DROP TABLE TEAM NAMES

Clue-Less Project Plan

02.23.2021

—

Drop Table Team Names
Martin Nganga
Joshua Persechini
Larry Thrower
Charlie Hammond
Wes Ottey

Drop Table Team Names Project Plan

Table of Contents

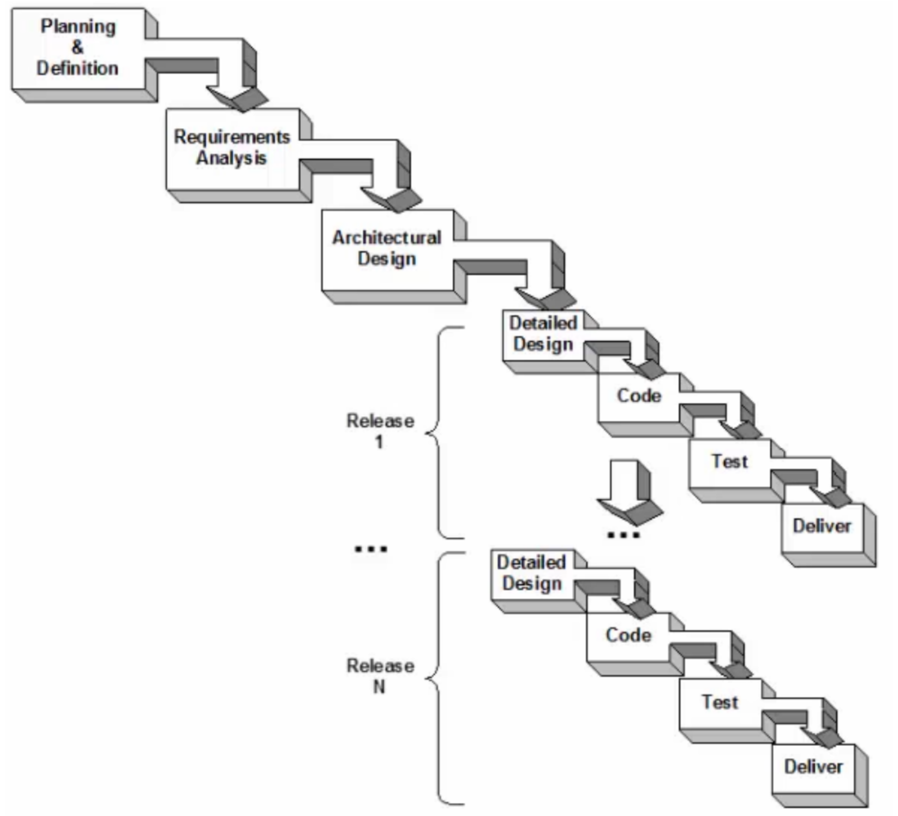
Scope

Project Objectives

Our team's objective is to create a simplified version of the game Clue which can be played online by multiple concurrent players. This project will entail initial planning and requirements gathering, followed by design, implementation, testing, and delivery of three incremental releases. It will also include ongoing risk analysis and management.

Life Cycle Description

Our project plan follows the Staged Delivery lifecycle model, with three scheduled incremental releases. These are described in more detail below, in the Work Products section. In this life cycle model, we'll create an overall requirements document and architectural design document that summarizes the total functionality required in the product, and then break that down into three increments: skeletal, minimal, and target. We will code, test, and demonstrate (deliver) each increment in sequence.



Work Products

Skeletal Increment

The skeletal increment will setup infrastructure and prove core system functionality. In this increment, we will set up the accounts required to host our application, bootstrap the application, develop simple messaging to show end-to-end functionality, and publish the application. At the end of this phase, we will have a web hosted application where we can pass simple messages between clients and the database.

Minimal Increment

The minimal increment will create the building blocks for the game application. At the end of the phase, our application will have core data structures, core game logic, and a basic user interface. To accomplish this, we will build the following:

- Data structures: we will develop data structures for game state and users. These core structures will live on the database and sent between the database and client using APIs.
- Core Game Logic: we will develop enough functionality for users to complete the game successfully.
- User Interface: we will develop a basic user interface where users can play the game. It will be something like a basic HTML web page with a grid for the game board and basic icons for pieces on the board.

