

Principles of Software Engineering  
2805ICT  
3815ICT  
7805ICT  
Milestone Two  
(INDIVIDUAL SUBMISSION)  
Weight 10%

Vladimir Estivill-Castro  
Larry Wen

July 22, 2020

## **1 Objectives**

1. Practice reporting the progress on a software development project.
2. Describe a testing plan and testing performed within the current progress.
3. Use the UML as a tool to describe and specify software designs, and be able to attach semantic meaning to UML constructs by establishing relationships with the code the artifact models.
4. Discuss software design patterns meaningfully.
5. Review the potential evolutions that a software project may have regarding software architecture and software design paradigms.
6. Model dynamic behaviour of software systems using suitable UML notation.

7. Describe the progress concerning user interface design.
8. Review metrics regarding progress by reviewing historical data on version control system.
9. Discuss the properties of good software design including the nature and the role of associated documentation.
10. Create associated descriptions to document UML diagrams.

## **2 Progress Report**

In this milestone, you are to produce a progress report on your project. This progress report should evaluate the progress made towards completing a software for Pac-Man as specified in the global documentation.

Using the approximation that 1 page is 500 words single-spaced, or 250 words double-spaced, your progress report should be 2,000 words excluding diagrams, figures and references (citations).

The progress report should have the following sections.

1. Requirements. A summary of those requirements that are completed and working in the current version and a prioritized list of those requirements still on-going.
2. Product Use Cases. A description of the use-cases relative to the requirements. Potentially a classification of completed use-cases and pending use-cases. It may be that some scenarios in some use-cases are completed but some not.
3. Summary of software architecture being used and its feasibility to remain the same until project completion. Describe the main programming language(s) used, tools used, platforms used. Global architecture of modules and components.
4. Summary of design. Describe design goals, and provide artifacts of the design: Class Diagrams, Dynamic Model, Subsystem Decomposition.
5. Discussion of the level of sophistication regarding Persistent Data management, Access control and security, and User Interface.

6. Describe what sections of the code have been tested, what testing plan has been performed and what testing will be performed in the future. Describe how testing is used to ensure the introduction of new functionality does not regress with respect to previous functionality.
7. Extract logs of version control commits and create a historical analysis, perhaps with a histogram of effort across time dedicated to the development of the software that represents the final milestone.

### **3 Putting UML to use.**

Ensure that for your progress report you lift out of your working prototype,

1. your design and
  2. the associated documentation.
- . You should include in your report the following items (it is very important that you show in the submission of this milestone the correspondence between each diagram and your code).

1. For Dynamic Modelling
  - (a) Interaction diagrams describing the behaviour between objects
    - i. At least one sequence diagram with the set of participating objects arranged in sequences columns and time flowing from top to bottom.
    - ii. At least one collaboration diagram, perhaps showing the sequence of events with numbering labels.
  - (b) State-based diagrams.
    - i. At least one state machine that describes the flow of states for a given class in response to the outside stimuli.
    - ii. At least one activity diagram, where all the states are actions states (sometimes called a Moore Automaton).
2. For Structural (static) Modelling
  - (a) A class diagram.

3. For the use of Software Patterns.

- (a) Describe at least one implementation of a software pattern illustrated by some existing code in the current version of the project or prototype.

**IMPORTANT:** It is part of the task to provide descriptions and explanations that link the diagrams to the code, or to provide templates for the diagrams that ensure the diagrams are comprehensive models of the aspect for which they are a representation.

## 4 What do you need to submit

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

1. PDF document with the progress report. We recommend that you refer to Milestone 1.
2. The collection of links between code snippets and diagrams. We do not need your full code or any program. We will not be installing and executing your code. You must take screen-shots or insert the code into the PDF that also includes the diagram, so the correspondence between modelling tool and program source-code is transparent.

## 5 Marking Criteria

This assignment is allocated 10 marks as follows.

Quality of report regarding progress on functional requirements	1 mark
Quality of report on progress of implementation of use cases	1 mark
Comprehensive review on software architecture	1 mark
Review of progress on design	1 mark
Review of testing and further testing plan	1 mark
Analysis of major issues and productivity	1 mark
Dynamic Modelling diagrams and illustration in code	2 mark
Static modelling and illustration in code	1 mark
Illustration of one Software pattern	1 mark