

## XC - SimSE

### Objectives

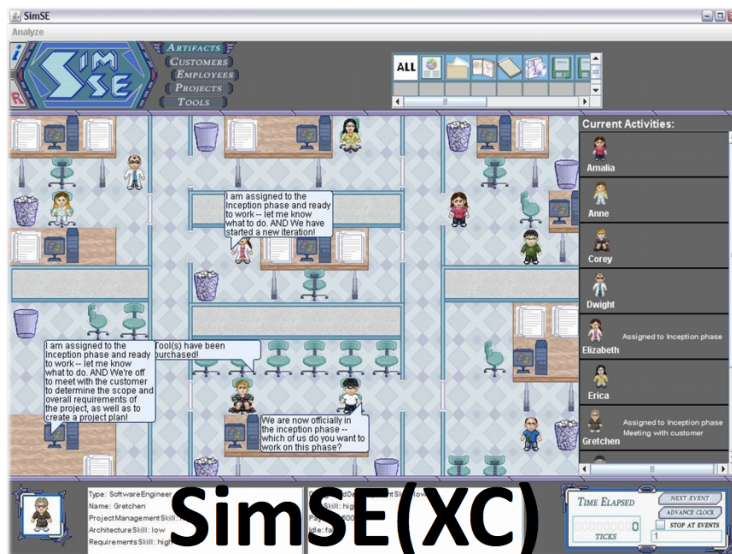
- Play project manager in SimSE

---

### PRE-LAB

Download and install the simulator. It requires an older version of Java, so this can get tricky.

---



### Steps

This has to be done during the lab as this is extra credit opportunity.

- If you get a 70 or up then show the TA to get extra credit.
- If you get less than 70, type out:
  - 1) What went wrong? Why did your project fail?
  - 2) What would you do differently next time?
  - 3) What was frustrating about the process?
- Note, you cannot get credit if you get a score of less than 30.

**To get credit for this lab exercise, show the TA and sign the lab's completion log.**

# Playing SimSE-Waterfall

## 1. Introduction

SimSE Waterfall is a game designed to simulate the management of a moderately sized software development project. You as the project manager must allocate your team of seven software engineers to develop a web-based grocery ordering system.

You manage your team of software engineers by giving them tasks. So long as the software engineer isn't on a break, or isn't sick, he or she will faithfully execute the tasks. Each software engineer, however, has varying proficiency at each task. Furthermore, when too many team members are allocated to a task, the performance of the team at the task drops.

Initially, you are given \$280,000 and 1350 time units for this project. However, the customer will change its mind and give you new requirements. When that happens, you will usually be given additional money and time to handle the new requirements.

SimSE Waterfall requires the Java Runtime environment version 1.5, 1.6, or 1.7 (does not work for 1.8. You can figure out which version of java by typing 'java -version'. Use either your computer or lab computes). To start SimSE Waterfall, either double-click on the file, or at the command prompt type

```
java -jar [name of jar file]
```

## 2. Software Engineer Characteristics

The performance of each software engineer is influenced by the following factors:

- Energy: Once this number hits 0.2, the engineer will take a break
- Mood: This influences the quality of the engineer's work
- Experience: This is the engineer's skill level at each particular task
- Pay rate: The amount of money the engineer earns. An engineer earns money regardless of whether the engineer is employed on a task.

## 3. Tasks

To assign a task to a software engineer, right click on the software engineer, and select the desired task. Software engineers can perform more than one task, but this reduces their energy at a faster rate.

Once tasks are assigned to engineers, select either "next event" or "advance clock" to have the engineer perform the task. If you select "Next event," the engineers will perform their tasks until some event occurs (e.g., an engineer takes a break, or a task is completed). If you select "advance clock," the engineer will perform the task for the stated number of time



units (default 1). Also, if you check the box “stop at events”, the clock will advance until either the number of time units stated have passed, or an event occurs, whichever happens first.

If no new tasks are assigned, the engineer will continue to perform the last stated task until the task is completed.

### 3.1 Common Tasks

There are four types of common tasks: (1) develop requirements, (2) develop design, (3) develop code, and (4) test system. Each software engineer has varying proficiency at each task.

You can monitor progress on each task by examining the artifacts created. Each artifact tells you:

- **PercentComplete:** The amount of progress you have made completing the task
- **NumKnownErrors:** The number of errors in the artifact that you know about.
- **PercentIntegrated:** The amount of code successfully integrated.

Each of these tasks has three to five subtasks:

- **Create:** A software engineer assigned to this subtask develops an initial version of the requirements, design, code, or system test plan. The quality and speed by which each deliverable is created depends on the number of software engineers allocated, and the quality of the individual engineers. If too many engineers are allocated to a task, they get in each others' way.
- **Inspect/Review:** A software engineer assigned to this subtask will review or inspect the created requirements, design, code, or system test plan for flaws. Note that code inspection can only be performed if at least three software engineers are simultaneously allocated to it.
- **Correct:** A software engineer assigned to this subtask fixes identified errors.
- **Integrate:** (This subtask is only for code). A software engineer assigned to this subtask synthesizes the code from disparate modules to build the final system.
- **Test system:** This subtask identifies bugs that were not discovered during code inspection.

