

Project Plan

TEAM BRIGHT IDEAS

KEVIN HICKS;FIONA LYONS;JENNIFER WALKER

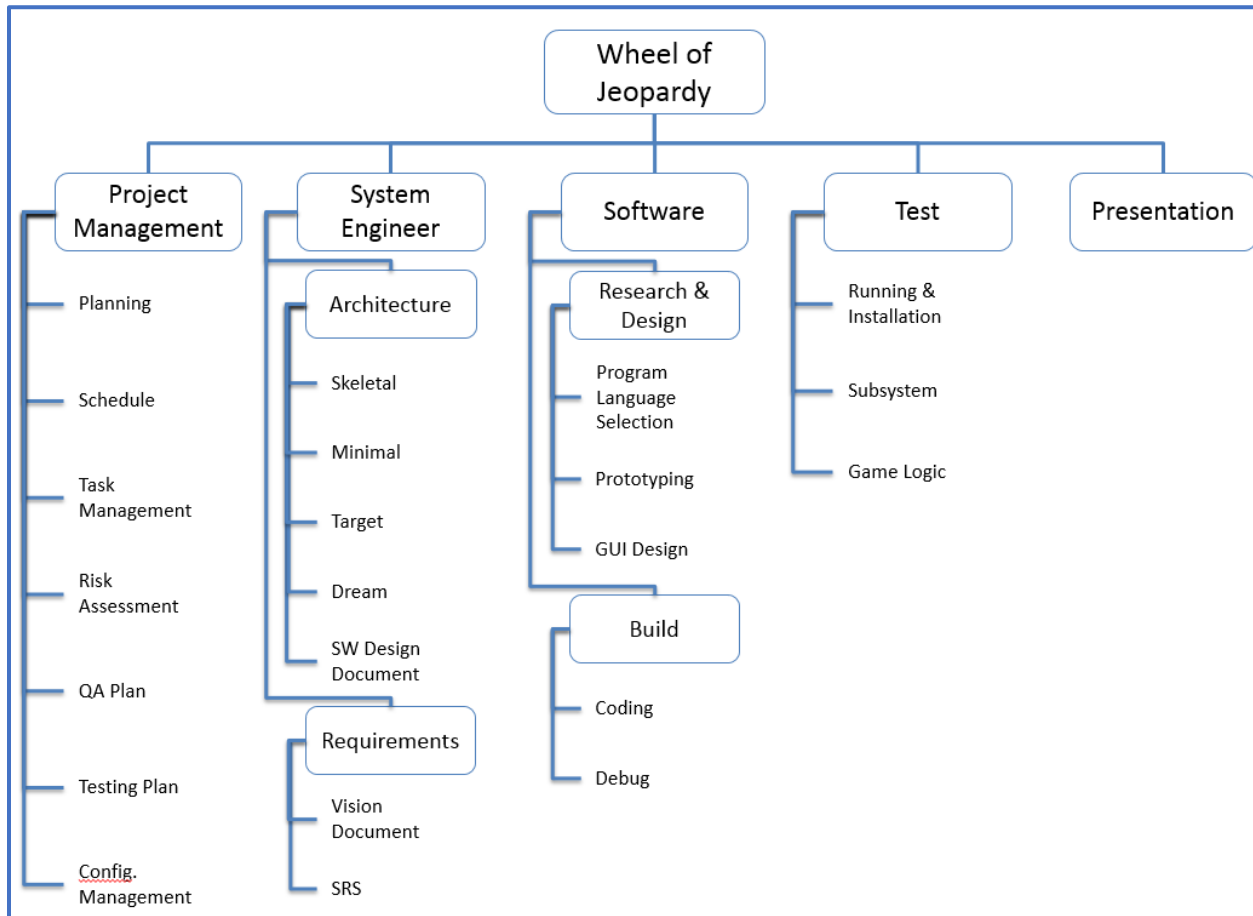
June 26, 2018

605.601.83.SU18

Foundations of Software Engineering

Work Breakdown Structure

The following is the work breakdown structure for Team Bright Ideas.



Features

Skeletal System

The Skeletal System outlines the major functionalities of Wheel of Jeopardy, and what should be considered as the underlying architecture for any further improvements.

