Project Documentation

Agile Development of a Graphical Calculator Using Pygame

Course: The Agile Professional (SE 3122)
Instructor: Dr. KEYAMPI WATIO Martial
Duration: 2 Weeks (14 Days)
Team Size: 9 Members

Scrum Team Structure

Product Owner
1 Member

Scrum Master
1 Member

Developers
7 Members

Faculty of Information and Communication Technologies
Fall 2025

Team Members and Roles

No.	Name	Matricule	Role
1	Kamdeu Yamdjeuson Neil Marshall	ICTU20241386	Scrum Master
2	Tembong Jennette Ndip	ICTU20241752	Product Owner
3	Karel Jess Bissakonou	ICTU20241743	Developer 1 (Core Logic)
4	Ngong Judd Ngum JR	ICTU20241253	Developer 2 (GUI & Layout)
5	OLEME ELOBO RONALD JEAN DE DIEU	ICTU20241912	Developer 3 (Input Handling)
6	Feutseu Kenmogne Junior Erwan	ICTU20234174	Developer 4 (Error Handling)
7	Asongwe Tony Khan	ICTU20241379	Developer 5 (UI/UX Design)
8	Bidias Killian	ICTU20234020	Developer 6 (Tester/QA)
9	Kosho Angelo	ICTU20234127	Developer 7 (Integration & DevOps)

Table 1: Complete Team Members List with Roles and Matricules

Contents

Abstract

This document provides comprehensive documentation for the Agile Pygame Calculator project, demonstrating the practical application of Agile methodologies in a software development context. The project involved the development of a graphical calculator application using Python's Pygame library, executed over two weeks by a 9-member Scrum team. The implementation combines Scrum for sprint planning and ceremonies with Kanban for workflow visualization, utilizing GitHub Projects for task management. This report details the Agile framework selection, team structure, development process, challenges faced, and lessons learned throughout the project lifecycle.

Chapter 1

Introduction

1.1 Project Overview

This project simulates real-world Agile software development by creating a graphical calculator application. The primary objectives were to:

- Develop a functional calculator with basic arithmetic operations
- Implement an intuitive graphical user interface using Pygame
- Apply Agile principles through Scrum ceremonies
- Utilize Kanban for workflow visualization and task tracking
- Foster team collaboration and communication

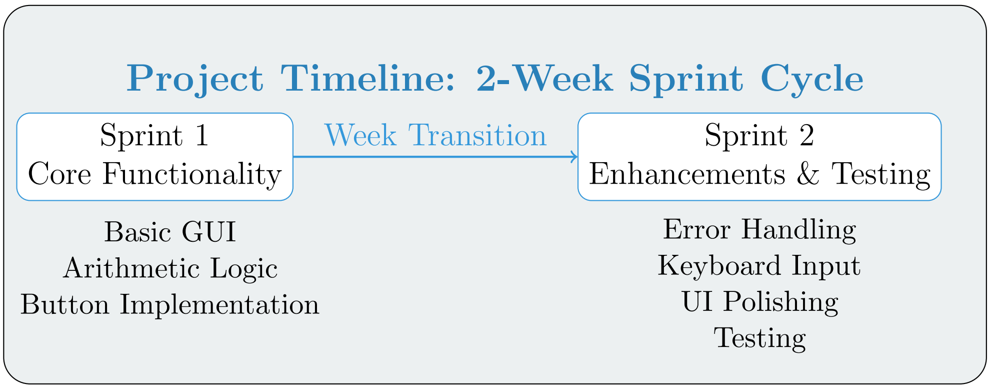


Figure 1.1: Project Sprint Timeline

Chapter 2

Agile Framework Selection and Justification

2.1 Hybrid Approach: Scrum + Kanban

We adopted a hybrid Agile approach combining Scrum and Kanban to leverage the strengths of both methodologies:

