Software Requirements Specification

(SRS)

For

COCOMO II Simulator

Table of Contents

1.0 INTRODUCTION

1.1 Scope

The project **COCOMO II Simulator** is an application software that allows one to estimate the cost, effort, and schedule when planning a new software development activity. It is based on the **CO**nstructive **CO**st **MO**del II which is the latest major extension to the original COCOMO® (COCOMO® 81) model published in 1981 by Dr. Barry Boehm.

It can be used for the following major decision situations:

- Making investment or other financial decisions involving a software development effort
- Setting project budgets and schedules as a basis for planning and control
- Deciding on or negotiating trade-offs among software cost, schedule, functionality, performance or quality factors
- Making software cost and schedule risk management decisions
- Deciding which parts of a software system to develop, reuse, lease, or purchase
- Making legacy software inventory decisions: what parts to modify, phase out, outsource, etc.
- Setting mixed investment strategies to improve organization's software capability, via reuse, tools, process maturity, outsourcing, etc.

This document gathers all the requirement applicable for the functioning of this project. It is supposed to estimate the cost, effort, and schedule of the software development activity based on the selected sizing method (sLOC / Function Points).

1.2 Technologies to be used

Front End Technology: **JAVA SE** 11.0.2
Back End Technology:
IDE:
Configuration Management tool:

2.0 GENERAL REQUIREMENTS

2.1 Functionalities

The project “COCOMO II Simulator” is supposed to perform the following functionalities:

- Option for user to input estimated software size either in
    1. Source Lines Of Code (**sLOC**)
    2. Function Points (**FP**)
    3. Set of various other inputs (for calculation of sLOC or FP)

- If the user selects option iii (from above mentioned options),

    - Option for user to either directly input the value for **DI** (Degree of Influence) or provide values for all 14 **GSCs** (General System Characteristics) for its calculation.

    - Option for user to either directly input the value for Unadjusted Function Point (UFP) or provide count for:
        - External Inputs (**EIs**),
        - External Outputs (**EOs**),
        - External Inquiries (**EQs**),
        - Internal Logic Files (**ILFs**),
        - External Interface Files (**EIFs**),

      along with the weighting factor which can be selected between:
        1. **Simple**,
        2. **Average**,
        3. **Complex**,

      for UFP’s calculation. The values for DI and UFP are further utilized for the calculation of FP.

    - Option for user to provide the value for **Language factor** of programming language used for calculation of sLOC from FP or vice versa.

- Option for user to select between different **development modes** of the software i.e.,

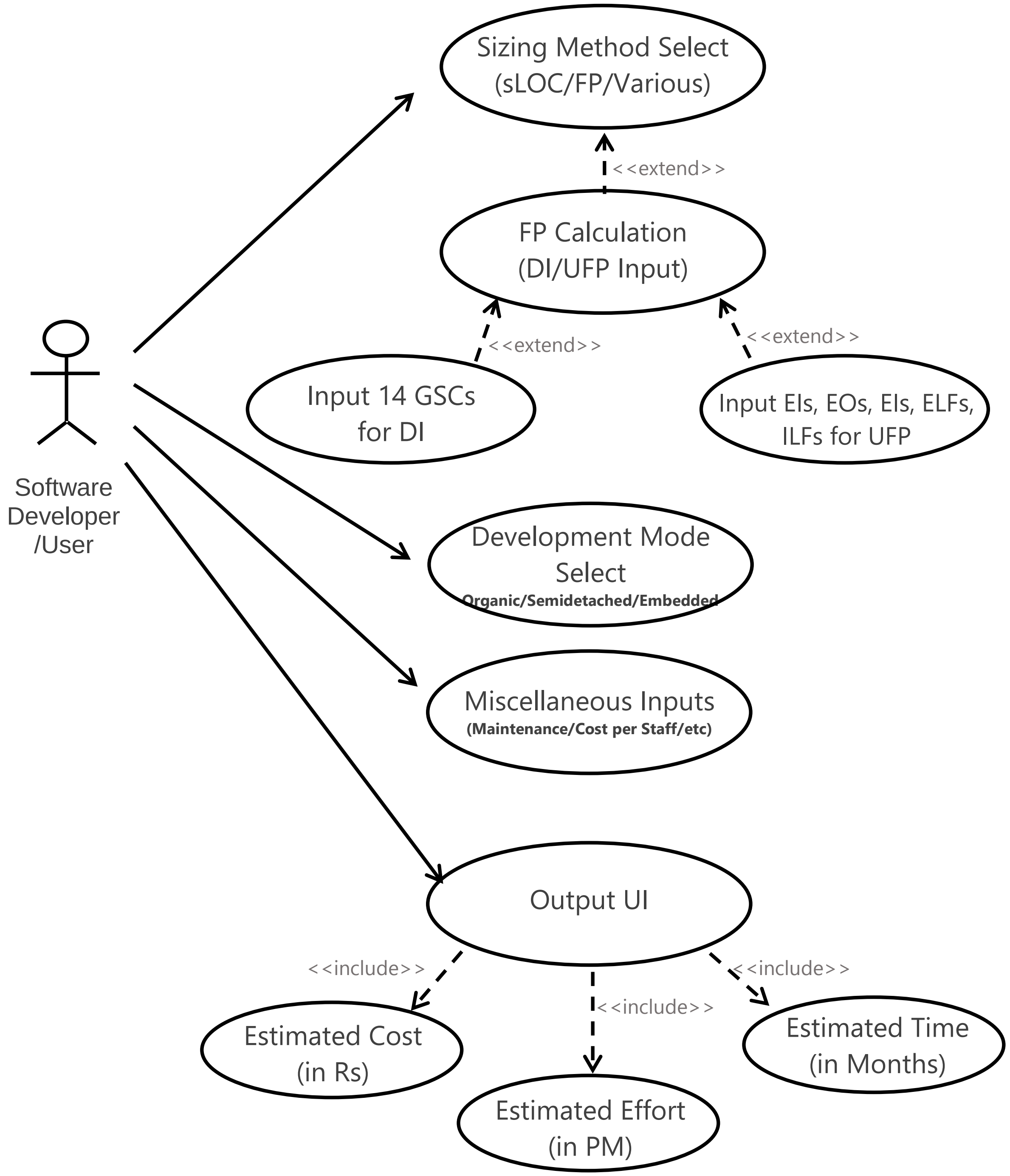
i. Organic
ii. Semi Detached
iii. Embedded

- Option for user to either take **maintenance** charges into account or not.

- Option for user to input “Cost per Staff” for the calculation of **Cost Estimation**.

- As the **Output**, the application will provide with the estimated values of
  - **Cost** (in Rs)
  - **Effort** (in Person-Months)
  - **Schedule** (in Months)

2.2 *Use Case Model Diagrams*



2.3 *Interfaces*

N/A

2.4 *General Constraints*

N/A

2.5 *Supplementary requirements*

N/A

3 Definition, Acronyms, and Abbreviations

- The key word <<include>> means that parent use case (tailing end of the arrow) must always include the child use case (pointing end of the arrow).

- The key word <<extend>> means that the parent use case (pointing end of the arrow) can extend to child use case (tailing end of the arrow) in some particular case, based on some special conditions.

4 References

N/A