

**ICS 52: Introduction to Software Engineering
Fall 2004**

Instructor: Dr. Richard N. Taylor

TA: Justin R. Erenkrantz

Assignment 1: Requirements

Issued: Monday, October 4th, 2004

Due: Monday, October 18th, 2004 (beginning of discussion)

Requirements Specification Assignment

The electronic gaming industry is undergoing a phase where “what is old is new again.” There is a substantial market for games that were released decades ago and exposing them to a new class of users. In this vein, Phoenix Games has decided to resurrect an old video game called *Klax* in time for the upcoming holiday season. As it turns out, one of Phoenix’s lead developers wrote a prototype of this game a few years ago. However, that developer was subsequently hired away by one of Phoenix’s larger competitors. Unfortunately, that developer didn’t believe in software engineering practices and did not leave any usable artifacts besides the source code.

Since Phoenix has only a few months before the holiday season begins, they are looking for an accelerated development cycle with a target for an early-December release. Yet, management has learned their lesson and has now bought into the software engineering process: they are mandating the creation of documents to fully explain the requirements, design, and code in order to assist new developers who may have to later work on this game. Phoenix is looking for requirement specifications, detailed design and architecture documents, extensive test cases, and finally they’ll want to discuss some additional features that will attract new players to this game.

The first deadline is **Monday, October 18th** and Phoenix is looking for a detailed requirements specification documents that fully describes what the game **should** do. In order to help determine what the game should do, Phoenix has provided you with the source code from the previous developer. You may run the software and “reverse-engineer” the requirements. Some employees remain at Phoenix who know how to run the software, but know little else about the requirements. Your project manager (aka TA) will hold twice-weekly meetings to guide you through the use of the initial prototype and the creation of the requirements document.

The requirements specification should specify the information that will be used to evaluate whether or not the final product meets the requirements. It should be concise but with sufficient detail to resolve any potential misunderstanding between you and Phoenix management. This document should not specify implementation details, such as the data structures and algorithms that will be used to implement *Klax*; **Phoenix is only currently concerned with what *Klax* does from the user’s perspective and not how it does it.**

Structure of the Requirements Specification

The Requirements Specification document should have the following structure:

1. Table of Contents

- Listing all sections and their page numbers

2. Introduction

- What is this document about?
- Who was it created for?
- Who created it?

3. Executive Summary

- A short introduction of the Klax software
- What is the purpose of this software?
- Identify the high-level objectives
- Briefly characterize how the user interacts with the system
- Identify the important processes, hardware, and software aspects of the system

4. World Model & Functional Requirements

- Introduce a high-level “World Model” abstraction that characterizes the system properties and behavior
- How might this world model change in the future?
- Identify all concepts, functions, features and information that the system provides to its users
- Provide an abstraction for each of those concepts, characterizing their properties and functions that are relevant to the user:
 - What is the system supposed to do?
 - What information does the system need?
 - What is supposed to happen when something goes right?
 - What is supposed to happen when something goes wrong?

5. Environmental Requirements

- Platforms
 - Hardware - Operating systems, types of machines, memory size, hard disk space
 - Software – Development environments, run-time environments
- Programming Languages

6. Potential Risks

- Determine the risks faced by the project.

7. Future Changes

- Determine the changes that the software will undergo in the future including potential future enhancements

8. Acceptance Test Plan

- Organization of tests
- Format of test cases

9. Definitions of Terminology

- Precise definition of terms used throughout the project

10. Reference documents

- Pointers to existing literature and tools
- Pointers to other similar software

Submission Criteria

You **must** submit a hard copy at the beginning of discussion that follows these guidelines, and you **must also** submit an electronic copy to the course EEE DropBox.

Printed Criteria

The printed copy of your requirements specification that you turn in for credit **must** include

- Page numbers at the bottom of each page
- Use proportional font similar to either Times (New Roman) or Arial.
- Major section headings in 14 point, subsection headings in 12 point, and body text in 12 point.
- One inch margins around the page
- Stapled once in the upper left hand corner, no binders, no plastic covers.
- A title page using a 18 point font with the following text centered vertically and horizontally:

Klax Requirements Specification

by:

Your_First_name Your_Last_name

{UCINetID: i.e. panteater@uci.edu}

ICS 52

Instructor: Dr. Richard N. Taylor

Fall 2004

Electronic Criteria

You must submit your electronic version of your requirements document via the EEE DropBox entitled: **ICS 52 HW #1**. The documents **must** be compressed via ZIP before submission. Word or PDF documents accepted. Other formats accepted by **prior arrangement only**.

Grading

The grading of this assignment will be broken down as follows:

50%. Accuracy (clarity, precision, completeness, and consistency of the specification)

25%. Presentation (organization and consistency of the document)

25%. Composition (spelling and usage of English)

This assignment counts **10%** towards your final grade for the course.

Note:

- Do not work in teams to complete this assignment
- No late assignments will be accepted