

**INTERNATIONAL SCHOOL**

**COURSE PROJECT 1**

**CMU-CS 246 BIS - TEAM 01**

**DEFINDED PROCESS DOCUMENT**

Version: 1.1

**SIMPLE CALCULATOR SYSTEM\_SCS**

**Mentor: Nguyen Dang Quang Huy**

***Team Member*:**

Nhung, Ngo Thi Tuyet

Dung, Nguyen Van

An, Ngo Nguyen Truong

Duyen, Huynh Thi My

Loc, Truong Tan

**Approved by Huy Nguyen Dang Quang**

**Project Plan Review Panel Representative:**

Name Signature Date

**Simple Calculator System - Mentor:**

Name Signature Date

Da Nang, 01/2024

**PROJECT INFORMATION**

|  |  |
| --- | --- |
| **Project Acronym** | SCS |
| **Project Title** | SIMPLE CALCULATOR SYSTEM | | | |
| **Start Date** | 20 – Jan–2024 | **End Date** | 27–Jan–2024 | |
| **Lead Institution** | International School, Duy Tan University | | | |
| **Team Member** | **Name** | **Email** | | **Phone** |
| 28209043094 | Ngo Thi Tuyet Nhung | nn8242115@gmail.com | | 0763770335 |
| 28219050114 | Nguyen Van Dung | mynameisdungz@gmail.com | | 0906577651 |
| 28211134788 | Ngo Nguyen Truong An | ngonguyentruongan2907@gmail | | 0919499505 |
| 28209043472 | Huynh Thi My Duyen | myduyenhuynh1010@gmail.com | | 0913228912 |
| 28219101466 | Truong Tan Loc | tanlocamsat@gmail.com | | 0393165051 |

**PROJECT PLAN DOCUMENT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Document Title** | Project Plan Document | | |
| **Author(s)** | Team 01 | | |
| **Date** | Jan 21th, 2024 | **File name:** |  |
| **URL** |  | | |
| **Access** | Project and CMU Program | | |

**SIGNATURE**

**Document Approvals:** The following signatures are required for approval of this document.

|  |  |  |  |
| --- | --- | --- | --- |
| **Mentor** | Nguyen Dang Quang Huy | **Signature:** |  |
| **Date:** |  |
| **Team Leader** | Ngo Thi Tuyet Nhung | **Signature:** |  |
| **Date:** |  |
| **ProDuct**  **Owner** | Nguyen Van Dung | **Signature:** |  |
| **Date:** |  |
| **Team member(s)** | Huynh Thi My Duyen | **Signature:** |  |
| **Date:** |  |
| Truong Tan Loc | **Signature:** |  |
| **Date:** |  |
| Nguyen Van Dung | **Signature:** |  |
| **Date:** |  |

**REVISION HISTORY**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Person(s)** | **Date** | **Description** |
| **1.0** | All members | Jan 20th, 2024 | Draf |
| **1.1** | Nhung, Ngo Thi Tuyet | Jan 27th, 2024 | Final |

**TABLE OF CONTENTS**

[**1. PROJECT DESCRIPTION 4**](#_Toc157118170)

[**2. PROJECT DEVELOPMENT APPROACH 4**](#_Toc157118171)

[**2.1. TECHNICAL PROCESS 4**](#_Toc157118172)

[**2.1.1. REASON FOR SELECTING 4**](#_Toc157118173)

[**2.1.2 WATERFALL METHODOLOG 5**](#_Toc157118174)

[**2.1.3 WATERFALL PROCESS 5**](#_Toc157118175)

# 1. PROJECT DESCRIPTION

|  |  |  |  |
| --- | --- | --- | --- |
| **Project code** | SCS | **Contract type** | Internal Project |
| **Customer** | N/A | **End-user** | USER |
| **Project type** | Internal | **Project Manager** | Nhung, Ngo Thi Tuyet |
| **Project category** | Simple calculation |  |  |
| **Application type** | App |  |  |

# 2. PROJECT DEVELOPMENT APPROACH

## 2.1. TECHNICAL PROCESS

* We use water fall methodology in our project, besides using software tools to manage work, assign tasks to team members such as Messenger, Google Drive, Discord, and Source code management tools like NetBean. And for project requirements, we use the Apache netbean for the app and Java for the database.