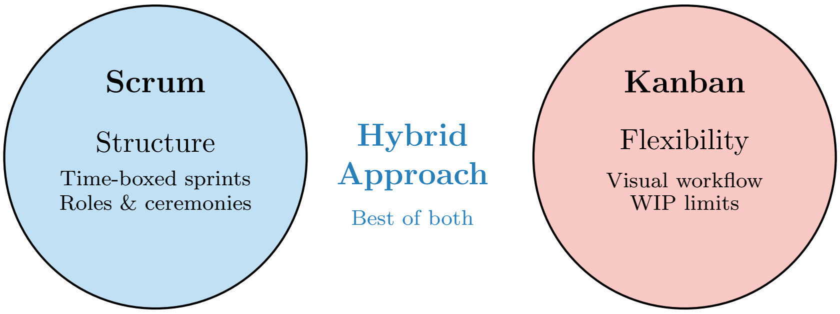


Figure 2.1: Scrum-Kanban Hybrid Approach

2.1.1 Scrum Implementation

Scrum was selected for:

- Clear sprint planning and time-boxing
- Defined roles and responsibilities
- Regular inspection and adaptation
- Structured ceremonies (Daily Stand-ups, Sprint Review, Retrospective)

2.1.2 Kanban Implementation

Kanban complemented Scrum through:

- Visual workflow management via GitHub Projects
- Work-In-Progress (WIP) limits
- Continuous delivery approach
- Real-time progress tracking

Chapter 3

Team Structure and Roles

3.1 Scrum Team Composition

The project team comprised 9 members with clearly defined roles:

Role	Responsibilities	Members
Product Owner	Defines product vision, manages backlog, prioritizes features, creates documentation	1
Scrum Master	Facilitates ceremonies, removes blockers, maintains Agile process	1
Development Team	Implements features, tests code, collaborates on delivery	7

Table 3.1: Team Role Distribution

3.2 Detailed Role Breakdown

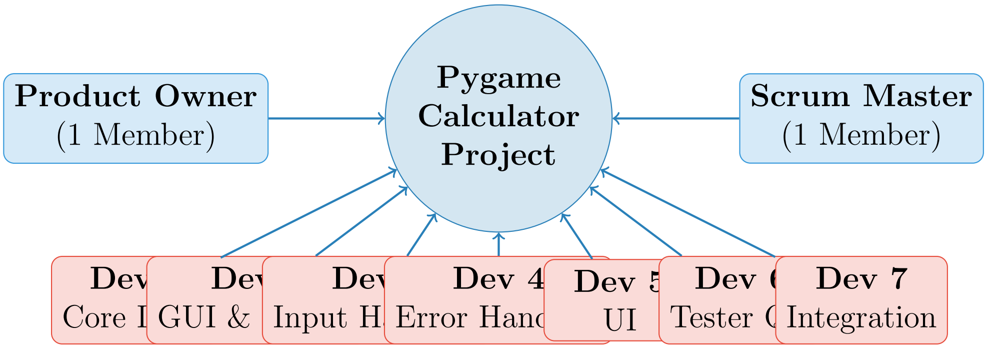


Figure 3.1: Team Structure Visualization

3.2.1 Developer Specializations

The 7 developers had specialized roles:

Developer	Specialized Focus and Responsibilities
Developer 1	Core arithmetic logic implementation (addition, subtraction, multiplication, division)
Developer 2	GUI framework and calculator layout design
Developer 3	User input handling (mouse and keyboard events)
Developer 4	Error handling and validation (division by zero, invalid inputs)
Developer 5	UI/UX design (colors, fonts, visual polish)
Developer 6	Quality assurance and testing
Developer 7	Integration and DevOps support (GitHub workflow, merges)

Table 3.2: Developer Specializations

Chapter 4

Project Execution and Development Process

4.1 Sprint Planning

The project was divided into two 1-week sprints:

4.1.1 Sprint 1: Core Functionality

Task	Description	Assigned Role
GitHub Setup	Create repository and Kanban board	Scrum Master
Pygame Window	Initialize main application window	Developer 2
Button Layout	Design calculator button grid	Developer 5
Arithmetic Logic	Implement +, -, ×, ÷ operations	Developer 1
Display Screen	Create input/output display area	Developer 2

