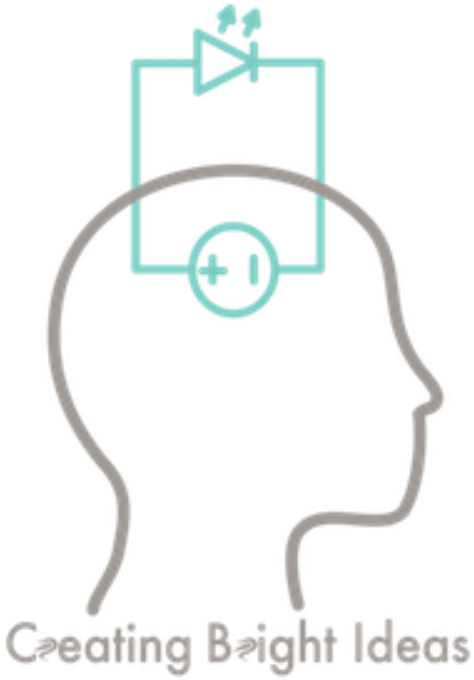


Project Plan

TEAM BRIGHT IDEAS

KEVIN HICKS;FIONA LYONS;JENNIFER WALKER

June 26, 2018

605.601.83.SU18

Foundations of Software Engineering

Work Breakdown Structure

WBS

Features

Skeletal

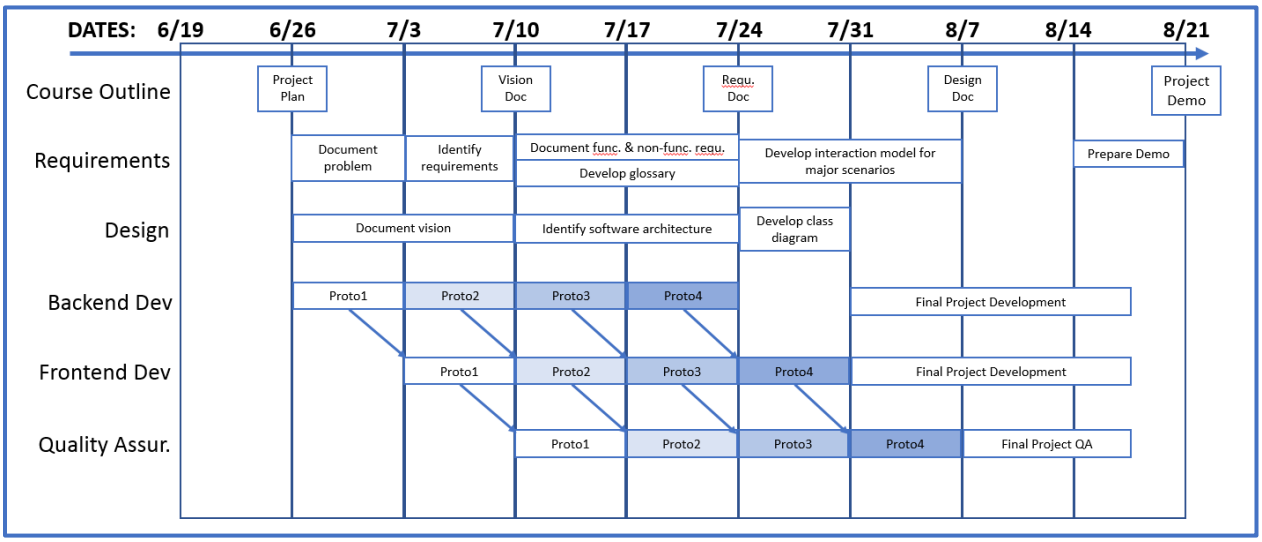
Minimal

Target

Dream

Schedule

For the development of this software product, Team Bright Ideas will be utilizing a rapid prototyping development model. Since there are so many subsystems in Wheel of Jeopardy, major functionalities will be prototyped then scrapped so that the programming approach will be understood, but we will not have to rely on code that is potentially uncooperative with other subsystems. The schedule is displayed below.



The backend development encompasses all Java development for the core code base. All inputs and outputs will be text based, and the code should account for all test cases. The frontend development is for the JavaScript development for the GUI of Wheel of Jeopardy. This portion will design GUI-based input and output, and will marry up with messages passed from the backend code.

Once frontend and backend prototypes have been developed, quality assurance and testing will be performed in order to ensure the prototypes meet requirements. This portion of the development cycle will test all relevant use cases, and provide feedback to the frontend and backend development teams for improvements that should be made for the final product.

The four prototypes are as follows:

- Prototype 1: I/O interface. This prototype includes input file format for questions, assignment of questions to categories, and assigning values to questions. Once data is loaded in, this prototype will allow for basic wheel spinning and question selection.
- Prototype 2: Score keeping. This prototype includes the basic flow of Wheel of Jeopardy, including awarding and displaying points, gaining/losing points, switching rounds, double jeopardy and bankruptcy.
- Prototype 3: Player interaction. This prototype includes selecting questions, correctly/incorrectly answering, taking turns, and switching between rounds.
- Prototype 4: Final features. This prototype is for any limiters such as the timer and maximum number of turns. This will also account for smaller subsystems such as starting the game, finishing and declaring a winner, and randomizing the daily double spot. This will also add in any feedback from the quality assurance and testing teams for additional features that may have been previously overlooked.

Risk Assessment Plan

Quality Plan

Quality Assurance

Testing

Testing will be done by each developer as a portion of the code is completed before it is shared with the rest of the team. The testing will aim to test general use cases, and corner cases which are still within the bounds of the stated requirements.

Configuration Management

Project configuration management will be performed through the use of GitHub, which retains a repository of past versions. Each user will be able to modify the code in a way that will not interfere with each other, then publish completed changes. Before sharing code, the user should thoroughly test it and write a detailed description of the code before sharing.

In the case that something is accidentally shared to the repository, or that a portion of the code breaks unexpectedly, the lead configuration manager will halt the repository. The repository will then be reverted to a previous version, and changes incrementally added back in until the problem is identified and can then be fixed. Any problems should be reported to the group as a whole to reduce risk of losing code.