Features

- Wheel
 - Players spin the wheel in order to generate questions or enable other options in accordance with the rules of the game
 - Consists of:
 - 12 Sectors

- One sector for each of the six categories on the Jeopardy board (6)
 - “lose turn” sector (1)
 - “free turn” sector (1)
 - Bankrupt sector (1)
 - Players choice sector (1)
 - Opponents choice sector (1)
 - Double score sector (1)
- Spin button/option to rotate the wheel by player
- Round indicator (only one round for this increment)
- “Spins” indicator to inform the user how many spins are left in the round
- Scoreboard that indicates the scores of the current player
- Question Board
 - The question board will display the categories and associated questions for the players.
 - Consists of:
 - 6 categories for questions
 - 5 questions per category
 - Questions are ordered by point value and level of difficulty. The easiest question per category is worth 200 points and increases incrementally by 200 up to 1000
 - Text box that displays the question that is selected
- Answer Board
 - Consists of answer button/option
- Scoreboard
 - Consists of text box for each player that contains player’s name and score

Interactions

- Wheel
 - Designated player spins the wheel then wheel lands on 1 of 12 sectors
 - After a pre-determined time period the resulting sector of the wheel spin will cause either the appropriate question board to display according to the category, or the consequence of the sector landed on will ensue if not a category (e.g., bankruptcy, lose turn, free turn, etc...)
 - The number of spins updated after every spin
- Question Board
 - Categories and questions display based on sector designated by the result of the wheel spin
 - Questions are to be selected in order of increasing point value and difficulty
- Answer Board
 - Correct answer is displayed then return to wheel subsystem
- Scoreboard
 - The scores for the players will increase or decrease based on wheel

- Selection of a category will automatically increase score
- Players
 - 2 player option
- Milestones
 - Wheel spins and selects a sector
 - Wheel passes information to question board when category selected
 - Wheel updates score for proper sectors and for non-implement sectors just updates spin counter
 - Question board displays correct question
 - Question board displays correct answer
 - Spin counter is updated

Minimal

Features

- Wheel
 - Round indicator
 - Free turn sector gives player a free turn option
- Question Board
 - Timer displaying amount of time left to answer the question
 - Round indicator (round 1 or 2)
 - “Spins” indicator to inform the user how many spins are left in the round
 - Scoreboard that indicates the scores of the current player
- Question Editor
 - The question editor provides a convenient way for someone to edit the set of questions and answers, as well as the categories.
 - Consists of:
 - Category field
 - Question field for all 5 questions per category
 - Answer field for all 5 answers per question
 - Save button/option
 - Exit button/option
- Answer Board
 - The answer board is a variation of the question board that has many of the same features in addition to the user’s ability to submit an answer
 - Consists of:
 - Display of categories and associated questions
 - A highlighted box or indicator around question that was given
 - Show answer button/option
 - “Right” and “wrong” buttons/options that allow the opposing players to determine if answer is correct
 - Round indicator (round 1 or 2)
 - “Spins” indicator to inform the user how many spins are left in the round
 - Scoreboard that indicates the scores of the current player
- Scoreboard

- The scoreboard will be displayed on the question and answer board. The score will be updated as answers are determined

Interactions

- Wheel
 - Updates scoreboard when appropriate
 - Updates spin count
 - Allows for selection of question when appropriate
 - Updates player with free turn when appropriate
- Question Board
 - Once question is selected and presented, then the timer begins for the player to answer the question
- Answer Board
 - Opposing players select the “show answer” button/option to reveal the correct answer after player orally gives answer
 - An answer that is determined to be correct will add the point value for that question to the player’s score
 - An answer that is determined to be wrong will subtract the point value for that question to the player’s score
 - Scores for correct answers in round 2 are double the point value
- Scoreboard
 - Score will not be altered in the case of the timer running out before an answer is submitted
- Question Editor
 - As categories, questions, and answers are edited and saved via the editor, the question board will update accordingly
- Milestones
 - All sectors of the wheel are implemented
 - Both rounds are implemented and score properly
 - Question timer works
 - Question and answer scoring works
 - Question editor adds categories, questions, and answers

Some of the requirements that our product needs to comply with and implement are mentioned in the features of the skeletal system increment. The remaining requirements listed below complete the minimal requirements needed to deem our product complete and successful.

- When the player spins a sector that indicates a category, he or she must answer the next question in that category.
- The questions are answered in the order of increasing point value.
- Correct answers are awarded the corresponding points and an additional spin
- Incorrect answers result in the corresponding points being subtracted from the player’s score, and the player loses his or her turn (Negative Scores are possible)
- If all of the questions in the selected category have been answered, the player must spin again.
- Player will lose their turn if “lose turn” sector is landed on by the wheel

- Player will get a token for a free turn to be used accordingly if wheel lands on “free turn” sector. The number of free turn tokens is unlimited
- Player will lose all of his/her points for the current round if the wheel lands on “bankrupt” for their spin. The player loses his turn and can’t use a token for a second chance.
- Player will be allowed to choose which category to answer if the wheel lands on “player’s choice” sector
- Player’s opponent will be allowed to select the category to answer if the wheel lands on “opponents choice” sector
- Player score is doubled if wheel lands on the “Double your Score” sector
- In the second round of the game, the point values are doubled from the first round
- The double your score and bankruptcy options only apply to the round score not the total score
- The players keep their first round score regardless of what happens in the second round
- In each round there is a maximum of 50 spins of the wheel
- A round is over if either all of the questions have been answered or if the spin count goes to zero
- If the time limit on answering a question expires no points are subtracted from the player’s score but the player does lose their turn
- When a player spins a category sector, the program should display the question (or answer) on the screen
- The player will answer the query orally to the other players. Then the players should select a “Show answer” option. This will display the correct answer.