Table 4.1: Sprint 1 Backlog

4.1.2 Sprint 2: Enhancements and Testing

Task	Description	Assigned Role
Error Handling	Division by zero validation	Developer 4
Keyboard Input	Support for keyboard entry	Developer 3
UI Polishing	Improve colors, fonts, and effects	Developer 5
Testing	Comprehensive feature testing	Developer 6
Documentation	README and final report	Product Owner

Table 4.2: Sprint 2 Backlog

4.2 Kanban Workflow Implementation

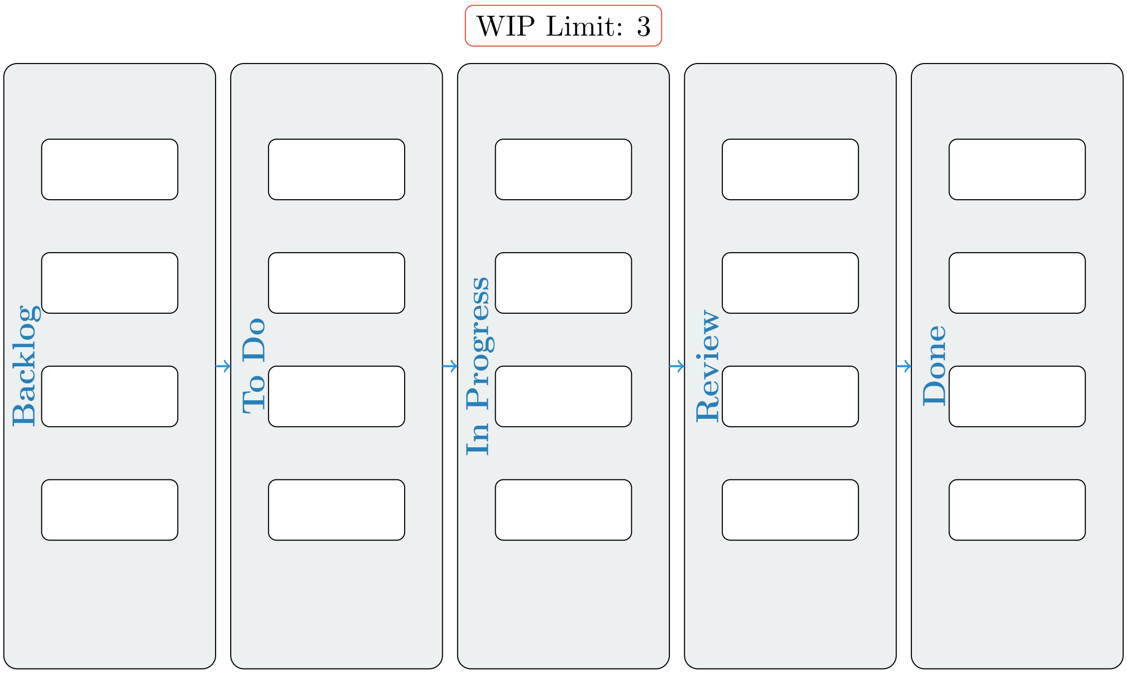


Figure 4.1: Kanban Board Structure (GitHub Projects)

Chapter 5

Scrum Ceremonies and Evidence

5.1 Daily Stand-ups

Daily 15-minute meetings were conducted to:

- Report progress since last stand-up
- Plan work for the current day
- Identify blockers and dependencies

5.2 Sprint Review and Retrospective

Ceremony	Outcomes and Evidence
Sprint Review	• Demonstrated working calculator • Collected feedback on features • Updated product backlog • **Evidence:** Meeting notes, demo recording
Sprint Retrospective	• Discussed what went well (effective communication) • Identified improvements (better task estimation) • Created action items for next sprint • **Evidence:** Retrospective board, improvement backlog

