SimSE Educational Game D:
Estimation with Non-Functional Requirements (NFR)
(Version: March 30, 2026)

Audience: Students

Editors:

Alain Abran

Key contributors:

Sion Israel Sion, Ling Guo, Jean-Marc Desharnais, Monica Villavicencio

Sponsors:

1. Introduction/Background

This SimSE game provides a set of scenarios with different levels of non-functional requirements (NFR). This simulation module is designed to teach students how to identify effort estimation ranges when considering the combined impact of distinct levels of NFRs.

2. How to Use This Module

This module is designed to be used as part of a course on software project management and how functional size and other effort drivers can be used in software projects estimation.

SimSE is intended to be used as a *complementary* component to a course, not as a standalone instructional tool.

Therefore, software functional size and software estimation steps and stages with additional effort drivers should be introduced to students either before, or in parallel with the students' exposure to the game (either through lectures (see Section 6), readings (see Section 7), or some other method).

SimSE's main strength lies in its ability to allow students to put concepts into practice that they otherwise would not have the opportunity to experience through other instructional methods.

Before students are given the assignment to play a SimSE game, it is imperative that they watch the SimSE Gameplay video tutorial. It is strongly recommended that they also watch the Explanatory Tool and Game Branching video tutorials as well. All video tutorials are available at http://www.ics.uci.edu/~emilyo/SimSE/downloads.html#Videos.

Our experience with SimSE has shown again and again how crucial the instructions a student receives in learning to play the game are to their success in learning from it.

These video tutorials have been designed to specifically highlight and address aspects of SimSE that are critical for students to understand for a maximally effective educational experience. Therefore, we suggest that you not only assign the students to watch these videos on their own time, but, if time and resources allow, show them in class as well, emphasize how important they are to watch, and also point them to the SimSE player's manual, available at:

http://www.ics.uci.edu/~emilyo/SimSE/downloads.html#Docs.

If time and resources further warrant, students should be required to attend a teaching-assisted (TA) -lead training session, in which they are shown the videos, given a printed player's manual and, and then allowed to try playing the game for a while with the TA (who should have already studied the manual and played SimSE themselves) available to answer any questions they may have.

Students should be given the questions to answer for this module (see Section 8) at the time they are asked to play the model. Having the questions to refer to while they play helps point them to some of the more subtle lessons encoded in the model, as well as provides you, as an instructor, with a way to assess whether or not they have completed the assignment and learned the concepts.

One could use this module as a mandatory part of a course, or else make it an extra-credit assignment. 5-10% extra credit is recommended. Certain prerequisites for this particular module apply (see Section 4).

1. Learning Objectives

The learning objectives are from a software management perspective. In this module the learning objectives are:

- Estimates should always correspond to effort 'ranges': this is in contrast to often seen 'single point' estimate). In the previous SimSE module, the regression model provided not a 'single point' estimate but an equation as well as information on the dispersion of the historical data from the equation line (e.g., the MMRE – Mean Magnitude of relative error). The regression model was based on a single independent variable: the functional size of the software requirements (for instance, sized in COSMIC Function Points). In this module, players will learn how take into account additional (here: non-functional requirements) and to use the regression information together to come up with an initial range of estimation values.

- Each distinct NFR can have its own impact on project effort, and it may vary across NFR levels (from 'low' to 'very high'), In this module 16 types of non-functional requirements (NFR), each with a corresponding level. In this SimSE module, one learning objective is learn one way to take these 16 NFR into account as a combined set, not individually.

- In this SimSE module 10 scenarios are provided, in which each of the 16 NFR have been pre-classified into one of 4 ordered levels (Low, Nominal, High and Very High). For each scenario, the players have to identify corresponding estimation ranges using the information provided by the regression model (equation, +, -MMRE and data points on the corresponding graphs).

4. Prerequisites

A student (e.g., a game player) should have a basic understanding of software engineering and management perspectives. (See Sections 6 and 7 of this document for ways to achieve this.)

For this game, participants must understand how multiple variables (e.g., the set of 16 NFR) may impact software projects efforts.

5. Time Commitment

The average time to play a single game is 10-15 minutes, but, of course, it is likely to take several iterations playing the game for the student to learn the concepts and be able to answer the questions. Participants should be given at least one week of out-of-class time to explore the game and answer the questions (see Section 8).