Target

The target system increment will describe the final deliverable that will be presented at the conclusion of the class. The target system will comply with and implement all of the previously mentioned requirements, final subsystems mentioned below, and provide a user interface that can be utilized by players to clearly follow and enjoy the game with ease.

Features

- Start Menu
 - The start menu will display the introduction screen to begin the game or utilize additional options
 - Consists of:
 - Start Button
 - Rules button/option
 - Exit button/option
- Score board
 - Winner Announcement

Interactions

- Start Menu
 - Start option is selected and the game starts
 - Rules option is selected then “Rules” display screen appears
 - Exit option is selected then program closes

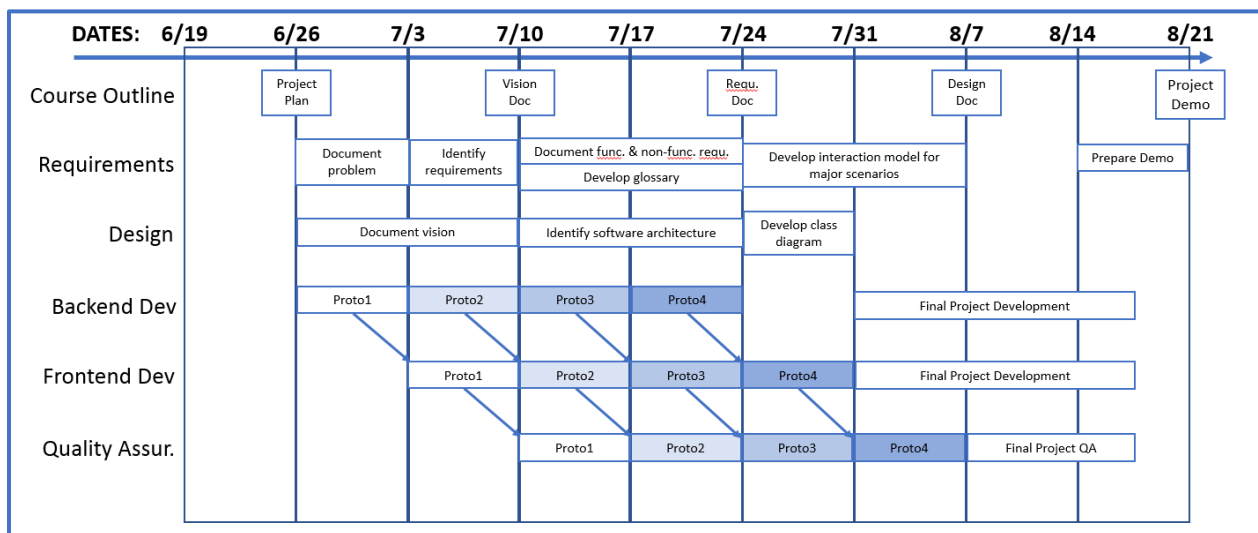
Dream

The dream system increment is what the development team thinks would be possible with more time and resources to develop the product. Ideally, the team would like to add enhancements that go beyond the minimum requirements for the product.

Some features that would be convenient for the player to have include the option of more than two players, and for a gui-based question editor that makes customizing questions easier than the test based option. Additional features such as a double jeopardy option, more animation the question board and wheel, the theme music that accompanies the timer when waiting for an answer to a question would be excellent additions that the team would hope to implement over time.

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, free turns, missed 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

Risk Assessment Steps

1. Identify risks
2. Qualify risk
 - a. Assess impact if risk occurs
 - b. Assess probability of risk occurrence
 - c. Identify risk priority
3. Manage risk
 - a. Reduce probability of risk occurring
 - b. Reduce impact of risk occurrence

Identify Risks

To identify risk, review the requirements and think of any issues that might keep from completing the requirement. Examples of risks to identify are:

- Review the deadline and schedule for any timelines that may be difficult to complete on time
- Identify tasks where the team may lack expertise
- Identify tasks that has many dependencies to work properly
- Identify tasks that requires a high amount of resources
- Consider impact if a team member is unable to continue working for a short period or permanently

Qualify Risks

Each risk is qualified on two factors: impact and probability. Each factor is rated on three levels: low, medium, and high.

Impact is how much an occurrence of the risk would affect the project. Probability is the likelihood the risk will occur.