### 3.2 Special Tasks

You can also opt to allocate a software engineer to the following additional tasks:

- **Give Bonus:** The software engineer will gain a temporary increase in mood.
- **Raise pay:** The software engineer will gain a permanent increase in mood.
- **Fire:** The software engineer will leave the project.
- **Deliver Product to Customer:** This ends the game and shows you your score.
- **Purchase tools:** The software engineer will acquire one or all of the following tools (your choice):
  - *RequirementsCaptureTool (SteelTrace):* This will speed up requirements capture and reduce the number of errors introduced into the requirements document. It costs \$10,000 to purchase this tool.

- *DesignEnvironment (RationalRose)*: This will speed up design and reduce the number of errors introduced into the design document. It costs \$5,000 to purchase this tool.
- *IDE (Eclipse)*: This will speed up coding tasks and reduce the number of errors introduced into the code. This tool is free to purchase (its available elsewhere in the company).
- *AutomatedTestingTool(JUnit)*: This will speed up testing and reduce the number of errors introduced into the system test plan. This tool is free to purchase (its available elsewhere in the company).

## 4 The Engineers

The following are the employees assigned to your project.

### 4.1 Andre



Energy: 0.97

Mood: 0.88

Requirements Experience: 10 years

Design Experience: 11 years

Coding Experience: 7 years

Testing Experience: 9 years

Pay Rate: 35.00

Andre is one of your two senior software engineers. He has been involved in countless projects, and is good at almost everything. However, he really shines at design. Andre is a very fast coder, but he tends to be sloppy in his coding practices.

### 4.2 Emily



Energy: 0.70

Mood: 0.80

Requirements Experience: 3 years

Design Experience: 5 years

Coding Experience: 6 years

Testing Experience: 1.5 years

Pay Rate: 30.00

Emily is the second of your senior software engineers. While she doesn't have much experience, she was earmarked early for her leadership qualities and is expected to be promoted to project manager soon. She has had very little contact with clients, and her experience has mainly been working with designers, and coders.



### 4.3 Anita



Energy: 0.70

Mood: 0.60

Requirements Experience: 8 years

Design Experience: 5 years

Coding Experience: 2 years

Testing Experience: 6 months

Pay Rate: 33.00

Anita is a real people person. She likes talking to the customer to find out what the customer wants. She has real problems with programming logic, however.

### 4.4 Mimi



Energy: 1.00

Mood: 0.80

Requirements Experience: 3 months

Design Experience: 5 months

Coding Experience: 3 months

Testing Experience: 8 years

Pay Rate: 20.00

Mimi is Anita's polar opposite. She's extremely caustic and critical, and doesn't work well with most people. As a result, most previous project managers only call on her to do system tests, a job she excels at.

### 4.5 Pedro



Energy: 0.48

Mood: 0.39

Requirements Experience: 7 years

Design Experience: 2 years

Coding Experience: 8 years

Testing Experience: 15 years

Pay Rate: 28.50

Pedro has had almost as much experience as Andre. However, his slothful habits, and lack of personality have hindered his growth in the company. Pedro also doesn't like to do design. He prefers to just code straight from the requirements document.



## 4.6 Calvin



Energy: 0.30

Mood: 0.60

Requirements Experience: 9 years

Design Experience: 8 months

Coding Experience: 6 years

Testing Experience: 2 weeks

Pay Rate: 32.00

Calvin is a rarity in the world of software engineering, a skilled programmer who doesn't have unlikable quirks. He spent six years as a programmer for other companies, and then left to spend another six as a systems analyst for Ernie the Young, the famous consulting company. He has been doing requirements analysis for your company for the last three years, and his performance appraisals suggest that he is the best one in your company at eliciting user requirements. However, Calvin doesn't have the energy to focus on a dedicated task for very long. He often takes sick leave, or long vacations when the project reaches its zenith.

## 4.7 Roger



Energy: 0.30

Mood: 0.80

Requirements Experience: Beginner

Design Experience: Beginner

Coding Experience: Beginner

Testing Experience: Beginner

Pay Rate: 10.00

Roger's a new kid, straight out of college. He's enthusiastic, but awfully naïve.

# 5 Tips

- Don't allocate too many people to a task
- It is better to allocate a highly skilled employee to a task than several employees who dislike or are unskilled at the task
- An overworked employee will take frequent breaks or get sick more easily.
- You can perform design, coding or testing even if requirements, design, or coding are incomplete. However, if the percentage of design, coding, or testing completed

is higher than requirements, design, or coding, you will have more errors in that artifact.

- Tools will help you finish your work faster, and more accurately. However, you need to keep in mind your project budget.