Table 5.1: Scrum Ceremonies Documentation

5.3 Visual Evidence of Agile Practices

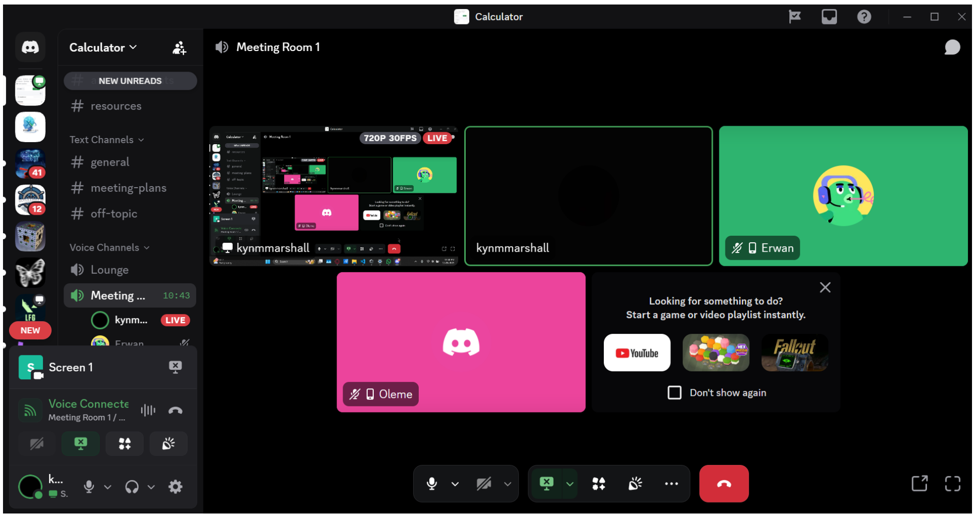
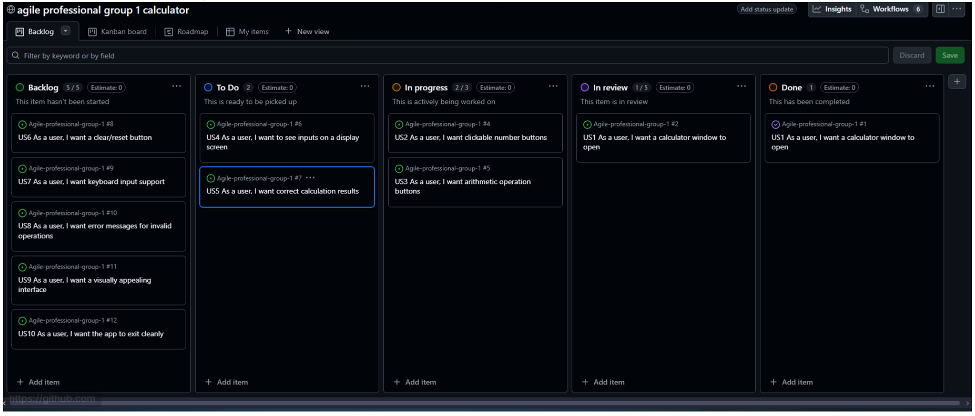


Figure 5.1: Discord Daily Stand-up Screenshot



Backlog 5 / 5 Estimate: 0
This item hasn't been started

Agile-professional-group-1 #8
US6 As a user, I want a clear/reset button

Agile-professional-group-1 #9
US7 As a user, I want keyboard input support

Agile-professional-group-1 #10
US8 As a user, I want error messages for invalid operations

Agile-professional-group-1 #11
US9 As a user, I want a visually appealing interface

Agile-professional-group-1 #12
US10 As a user, I want the app to exit cleanly

+ Add item

To Do 2 Estimate: 0
This is ready to be picked up

Agile-professional-group-1 #6
US4 As a user, I want to see inputs on a display screen