Qualifying the risks can be done using historical data, expertise, and gut feeling. Gut feeling is generally considered the least desirable method though the most available especially to new and inexperienced teams.

Priority of the risk is based on the combination of the impact and probability. The priority is used to decide the amount of resources that should be used to manage a risk. It is also used to decide the risk to handle first if there are multiple risk occurrences.

		Impact		
Probability		High	Medium	Low
	High	High	High	Medium
	Medium	High	Medium	Low
	Low	Medium	Low	Low

Manage Risks

We can manage a risk's probability and impact to help mitigate the effects of the risk. It is important to understand the risk is never zero, but the overall impact may be negligible. It may be acceptable to simply accept the risk for low priority risks, as the mitigation may not be worth the effort.

Ways of reducing risk include reallocating resources, planning, redefining requirements, conservative estimates, alternative processes, and communication.

Quality Assurance

The requirements for software quality for a project can vary based on the expectations of the users and stakeholders along with any regulation or legal requirements.

For this project software quality is defined as:

1. Software that works on the software and hardware listed in the Resources section
2. Installs according to instructions
3. Plays a full game properly according to game rules
4. GUI based program
5. Completed by deadline

Software Quality Definition Table

Software Quality	Definition	Details and Updates
Hardware and Software Resources	Program works on stated hardware and software.	Hardware requirements are based on team's hardware. Software, other than OS requirements, can be change to meet quality needs. Such updates need to be reflected in Resource section and Installation documentation.

Installs	Program installs and opens on stated hardware using stated software	Ensure Installation documentation is updated when fixing related issues.
Game Play	Program plays according to game rules	Requirements from Wheel of Jeopardy document are final. Others used to clarify any ambiguousness can be updated to meet quality needs. Such updates need to be reflected in Architecture, Design Document and Game Play Instructions.
GUI	Layout and visuals	Update any changes to Architecture, Design Document, or Game Play Instructions if needed.
Deadline	Final program, documentation, and presentation due by August 11 th , 2018.	All requirements and features implemented must meet the final deadline date.

Identifying Quality Issues

Quality issues can be found during any phase and should be reported when found, not just during testing. Small issues and bugs found during programming that can be fixed immediately, without changing requirements of the project, can be implemented without reporting the issue. This is considered part of the programming process. This type of issue should be documented if the programmer can not immediately address it.

During testing, known issues are provided to the tester. Testers do not need to report known issues unless they have an update or correction. All other issues should be reported.

Reporting Quality Issues

Reporting of issues will be done through the Blackboard Group Tasks. Each issue will be given a priority level and a description that includes which quality definition it affects.

High Priority Issues. Issues that stop the game from completing. This includes unable to install the program, crashes, or logic issues that keep the game from completing.

Normal Priority Issues. Issues that have incorrect logic, incorrect game rules, or do not meet the requirements but the game can still be completed. This includes wrong number of spins, scoring is incorrect, or free spin token does not work.

Low Priority Issues. Issues that affect overall user experience, but does not affect any of the requirements. This includes awkward layout, confusing instructions, and misspellings.

Description

The description will include the following

1. Brief description of issue
2. Conditions under which the issue happened

3. Related software quality definition

Updates to issues will be added after the description and the last update. Updates include the following

1. Date of update
2. Initials of updater
3. Description of update

Fixing Issues

Issues fixes will be implemented using the Configuration Management Plan. The fix needs to not only meet the stated quality definition, but all quality definitions. For example, a fix will only work if it can be done before the deadline. Additionally, related documented will be updated to reflect any implemented fixes. See Software Quality Definition table above.

An issue may require a change in the requirements or software logic. Such changes will be requested through the Blackboard Discussion Board or at the weekly update meeting and approved by the team.

Documenting Fixes

All documentation related to a fix or known issue will be updated accordingly. The Quality Assurance Definition Table can be used to help ensure appropriate documentation is updated.

For each version of the software released, including beta releases for tester, a list of fixes and known issues for that version will be given.

Testing

The testing process is an inspection that is conducted in order to provide the end users/target audience with information about the standard of the software product. The ultimate goal of testing is to determine that the software product is fit for use. Testing for this software product will consist of executing the program in an effort to:

- ensure the program follows the game rules and complies with all stated requirements
- verify that subsystems function independently and in conjunction with other subsystems
- detect, report, and mitigate any errors/bugs that may hinder the programs performance

Considering the team is using a rapid prototype development approach to the software development, it is expected that programming and testing will be done concurrently. Testing will be conducted by each developer as a portion of the code is completed before its shared with the rest of the team. In addition to the high level goals of testing mentioned above, the testing will specifically verify that the subsystems respond correctly to various inputs including general and extreme case situations. Testing will also determine if the program performs all functions in a timely manner.

Bugs and issues are reported according the Quality Assurance Plan.

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.