Target Increment

The target increment will be the final product of the term. It will include a fully functioning application with a rich user interface. We will improve upon the minimal increment in the following ways:

- Rich User Interface: we will use more user friendly components to create a richer application for users to interact with.
- Game Lobby: we will develop a page to host the players before the game starts. This will make the user experience better because users can join at the same time.
- Expand Game Logic: we will expand the game logic to ensure it is complete to the assignment guidelines and works correctly.

Dream Increment

A dream increment would involve a fully functioning enterprise Clue game. This would include functionality for multiple games, persistent user profiles, complex graphics and sound. Additionally, user verification and authentication would be present to ensure a secure application. The target state will have more basic security and users could in theory call APIs directly or update the client state to manipulate the game in ways not present on the user interface.

Milestones

The table below summarizes significant milestones; see the Organizational Structure section for more details on the work involved in each milestone.

Milestone	**Projected Completion Date**
Team Charter Finalized	9-Feb-2021 (Completed)
Project Plan Finalized	23-Feb-2021
Vision Document Finalized	9-Mar-2021
Requirements Document Finalized	16-Mar-2021
Skeletal Increment of Product Delivered	23-Mar-2021
Design Document Finalized	6-Apr-2021
Minimal Increment of Product Delivered	13-Apr-2021
Target Increment of Product Delivered	11-May-2021

Organizational Structure

Team Structure

Our team structure is as follows. Tasks will be assigned by general group consensus during our weekly meetings, and will be based on completing needed tasks during each week. So, team members will work outside the responsibilities defined here depending on the primary goals for each week.

Role	**Team Member**	**Responsibilities**
Project Manager	Josh Persechini	- Coordinating and planning project component - Analyzing work needed to complete project deliverables
Lead Architect	Charlie Hammond	- Determine structure for different components of the project - Describe needed functionalities
Lead Programmer	Martin Nganga	- Plan and structure code to implement each functionality
Lead Tester/Quality Assurance Engineer	Larry Thrower	- Define testing process for each component of the project - Analyze risks present in product
Programmer	Wes Ottey	- Assist in coding aspects of the project

