PROJECT PLAN

RETRO ROBOTS

**Contents**

[1.0 Scope 3](#_Toc106477823)

[1.1 Project Objectives 3](#_Toc106477824)

[1.2 Life Cycle Description 3](#_Toc106477825)

[1.3 Work Products 5](#_Toc106477826)

[2.0 Organizational Structure 5](#_Toc106477827)

[2.1 Team Structure 5](#_Toc106477828)

[2.2 Work Breakdown Structure 6](#_Toc106477829)

[2.3 Project Schedule 7](#_Toc106477830)

[3.0 Risk Assessment Plan 8](#_Toc106477831)

[4.0 Quality Plan 8](#_Toc106477832)

[4.1 Test Plan 9](#_Toc106477833)

[4.1.1 Purpose of Test Plan 9](#_Toc106477834)

[4.1.2 Test Risks 9](#_Toc106477835)

[4.1.3 Test Methodology 9](#_Toc106477836)

[4.1.4 Test Deliverables 9](#_Toc106477837)

[4.1.5 Test Environment 9](#_Toc106477838)

[4.1.6 Test Suspension / Resumption Criteria 9](#_Toc106477839)

[4.1.7 COMPATIBILITY Testing 10](#_Toc106477840)

[Items to Test 10](#_Toc106477841)

[Test Pass / Fail Criteria 10](#_Toc106477842)

[Test Entry / Exit Criteria 10](#_Toc106477843)

[4.1.8 Conformance Testing 10](#_Toc106477844)

[Items to Test 10](#_Toc106477845)

[Test Pass / Fail Criteria 11](#_Toc106477846)

[Test Entry / Exit Criteria 11](#_Toc106477847)

[4.1.9 Functional Testing 11](#_Toc106477848)

[Items to Test 11](#_Toc106477849)

[Test Pass / Fail Criteria 12](#_Toc106477850)

[Test Entry / Exit Criteria 12](#_Toc106477851)

[4.1.10 Performance Testing 12](#_Toc106477852)

[Items to Test 12](#_Toc106477853)

[Test Pass / Fail Criteria 12](#_Toc106477854)

[Test Entry / Exit Criteria 12](#_Toc106477855)

[4.1.11 Regression Testing 12](#_Toc106477856)

[Items to Test 12](#_Toc106477857)

[Test Pass / Fail Criteria 12](#_Toc106477858)

[Test Entry / Exit Criteria 12](#_Toc106477859)

[4.1.12 System Testing 13](#_Toc106477860)

[Items to Test 13](#_Toc106477861)

[Test Pass / Fail Criteria 13](#_Toc106477862)

[Test Entry / Exit Criteria 13](#_Toc106477863)

[4.1.13 Unit Testing 13](#_Toc106477864)

[Items to Test 13](#_Toc106477865)

[Test Pass / Fail Criteria 13](#_Toc106477866)

[Test Entry / Exit Criteria 13](#_Toc106477867)

[4.2 Configuration Management Plan 13](#_Toc106477868)

# Scope

This document contains the Project Plan established to develop the software Wheel of Jeopardy v. 1.0

## Project Objectives

The objective of this project is to provide an implementation of the game Wheel of Jeopardy. This project will deliver 4 software increments, defined as skeletal, minimum, target and dream as defined below.

## Life Cycle Description

The software will be developed utilizing the iterative/incremental life cycle. See fig.1

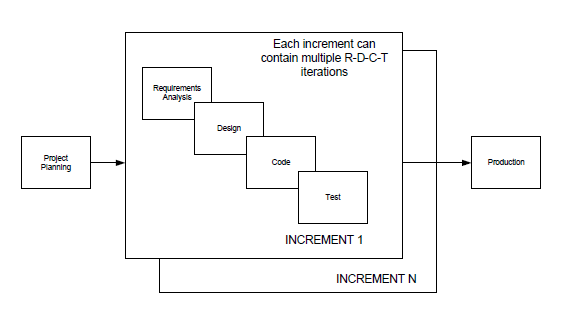


Fig. 1 Software Life Cycle Model

Source: Module 2 Slides

**Features of Four Increments**

1. **Skeletal Increment**

* Subsystems
  + Players
  + Game logic
  + GUI
* Features
  + Take player input for questions and answers and process
  + Read local files to retrieve questions
  + Display questions with given category in order
  + Update(answer correct/incorrect) and track scores
  + Opponents chooses the category
  + Counter to track remaining spins in the round

1. **Minimal Increment**

* Subsystems
  + Database
  + Server
  + GUI
* Features
  + Player sends request to server
  + Retrieve information from database

1. **Target Increment**

* Subsystems
  + Database
  + Server
  + GUI
* Features
  + A functional playable game
  + Satisfy all requirements outlined