Agile-professional-group-1 #7
US5 As a user, I want correct calculation results

+ Add item

In progress 2 / 3 Estimate: 0
This is actively being worked on

Agile-professional-group-1 #4
US2 As a user, I want clickable number buttons

Agile-professional-group-1 #5
US3 As a user, I want arithmetic operation buttons

+ Add item

In review 1 / 5 Estimate: 0
This item is in review

Agile-professional-group-1 #2
US1 As a user, I want a calculator window to open

+ Add item

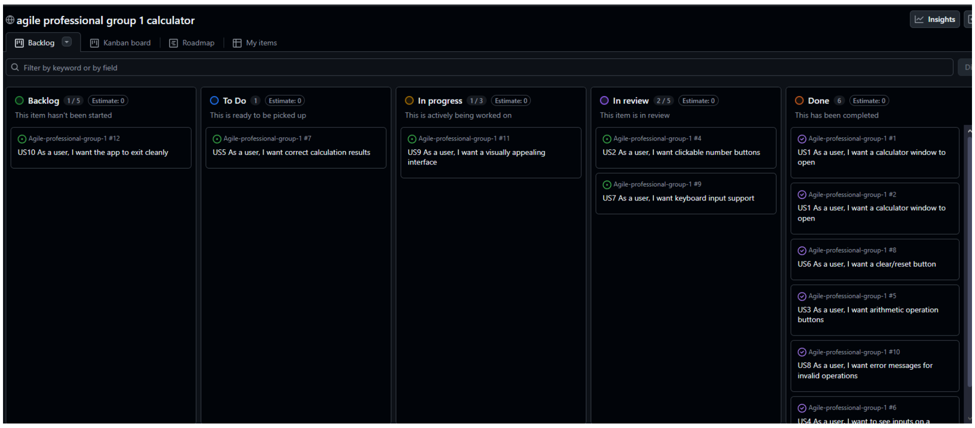
Done 1 Estimate: 0
This has been completed

Agile-professional-group-1 #1
US1 As a user, I want a calculator window to open

+ Add item

https://github.com

Figure 5.2: GitHub Projects Kanban Board 1


agile professional group 1 calculator
Insights
Backlog
Kanban board
Roadmap
My items
Filter by keyword or by field

Backlog 1 / 5 Estimate: 0
This item hasn't been started

Agile-professional-group-1 #12
US10 As a user, I want the app to exit cleanly

To Do 1 Estimate: 0
This is ready to be picked up

Agile-professional-group-1 #7
US5 As a user, I want correct calculation results

In progress 1 / 3 Estimate: 0
This is actively being worked on

Agile-professional-group-1 #11
US9 As a user, I want a visually appealing interface

In review 2 / 5 Estimate: 0
This item is in review

Agile-professional-group-1 #4
US2 As a user, I want clickable number buttons

Agile-professional-group-1 #9
US7 As a user, I want keyboard input support

Done 6 Estimate: 0
This has been completed

Agile-professional-group-1 #1
US1 As a user, I want a calculator window to open

Agile-professional-group-1 #2
US1 As a user, I want a calculator window to open

Agile-professional-group-1 #8
US6 As a user, I want a clear/reset button

Agile-professional-group-1 #5
US3 As a user, I want arithmetic operation buttons

Agile-professional-group-1 #10
US8 As a user, I want error messages for invalid operations

Agile-professional-group-1 #6
US4 As a user, I want to see inputs on a

Figure 5.3: GitHub Projects Kanban Board 2

Chapter 6

Technical Implementation

6.1 Technology Stack

Technology	Purpose and Implementation
Python 3.x	Primary programming language
Pygame Library	Graphical user interface and rendering
GitHub	Version control and collaboration
GitHub Projects	Kanban board for task management
Discord/WhatsApp	Team communication and stand-ups

Table 6.1: Technology Stack

6.2 Code Structure

Key components of the calculator implementation:

```
# Calculator Architecture Overview
1. Main Application Loop
   - Event handling (mouse/keyboard)
   - Screen rendering
   - State management

2. GUI Components
   - Display screen
   - Button grid
   - Color schemes and fonts

3. Business Logic
   - Arithmetic operations
   - Input validation
   - Error handling

4. User Interaction
   - Mouse click detection
   - Keyboard input mapping
   - Button hover effects
```