6. Suggested Supporting Lectures

The book 'Software Project Estimation' - by Alain Abran is recommended.

7. Optional Supplementary Readings

1. Navarro, E.O. *"The Fundamental Rules" of Software Engineering.* 2008 [Available from: http://www.ics.uci.edu/~emilyo/SimSE/se_rules.html].

8. Assignment

Instructions

Step1) Watch the SimSE video tutorials at:

http://www.ics.uci.edu/~emilyo/SimSE/downloads.html#Videos.

Step 2) Download the SimSE player's manual at:

http://www.ics.uci.edu/~emilyo/SimSE/downloads.html#Docs.

Note 1: Be sure to watch the video and read the manual carefully, as they will highlight several important things that will significantly help you in successfully playing SimSE and correctly answering the questions.

Step 3) Next, download the Measurement game at:

https://chairegestionproduitsprojetslogiciels.github.io/simse-software-estimation-capability/games/game-d-nfr-effort-range.html#downloads

The download consists of a "readme" text file and an executable game, which you can run by simply double-clicking on it. If you do not have the current version of Java installed on your machine, you will have the opportunity to install it when you try to run a game.

Questions

We recommend choosing approximately three of the following questions to use with this module.

1. Explain the COCOMO-like approach to classify effort drivers, such as non-functional requirements (NFR), and classifying them into levels, and next assigning them an effort impact factor.

2. What is the meaning of MMRE for software effort estimation purposes?

3. Why aren't the +MMRE and –MMRE lines parallel to the equation line?

4. What is the impact on effort when all NFRs are at nominal, low, and high levels respectively?

5. What is the impact on effort when 8 NFR are at a nominal level and the other 8 NFR are at a low level?

9. How to Use This SimSE Module with Other Modules

This module has been successfully used in conjunction with other effort estimation-related SimSE modules, making an assignment that consists collectively of modules and associated questions.

10. Other Notes

There are several other potentially effective uses for SimSE, most of which have yet to be fully explored:

- Have more advanced students modify an existing model (or build one from scratch, which should only be used with extremely advanced students) using SimSE's Model Builder tool and one of the existing models (available at http://www.ics.uci.edu/~emilyo/SimSE/downloads.html).
    - This has been tried, and results published in T. Birkhoelzer, E. Oh Navarro, and A. van der Hoek. *Teaching by Modeling instead of by Models*. Sixth International Workshop on Software Process Simulation and Modeling, May 2005 (available at http://www.ics.uci.edu/~emilyo/papers/ProSim2005.pdf).
- Our experience has suggested that an observer presence can have a positive effect on learning in SimSE. Although we have not tried this ourselves in classroom settings (only in controlled experiment settings), some suggested ways to try this are having students play SimSE in pairs or having them play SimSE in a lab setting while observed by an instructor or TA.
- Have students play in teams, especially teams that have also done, or are doing, a class project together. This can add both a collaborative aspect to learning and, if set up to be a competition between teams, can add a competitive aspect.
- Make the assignment mandatory, rather than optional or extra-credit, to increase motivation.
- Have students play in a lab setting, both to add a competitive aspect and to allow them to collaborate.
    - Keep in mind, however, that a lab setting generally does not provide enough time to play a game enough to be able to answer all the questions.
    - An appropriate approach might be to allow students to play the game first in a lab session (this would also allow them to ask any questions that may arise), and then let them complete the rest of their playing and question-answering out of class.
- If a project is also being done as part of the course, have students pick one or more of the SimSE models and write an essay on comparisons between the SimSE process model(s) and the one followed in their project.

11. Feedback?

If you have any comments, suggestions, feedback, or experience regarding this course module or SimSE in general, please send an email to Alain Abran<Alain.Abran@etsmtl.ca>

References

1. COSMIC. (2020). Early Software Sizing with COSMIC: Practitioners Guide.

2. COSMIC. (2020). Early-Sizing-Experts-Guide-May-2020-1. https://doi.org/10.13140/RG.2.1.4195.0567

3. Abran, A. (2015). Software project estimation: The fundamentals for providing high quality information to decision makers. *John Wiley & Sons.*