## 2.1.1. REASON FOR SELECTING

* In today's tech-obsessed world, where fancy features and bells and whistles abound, choosing a simple calculator might seem like stepping back in time. But hear me out - there's a quiet power in embracing the basic.
* Our desire for a flexible and adaptable project development model mirrors the appeal of the simple calculator. Just as Agile teams tackle features in short sprints, the calculator focuses on the core functions of math, offering precise results without needless complexity. This laser-sharp focus allows us to learn and iterate quickly, building confidence and momentum, just like a small development team nailing each sprint target.
* Think of the traditional, feature-packed calculator as a waterfall project – rigid, inflexible, and prone to errors if requirements shift. The simple calculator, on the other hand, is agile personified: lightweight, adaptable, and ready to tackle new problems (like long division on the fly) whenever they arise.
* Yes, our team is learning the ropes of development, but that's where the simple calculator shines. Its intuitive interface and familiar functions are the perfect learning platform, building our skills without overwhelming us. Just like small, measurable wins in Agile keep developers motivated, mastering basic calculations on the calculator fuels our confidence and propels us towards more complex tasks.
* Sure, we might encounter development stage problems, and requirements might evolve. But just like Agile embraces change, the simple calculator adapts too. It handles unexpected calculations with grace, offering solutions without demanding a complete overhaul. It's the perfect partner for a team navigating the dynamic world of software development.
* So, let's embrace the simple calculator, not as a relic of the past, but as a beacon of adaptability and efficiency. In its unassuming interface lies the power to learn, grow, and conquer any mathematical challenge, just like an Agile team tackling ambitious projects one sprint at a time.
* Remember, sometimes, the best way to move forward is to take a step back and rediscover the power of simplicity. In the hands of a curious mind, the humble calculator becomes a springboard for learning and accomplishment, a testament to the power of agile principles in any endeavor, from software development to everyday life.

## 2.1.2 WATERFALL METHODOLOG

1. Requirements gathering - This stage involves identifying user needs through interviews or surveys with stakeholders or customers; defining project scope; creating user stories or use cases; documenting requirements; and prioritizing them based on business value.

2. Design - In this stage, developers create detailed designs for the software application based on the requirements gathered in the previous stage. This stage involves creating wireframes, mockups, and prototypes to visualize the user interface and user experience.

3. Implementation - This stage involves coding the software application based on the designs created in the previous stage. Developers write code, test it, and debug any issues that arise.

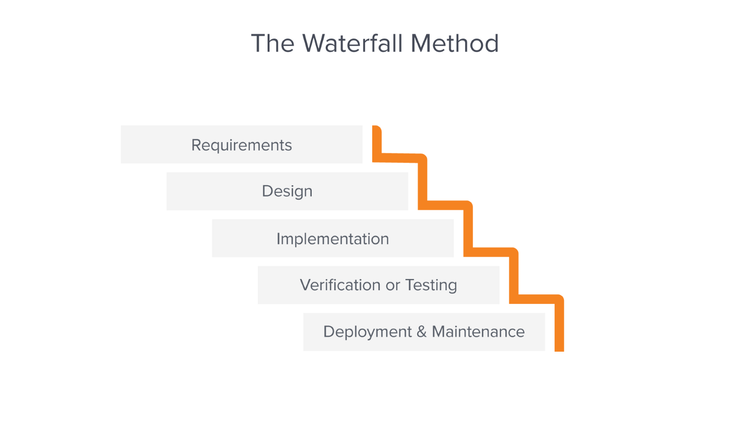
4. Testing - In this stage, the software application is thoroughly tested to ensure it meets the requirements and is free from bugs or defects. This stage includes functional testing, performance testing, and regression testing.

5. Deployment - This stage involves releasing the software application to production, making it available to end-users. This stage includes configuring servers, setting up databases, and deploying the code to the production environment.

6. Maintenance - This stage involves ongoing support and maintenance of the software application after deployment. This stage includes fixing bugs, addressing user feedback, and making updates or enhancements to the software application as needed.

Overall, the Waterfall methodology is a structured and predictable approach to software development that emphasizes documentation and a linear process for delivering software applications. However, it can be less flexible than other methodologies, such as Agile, and may not be as well-suited for complex or rapidly changing projects.

## 2.1.3 WATERFALL PROCESS



* Developers can catch design errors during the analysis and design stages, helping them to avoid writing faulty code during the implementation phase.
* The total cost of the project can be accurately estimated, as can the timeline, after the requirements have been defined.
* With the structured approach, it is easier to measure progress according to clearly defined milestones.
* Developers who join the project in progress can easily get up to speed because everything they need to know should be in the requirements document.
* Customers aren’t always adding new requirements to the project, delaying production.

**3. REFERENCES**

**https://business.adobe.com/blog/basics/waterfall#:~:text=The%20Waterfall%20methodology%20—%20also%20known,before%20the%20next%20phase%20begins.**