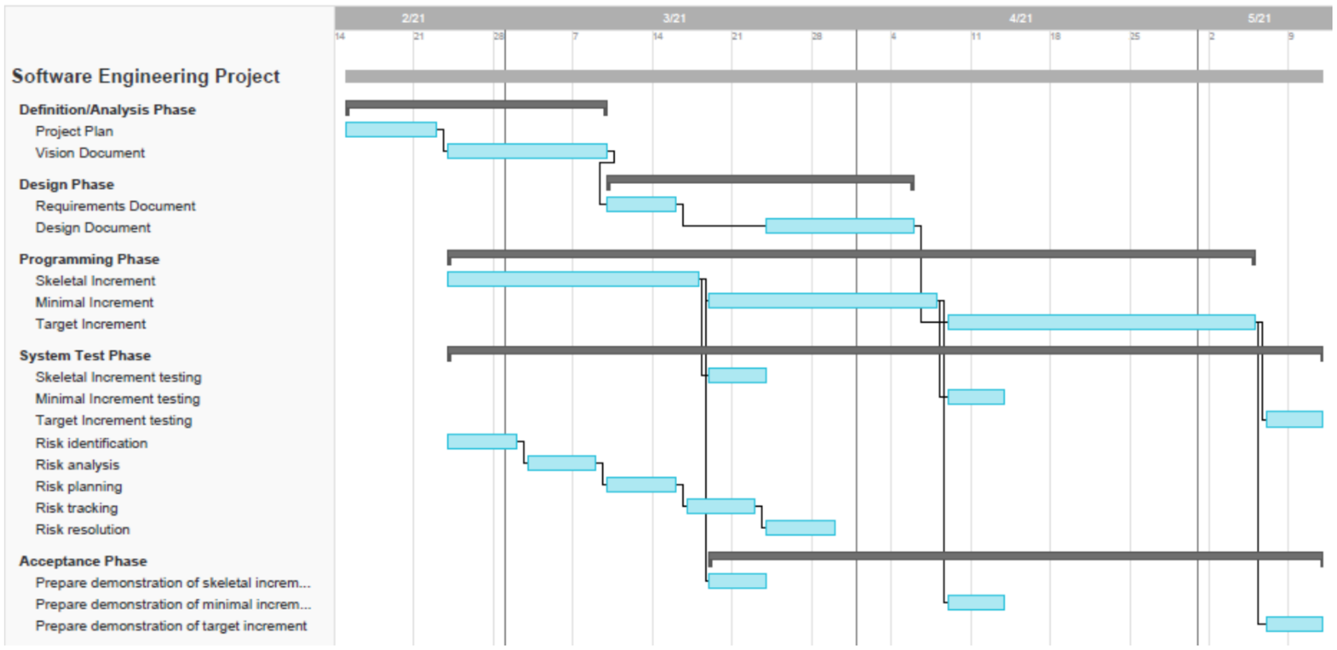
Work Breakdown Structure

The Work Breakdown Structure is a living document which will be updated as tasks are completed or assigned, new tasks or subtasks are identified, or due dates are modified. Please see the separate “WBS-DropTableTeamNames.xlsx” file for a detailed WBS.

Preliminary Schedule

Preliminary dates are given in the Work Breakdown Structure above. Below, the schedule and start dates for each phase and task is summarized in a Gantt chart. Tasks with the same start/end date in the WBS are consolidated here into a single task for clarity. Dependencies between tasks are indicated by a black line connecting those tasks. Dates are shown at the top, beginning in February 2021 and ending in May 2021. For a higher quality version of the Gantt Chart, see separate "Software_Engineering_Project.pdf" file.

Note that risk planning steps are shown here occurring in weekly increments; while these are the times we expect to be focusing on each component of risk mitigation, risk planning will also be an ongoing component of all project tasks.



Resources

Team Drop Table Team Names is planning to build a web based application for our project. The web application allows multiple users to simultaneously play Clue on multiple clients. Additionally, we will use a web socket to build a real time Clue application. There are several different components required to build a successful website. For our project, we will use the following:

- Technology Stack:
  - React - Javascript library that we will use for the front end.
  - Reactstrap - library to bootstrap components
  - Socket.io - library for a web socket. This enables a realtime application
  - Node.js - web server
  - MongoDB - database
- GitHub - version control for our software. GitHub will allow us to work on the project collaboratively and implement version control.
- Heroku - Cloud based PaaS hosting vendor for the web server. Heroku will allow us to quickly deploy the application.
- mLab - Database as a Service hosting vendor.
- Visual Studio Code - IDE

Not all members of the team have web development experience. We have found several resources to help teach us enough basics to successfully create a web based application for the project.

- Realtime application tutorial - https://www.freecodecamp.org/news/how-to-create-a-realtime-app-using-socket-io-react-node-mongodb-a10c4a1ab676/
- Heroku tutorial - https://www.freecodecamp.org/news/how-to-deploy-an-application-to-heroku/
- React tutorial - https://reactjs.org/tutorial/tutorial.html
- Reactstrap tutorial for building the initial application - https://reactstrap.github.io/
- mLab quickstart guide - https://docs.mlab.com/
- mLab and Heroku tutorial - https://medium.com/fbdevclagos/deploying-to-mlab-and-heroku-302b9ce9665e

Monitoring and Control Procedures

Project Tracking and Control

Defined steps, control points, and actions are necessary to monitor and control the project. These controls are used to keep associated tasks from deviating from the stated objectives and also provide a guide for getting the project back on track should it get off course. Strict adherence to the stated Functional Requirements in the assigned Project Document will be used to reduce the amount of deviation from the project baseline.

Performance will be measured using the deadlines assigned to tasks given to group members. Completion of these tasks within the allotted time will give an indication of whether the stated project incremental delivery dates can be met. If there is a lot of variance from the baseline, for instance, if it is expected that the project duration will exceed the planned duration by 20%, actions will be taken to meet the project targets. Such actions may include reassigning tasks to different group members, changing the software resources being used to accomplish a specific task or adding more members to the task causing project delays.

Integrated Change control during the programming phase will be implemented using Git as a source control tool. Changes in a project must be implemented in an integrated manner. Small changes in one aspect of the project might impact the overall project. Performing an integrated change control evaluates the changes and its impacts on the project. The lead programmer with the help of the Architect will be responsible for ensuring proper change implementation is planned to minimize the risk associated with changes.

