation • Sequence diagrams for the relevant object interactions • Other diagrams	1st 2nd 3rd 4th
Testing	• Unit Test Cases definition • Functional Test Cases definition • Test results	1st 2nd 3rd
Implementation	• Source Code writing • Unit Test Cases writing	1st 2nd

2.5 Work Breakdown Structure



3 Project Schedule

3.1 Gantt Chart



3.2 Effort

Task	Effort
• Scope Statement • Scheduling Tasks • Gantt chart Creation • Group Defining • Risk management Planning	*2 days* *3 days* *2 days* *1 day* *3 days*
• System Requirements Definition • Use Cases Definition • Use Cases Diagram(s) • Interface Mock-up(s) • Traceability Matrix	*3 days* *5 days* *5 days* *7 days* *5 days* *2 days*
• System Design Definition • Class Diagram(s) Creation • Sequence Diagram(s) • Other diagrams	*3 days* *4 days* *4 days* *3 days* *2 days*
• Unit Test Cases Definition • Functional Test Cases Definition • Test results	*4 days* *3 days* *3 days* *2 days*
• Source Code Writing • Unit Test Cases Writing	*3 days* *3 days*
• Monitoring Issues and Risk	*6 days*

3.3 Dependencies

Activity	Depends on	Dependency Type
Gantt chart Drawing	Scheduling Tasks	Finish-to-start (0-day lag)
Use Case Definition	System Requirements Definition	Finish-to-start (0-day lag)
Use Case Diagram(s)	System Requirements Definition	Finish-to-start (0-day lag)
Class Diagram(s)	System Design Definition	Finish-to-start (0-day lag)
Sequence Diagram(s)	System Design Definition	Finish-to-start (0-day lag)
Test Results	Unit Test Case(s)	Finish-to-start (0-day lag)
Test Results	Functional Test Cases	Finish-to-start (0-day lag)
Write Unit Test Case(s)	Write Source Code	Finish-to-start (0-day lag)

4 Risk Plan

4.1 Scope

Risk management objectives aim to ensure project success, efficient resource management mitigation of adverse impacts, and proper execution of monitoring activities, with clear guidance on how to implement them.

4.2 Risk Identify & Analysis

Risk	Description	Probability	Impact
Missing Key Team Members	Some crucial team members may be missing for personal or professional reasons, resulting in a delay in scheduled tasks.	Average	High
Lack of Work on Weekends	There may be a lack of availability or willingness to work on weekends, potentially delaying the completion of necessary tasks.	Low	Average
Lack of skills in a key area	Some team members may lack the required expertise in a particular technology or critical aspect of the project.	Average	High

4.3 Risk Response Planning

Risk	Mitigation Strategy	Acceptance Strategy
Missing Key Team Members	• Implementation of rotation of responsibilities to ensure that multiple team members are aware of key activities.	• Planning of time reserves in key activities to manage possible delays.
Lack of Work on Weekends	• Clear communication regarding availability on weekends. • Scheduling crucial meetings or deadlines on standard business days.	• Flexibility in scheduling noncritical activities to accommodate weekend misses.
Lack of skills in a key area	• Prior identification of competency gaps through internal assessments. • Targeted training to fill competency gaps.	• Planning for group training if necessary.

4.4 Risk Monitoring and Control

Monitoring Procedures	Control Mechanisms
• Implement regular team meetings to assess the status of identified risks. • Utilize communication tools to gather relevant information on preparedness and accordingly create a plan.	• Appoint a manager to consistently track the team's attendance and availability. • Regular updates and evaluation of management strategies.
• Ongoing assessment of team competencies during project phases. • Implementation of feedback sessions and review of competencies.	• Early identification of any difficulties or gaps in competence. • Adoption of corrective measures, such as additional training or allocation of more resources.