

Distributed and Outsourced Software Engineering (DOSE 2010)

Assignment 2: Requirements Document

(30 points of 100)

First draft deadline: Tuesday October 26th - 8 am (Zurich time)

Final Deadline: Thursday November 4th - 8 am (Zurich time)

Commit the files at

https://svn.origo.ethz.ch/dose2010/requirements/group_n/team

where n is your group number and team is your team name.

The main task of this assignment is to develop a requirements document. Each team writes its own requirements document; we recommend that teams in the same group communicate to agree in common terminology. However, different documents have to be developed. Following we describe the tasks.

1. Questionnaire

Each team should have collected the time expended in Assignment 1. Please send this information by e-mail to Martin Nordio (if possible use the excel file you got) before **Friday Oct 15th**.

Important: the time expended is **collected by teams** and calculated in person/hours. For example, if 3 people participated in a meeting for 2 hours, then the expended time is 6 hours.

For assignment 2, you should also collect this information:

1) How many person hours did you dedicate to Assignment2?

a) How many person hours of Assignment 2 did you use for communication? Split the answer in two categories:

I. Communication with members of the same team (located in the same university)

II. Communication with members of other teams (located in other universities)

b) How many person hours of Assignment 2 did you use for development of the assignment result (preparing the documents, answering questions, writing documents, reading documentation, etc)?

2) From question 1.a):

a. How many hours were used for voice communication?

b. How many hours were used for other way of communication such as e-mail, forums,, etc. ?

c. How many hours did you use for face-to-face meetings?

3) How long did you have to wait for a reply from the other team? Indicate best case, worst case and average in hours.

2. Repository Structure: General Information

The dose2010 repository has the following structure:

```
https://svn.origo.ethz.ch/dose2010/assignments/  
https://svn.origo.ethz.ch/dose2010/requirements/  
https://svn.origo.ethz.ch/dose2010/requirements/group_1/TEAM1  
...  
https://svn.origo.ethz.ch/dose2010/requirements/group_11/TEAM3  
https://svn.origo.ethz.ch/dose2010/scope/  
https://svn.origo.ethz.ch/dose2010/scope/group_1  
...  
https://svn.origo.ethz.ch/dose2010/scope/group_11  
https://svn.origo.ethz.ch/dose2010/src/  
https://svn.origo.ethz.ch/dose2010/src/images  
https://svn.origo.ethz.ch/dose2010/src/skeleton  
https://svn.origo.ethz.ch/dose2010/src/group_1  
...  
https://svn.origo.ethz.ch/dose2010/src/group_11  
https://svn.origo.ethz.ch/dose2010/templates/
```

The **scope** directory contains the overall scope document for the course, and the subdirectories the individual scope documents of the groups. The **requirements** directory contains subdirectories for each group and team containing that team's requirements document. The **src** directory **contains all the source code** for the projects. It consists of the

main Eiffel project, a skeleton structure for the individual group projects and the source code of the individual language components.

3. EiffelStudio Project

All groups will be working on the same EiffelStudio project. The codebase provided by us handles the main GUI, which allows to launch a language component. To open the project, you can open the *language_learning.ecf* file in EiffelStudio. This will be the project you are working on during the whole course.

When you launch the project, a window will open where each group will add a button to launch their language component.

3.1 Tasks (this task has to be done by each group)

We provide you with skeleton classes which you can use to implement your language component. Each group should copy the *skeleton* directory and rename it to **group_n**, where n is your group number. The directory structure will then look like this:

```
src/images
src/skeleton/gui
src/skeleton/logic
src/skeleton/store
src/group_1/gui
src/group_1/logic
src/group_1/store
...
src/group_1/gui
src/group_1/logic
src/group_1/store
```

Each group will select a class prefix for their component. This is a two-letter code which you will use as a prefix for each class in your component. This is needed to avoid name clashes between the class names of the different groups. The skeleton classes use the one-letter prefix “S” for their classes. You have to rename all the class files and class names to match your selected prefix. Don’t forget to change the type of the window opened in the launcher class, otherwise you will open the skeleton window.

The last step is to add a button for your component in the main GUI. To do this, open the `MAIN_WINDOW` class (file `main_window.e`) and edit the “`Language_entries`” feature (at the bottom of the class). Copy the dummy entry and edit the information according to the comment in the class.

Since everybody is working on the same project, you should take extra care to make sure your code compiles before committing it to the repository. Also, write a log message for each commit!

4. Requirements Document

In this assignment each team has to write a requirements document for the subcomponent the team is responsible for. Each team is responsible for its own requirements document. However, we recommend communicating in the groups and using the same tool for writing. For example, a good idea would be to define the same acronyms, definitions and abbreviations within the group.

4.1 Templates

You can find templates for the requirements document at:

<http://se.ethz.ch/teaching/2010-H/dose-0273/index.html>

Furthermore, you can find examples of good requirements documents on the same page. Please consider IEEE recommendations given by IEEE Std 830-1998. Students from universities that do not have access to IEEE library please ask ETH students to send you the document.

4.2 First Draft

You have to commit a first draft by October 26th. **This version will not be graded but it is mandatory to commit the first draft. Not committing the first draft implies failing the course.** We recommend committing your document frequently so that other members of the group can see the document.

What is a first draft?

A first draft of a requirements document is a document where most of the sections have been described, and the most important requirements have been defined. However, the document still needs improvements such as typos, grammar, etc.