1. **Dream Increment**

* Allow user to choose certain categories for the whole game
* Animation of the wheel and sound effect
* A version in another language, e.g., Chinese
* Save user’s account information to allow multiple logins

## Work Products

The following deliverables will be produced

1. Vision Document
2. Software Requirements Specification
3. Software Design Document
4. Skeletal System Demo
5. Minimum System Demo
6. Target System Demo
   1. **Milestones**

The following milestones are planned.

| **Milestone** | **Expected Date** |
| --- | --- |
| **Planning Complete** | 06/28/2022 |
| **Software Requirements Complete** | 07/12/2022 |
| **Skeletal Increment Complete** | 07/19/2022 |
| **Software Design Complete** | 08/02/2022 |
| **Minimum Increment Complete** | 08/09/2022 |
| **Target System Complete** | 08/23/2022 |

Table 1. Milestones

# Organizational Structure

## 2.1 Team Structure

**Project Manager:** Tatiana Correia

**Lead GUI Developer:** Keegan Riley

**Lead Software Architect:** Wenjun Zhou

**Lead Tester:** Nick Champagne

**Lead Configuration Manager:** Keegan Riley

**Lead Quality Manager:** Tatiana Correia

## 2.2 Work Breakdown Structure

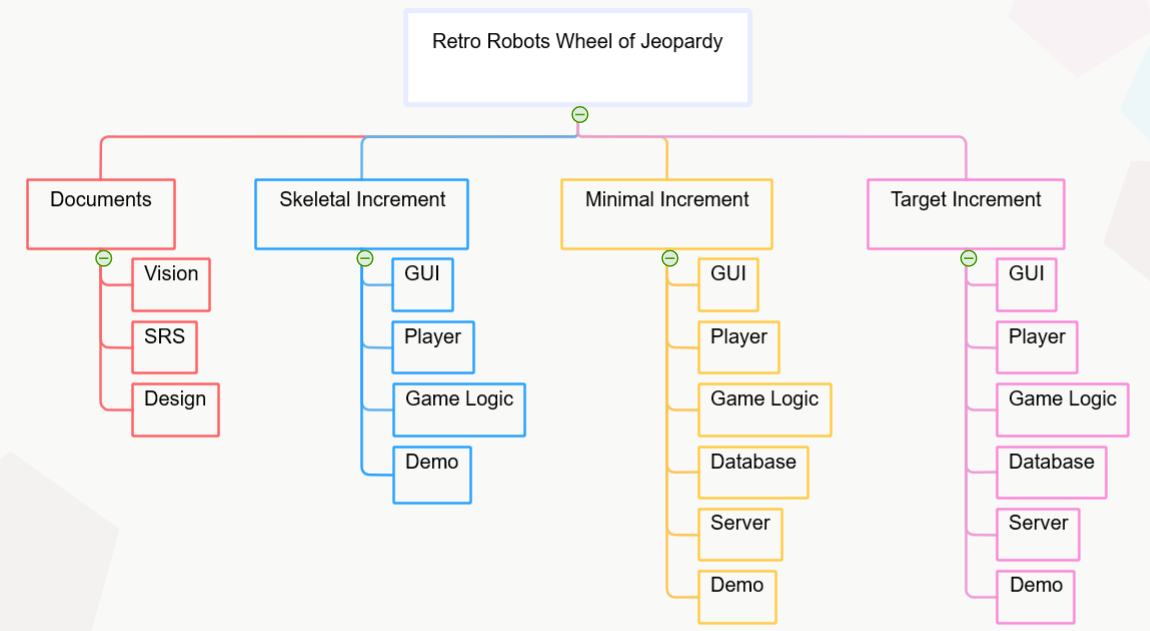


Fig. 2 WBS

## 2.3 Project Schedule

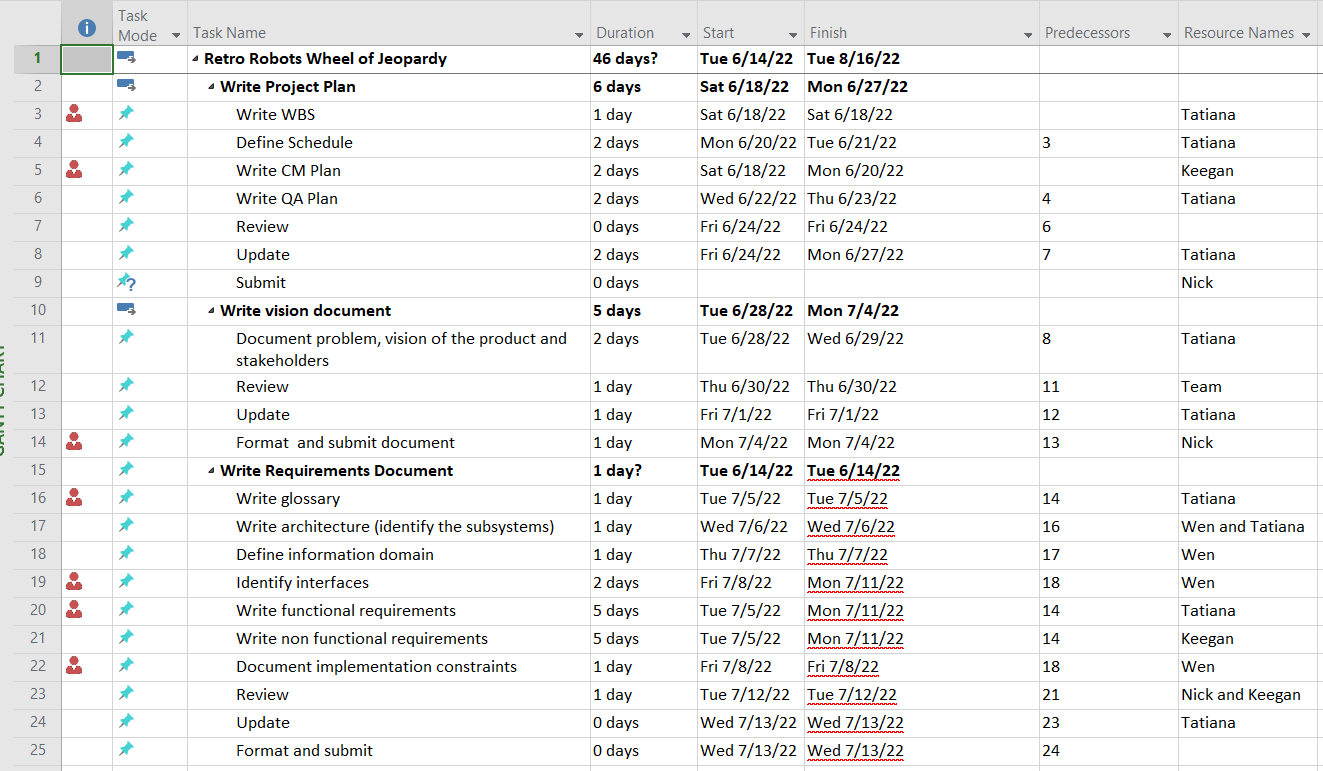






Fig. 3 Schedule

# Risk Assessment Plan

The main risk for this project is schedule risk. Schedule risk is caused by inadequate estimates, inaccurate assumptions, delay in activities caused either by a team member assigned to other projects, personal emergencies or by a delay in a predecessor activity. Schedule risk mitigation consists in periodic meetings to monitor and control the schedule and assignment of more than one team member to critical tasks.

# Quality Plan

Quality assurance processes will consist of reviews, dynamic tests and configuration control. Reviews will be conducted by independent reviewers.

Software Requirements shall be reviewed according to the following criteria:

* Unambiguous
* Testable
* Complete
* Consistent with other requirements

Software Design Document shall be reviewed according to the following criteria:

* Modularity
* Low coupling
* High cohesion
* Encapsulation

Team will adopt the following coding standard:

Software code will be tested dynamically, per below test plan.

Integrity of builds and baselines will be maintained per below configuration management plan.

## 4.1 Test Plan

## 4.1.1 Purpose of Test Plan

This software test plan outlines testing order, goals, and responsibilities in accordance with well-defined project requirements.

## 4.1.2 Test Risks

The primary testing risk is delay of project milestones and deliverables, which is possible in two ways. The first is the risk of over testing or going beyond the reasonable amount of testing required to verify software behavior. The second is delayed identification of problems and faults that might slow the software development lifecycle (SDLC).

## 4.1.3 Test Methodology

The Retro Robot team SDLC for the Wheel of Jeopardy project will be staged delivery, and the testing technique will be specifications-based. A finite number of test cases will be selected to verify software behavior. Software testing will occur throughout the SDLC and will be refined throughout as well. Periodic reports will be provided to programmers to aid in identifying and preventing problems and faults.

For the categories listed below, no items will be excluded from testing. The Retro Robots team will be responsible for testing the Wheel of Jeopardy software prior to delivery, and the effort will be led by Nick Champagne, Lead Tester. If the team does not select anyone else (or no one else volunteers) to perform testing, the responsibility falls to Nick Champagne, Lead Tester.

## 4.1.4 Test Deliverables

Given the test plan will be revised throughout the SDLC, additional test deliverables may be added. Test plan deliverables include but are not limited to:

1. Weekly or bi-weekly testing reports (depending on deadlines)
2. Milestone testing reports for each stage of project delivery
   1. Skeletal stage
   2. Minimal stage
   3. Target stage

## 4.1.5 Test Environment

Testing environments include Eclipse and IntelliJ integrated development environments (IDEs) on Windows 10 or Windows 11 operating systems.