Listing 6.1: Main Calculator Structure

6.3 Key Features Implemented

- Basic arithmetic operations (+, -, ×, ÷)
- Graphical user interface with button grid
- Dual input methods (mouse and keyboard)
- Error handling for division by zero
- Visual feedback for button interactions
- Clean application exit and reset functionality

Chapter 7

Challenges and Agile Solutions

Challenge	**Agile Solution**	**Impact**
Time Constraints	Short sprints (1 week), clear prioritization	Maintained velocity, delivered MVP
Skill Variations	Pair programming, knowledge sharing sessions	Improved team competency
Merge Conflicts	Feature branches, regular pull requests	Reduced integration issues
Scope Creep	Strict backlog management, PO approval	Focused on essential features
Bug Management	Continuous testing, sprint retrospectives	Higher code quality

Table 7.1: Challenges and Agile Solutions

Chapter 8

Conclusion and Lessons Learned

8.1 Project Outcomes

- Successfully delivered a fully functional graphical calculator
- Demonstrated practical application of Agile methodologies
- Achieved effective team collaboration across 9 members
- Implemented hybrid Scrum-Kanban approach successfully
- Created comprehensive documentation and presentation

8.2 Agile Lessons Learned

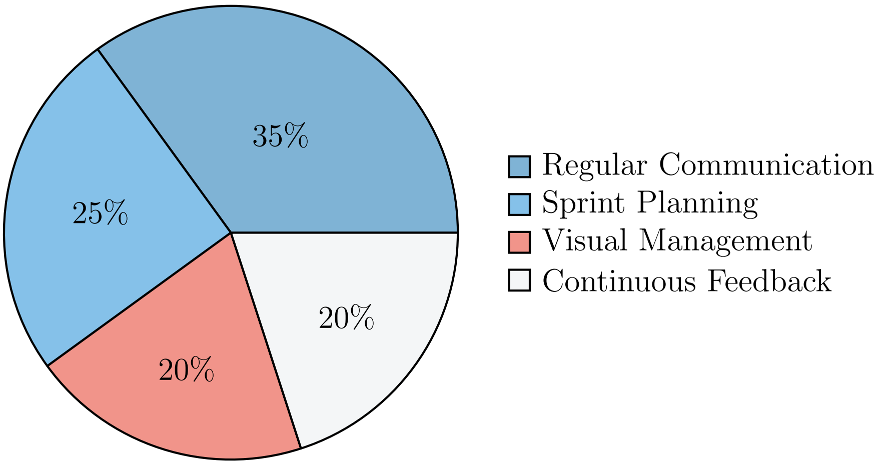


Figure 8.1: Key Success Factors in Agile Implementation

8.3 Recommendations for Future Projects

1. Start with clearer definition of done criteria
2. Implement automated testing earlier in the process
3. Use more detailed user story point estimation

4. Incorporate continuous integration practices
5. Conduct more frequent code reviews

Appendix A

Appendix A: Project Artifacts

A.1 Product Backlog Items

ID	User Story	Acceptance Criteria	Priority
US1	As a user, I want a calculator window to open	Window opens with correct dimensions	High
US2	As a user, I want clickable number buttons	All numbers 0-9 respond to clicks	High
US3	As a user, I want arithmetic operation buttons	+, -, ×, ÷ perform correct operations	High
US4	As a user, I want to see inputs on display	Input appears on screen as entered	High
US5	As a user, I want correct calculation results	Calculations match mathematical rules	High

Table A.1: Sample User Stories from Product Backlog

A.2 Meeting Minutes Template

Daily Stand-up Meeting Minutes
Date: ______________________________ Attendees: ______________________________
What was done yesterday: ______________________________________
What will be done today: ______________________________________
Blockers/Issues: ______________________________________

Table A.2: Meeting Minutes Template Used