

Principles of Software Engineering
2805ICT: System and Software Design
3815ICT Software Engineering
7805ICT Principles of Software Engineering
Milestone One
(INDIVIDUAL SUBMISSION)
Weight 10%

Vladimir Estivill-Castro

Larry Wen

July 6, 2020

1 Objectives

1. Understand the differences between conceptual design and physical design.
2. Carry out some initial steps in the process to create conceptual designs.
3. Create appropriate system models for the representation of functional requirements by capturing use-cases.
4. Use a design paradigm to design a simple prototype of a software system.
5. Select an appropriate software architecture as the design basis for a given software requirements specification, justify the selection based on its advantages over alternative architectures.
6. Create software programs that make use of appropriate design patterns.
7. Create user interface software using either event-driven or callback based designs.
8. Elaborate prototypes building a GUI using a standard widget set.
9. Discuss the properties of good software design including the nature and the role of associated documentation.
10. Create appropriate design documentation for a variety of different designs.

2 Requirements

Read carefully the description of the overall project for this course (the *Pac-Man Game* project). Such description of the overall project was released earlier, although its deadline is close to the end of the term. This document describes the first milestone.

Then investigate what are *functional requirements*, *non-functional requirements* and *constraints*. You can start with Wikipedia's page on *functional requirements*: en.wikipedia.org/wiki/Functional_requirement¹.

Write a report that specifies at least 5 (**five**) functional requirements for the project with sufficient detail you can be confident they can be used for complete and comprehensive communication to a qualified software engineer.

Complement the description by providing for one "Use Case" a complete documented template with description, actors, basic flow and alternative flow. Complement the template with a "Use Case" diagram and also connect the other 4 use-cases.

Write a report that specifies at least five (**five**) nonfunctional requirements for the project with sufficient detail you can be confident they can be used for complete and comprehensive communication to a qualified software engineer.

Finally, write 3 (**three**) *constraints* with the same amount of clarity and detail.

3 Project risk

Completed projects should look like the Waterfall Model but in reality, the development process is always partly evolutionary. Risk of failure in a project is lowered by:

1. Prototyping key components
2. Dividing into phases
3. Following a visible software process
4. Making use of reusable components

Write at least 2 paragraphs about your planning of the overall project and how such planning considers each of these aspects. Use the digital textbook by Ivan Marsic (Rutgers University) to guide you with this. Look at Section 2.4.4.

4 Feasibility Report

A feasibility report is a written document

1. For a general audience: clients, stake-holders, financial management, technical management, for example.
2. Short enough that everybody reads it

¹Wikipedia is a tertiary source of information and knowledge. It is not a reliable source, but is a very good start for getting a sense of a concept.

3. Long enough that no important topics are skipped

Write a feasibility report about the *Pac-Man Game* project. Incorporate your analysis of functional requirements, non-functional requirements and constraints as an appendix and feel free to make cross-references to it.

5 Prototype

Construct a prototype. It should be possible to play *Pac-Man Game* with your program, using a GUI and perhaps using some IDE. Maybe only one first version of the game, with limitations, like only square cells, only one size, but sufficient to make you evaluate the challenges the overall project represents.

6 Conceptual Design

We do not anticipate a full conceptual design will be possible by the due date of Milestone One; however, we expect you to start the discussion of the main classes and objects you expect to generate along with your project. Nevertheless, we expect you to investigate and complete the following two preliminary tasks regarding a conceptual design.

You may work iteratively for this activity as well as the activity of developing a prototype. Make sure that the final submission can compile and that you can generate automatic documentation for your prototype by using a tool for automatic documentations (see for example `doxygen`² to generate documentation from your prototype).

Classes and objects

You should create an analysis of your project requirements and its description and come up with a list of software classes and potentially an UML class diagram.

Use cases

You should capture the interaction of users and the scenarios that provided added value to a user by drafting the use-cases of this project.

7 What do you need to submit

You can collate all PDF documents into one or make separate submissions.

1. PDF document with results of requirements analysis. In particular
 - (a) specification of 5 (five) functional requirements. One of these functional requirement should be complete and presented as a use-case with documented template and diagram.

²www.stack.nl/~dimitri/doxygen/

- (b) Also specification of five non-functional requirements.
 - (c) Specification of three constraints.
2. PDF document with at least 2 paragraphs of project risk report.
 3. PDF document with a feasibility report.
 4. Link to a video showing a working *Pac-Man Game* prototype.
 5. PDF document reflecting on the GUI design.
 6. PDF document with a conceptual design that includes two parts.
 - (a) An object model presented as a class diagram of initial relevant classes/objects for the project.
 - (b) A cross-reference to the use-cases that are presented as result of requirement analysis in the first item above.

8 Marking Criteria

This assignment is allocated 10 marks as follows.

Quality of functional requirements	1 mark
One use-case comprehensibly presented	1 mark
Quality of non-functional requirements	1 mark
Quality description of constraints	1 mark
Project risk report	1 mark
Feasibility report	1 mark
Working prototype of GUI	2 mark
Reflection on GUI and one prototype class documented with doxygen	1 mark
Clarity of the “Object model”	1 mark