## 4.1.6 Test Suspension / Resumption Criteria

All types of testing will not be suspended for any reason and will continue throughout the SDLC.

## 4.1.7 Compatibility Testing

### **Items to Test**

Items that will be tested include:

* Microsoft Windows 10 operating system
* Other operating systems?
* Hardware?

### **Test Pass / Fail Criteria**

### Compatibility tests will pass if the software behaves as expected on the hardware and software items outlined.

### **Test Entry / Exit Criteria**

### Compatibility testing will begin towards the end of the skeletal stage and conclude at the end of the target stage.

## 4.1.8 Conformance Testing

### **Items to Test**

Items that will be tested include:

* User Interface
  + Player’s turn
  + Spin button
  + Spin counter
  + Player scores
  + Wheel
  + Game board
* Game Logic
  + Player Turns
    - “Lose turn” sector
    - “Free turn” sector
    - “Spin again” sector
    - “Player’s choice” (category) sector
    - “Opponent’s choice” (category) sector
  + Scoring
    - Add points to player score for correct answers
    - Subtract points to player score for incorrect answers
    - Store score for first round during second
  + Game Wheel
    - 12 Category sectors (placed randomly) – two for each category
    - One “Lose turn” sector
    - One “Free turn” sector
    - One “Bankrupt” sector
    - One “Player’s choice” (category) sector
    - One “Opponent’s choice” (category) sector
    - One “Spin again” sector
  + Rounds
    - Two rounds per game
    - Double point rewards in second round
    - Add first and second round scores at the end of the second round
    - Highest total score wins game
  + Spin Counter
    - Increment spin count for each spin
    - Maximum 50 spins per round
  + Game Board
    - Multiple choice questions
    - Six categories
    - Five questions per category

### **Test Pass / Fail Criteria**

### Conformance tests will pass provided software functionality meets project requirements.

### **Test Entry / Exit Criteria**

### Given the specifications-based approach to testing, conformance testing will begin at the start of the SDLC and will govern all other forms of testing. Conformance tests will conclude at the end of the target stage of the SDLC.

## 4.1.9 Functional Testing

### **Items to Test**

Items that will be tested include:

* Interaction between the wheel and game board
* Interaction between the spin button and the wheel

### **Test Pass / Fail Criteria**

### Functional testing will pass upon confirming expected behavior between software units and will fail if interaction between units does not meet project requirements.

### **Test Entry / Exit Criteria**

### Functional testing will begin at start of the SDLC and conclude in tandem with conformance testing at the end of the target stage.

## 4.1.10 Performance Testing

### **Items to Test**

Items that will be tested include:

* Graphical user interface (GUI) responsiveness
* Anything else?

### **Test Pass / Fail Criteria**

Performance testing will be qualitative in that the tests will pass if software behave feels responsive from a human perspective, and they will fail if the team does not unanimously affirm responsiveness. Given the software will not perform intensive calculations, there is no need for runtime or space complexity performance testing.

### **Test Entry / Exit Criteria**

Performance testing will occur throughout all stages of delivery as new units of code are added.

## 4.1.11 Regression Testing

### **Items to Test**

Items that will be tested include:

* All test cases that passed for all previous stages of the SDLC at the minimal and target stages of delivery.

### **Test Pass / Fail Criteria**

Regression testing will pass so long as all test cases that passed for the previous stage of delivery pass for subsequent stages of delivery. Regression testing will fail if any test case that passed for a previous stage of delivery fails in a subsequent stage of delivery.

### **Test Entry / Exit Criteria**

Regression testing will begin as soon as the minimal stage of delivery is started. The first round of regression testing will end upon successful testing of all skeletal stage test cases at delivery of the minimal stage in the staged delivery SDLC. Similarly, the second round of regression testing will begin at the start of the target stage of delivery and will end prior to target stage delivery. All regression testing ends prior to final unit, functionality, and conformance testing.

## 4.1.12 System Testing

### **Items to Test**

Items that will be tested include:

* Database
* Server

### **Test Pass / Fail Criteria**

System tests will pass if ?

### **Test Entry / Exit Criteria**

System testing will begin during the minimal stage of delivery and will end in the target stage before the final conformance testing.

## 4.1.13 Unit Testing

### **Items to Test**

Items that will be tested include:

* All classes and class methods

### **Test Pass / Fail Criteria**

Unit testing will pass provided individual unit tests produce the desired software behavior and conform to project requirements (i.e., conformance testing also passes). Tests will fail if either the individual unit does not produce the desired software behavior or does not pass conformance testing.

### **Test Entry / Exit Criteria**

Unit testing will begin at the start of the staged delivery SDLC and end in tandem with final conformance testing at the end of the target stage.

## 4.2 Configuration Management Plan

# 