Changes will be evaluated by the Group as a whole and if the Group rejects the change, it won't be implemented. If a change is approved, project plan revisions must be done and change should be implemented properly.

Software Quality Assurance and Testing

Quality assurance will be done using a combination of automated testing and manual testing. The React Testing Library will be used with Jest as the test runner. Jest is the environment where all the tests are actually executed.

Testing will be done by the group as a whole. The group will discuss the writing of test cases. Tests should test the functionality of the application by mimicking how it will be used by end users.

Testing will include Unit Tests, Integration tests and End to End Tests. Unit tests will be done to evaluate specific implementation details, Integration tests will be done to evaluate how well implemented features work well together. End to End testing will be done in different simulated browsers. The end to end test combines multiple unit and integration tests into one big test.

Configuration Management

The Project Architect with assistance from the rest of the Group will be in charge of Configuration Management. The configuration of our project will describe the meaningful and properly working combination of different modules or parts. The Architect will determine what specific software tools and other resources will be used and at what phases in the project these tools and resources will be used.

Requirements Management

Functional Requirements are provided in the Project Assignment Document. The requirements broadly cover:

a) Development of a Front End that is initially text based then eventually is developed as a Graphic User Interface. The Front End will allow multiple players to play an online version of the Game Clue in real time.
b) Real time messaging that notifies players when certain actions are taken in a game session for example when a player moves, makes a decision in the game etc..
c) Development of a Backend. The backend will store data used in the game such as player names, locations of rooms and hallways used in the game and any graphics used.

Detailed breakdowns of the aforementioned Functional Requirements are contained in the Work Breakdown Structure.

Project Communications

Project Communications will be done via the currently used Slack channel and the weekly Sunday Zoom meetings. Additional Channels may be created within the Slack Group for specific tasks to allow members to collaborate when working on a task. The weekly Zoom meetings will be used to brief the group as a whole on task progress and address any issues that come up.

Project Estimates

The Work Breakdown Structure and Preliminary Schedule sections above provide details on tasks. Estimates below are based on this preliminary schedule and broken down by weeks, defined as starting on Wednesday and ending on Tuesday (to align with course due dates).

High-Level Project Overview

This is a high-level overview showing the top-level tasks that will be worked on each week:

Week #	Week	Tasks Underway	Tasks Complete Before End of Week
1	24-Feb to 2-Mar	● **1.3 -** Vision Document ● **3.1 -** Skeletal Increment Development ● **4.2.1 -** Risk Identification	● **4.2.1 -** Risk Identification
2	3-Mar to 9-Mar	● **1.3 -** Vision Document ● **3.1 -** Skeletal Increment Development ● **4.2.2 -** Risk Analysis	● **1.3 -** Vision Document ● **4.2.2 -** Risk Analysis
3	10-Mar to 16-Mar	● **2.1 -** Requirements Document ● **3.1 -** Skeletal Increment Development ● **4.2.3 -** Risk Planning	● **2.1 -** Requirements Document ● **4.2.3 -** Risk Planning
4	17-Mar to 23-Mar	● **3.1 -** Skeletal Increment Development ● **4.1.1, 4.1.2, 4.1.3 -** Skeletal Increment Testing ● **5.1.1 -** Skeletal Increment Demo Prep ● **3.2 -** Minimal Increment Development ● **4.2.4 -** Risk Tracking	● **3.1 -** Skeletal Increment Development ● **4.1.1, 4.1.2, 4.1.3 -** Skeletal Increment Testing ● **5.1.1 -** Skeletal Increment Demo Prep ● **4.2.4 -** Risk Tracking
5	24-Mar to 30-Mar	● **2.2 -** Design Document ● **3.2 -** Minimal Increment Development ● **4.2.5 -** Risk Resolution	● **4.2.5 -** Risk Resolution
6	31-Mar to 6-Apr	● **2.2 -** Design Document ● **3.2 -** Minimal Increment Development	● **2.2 -** Design Document
7	7-Apr to 13-Apr	● **3.2 -** Minimal Increment Development	● **3.2 -** Minimal Increment Development

		● **4.1.4, 4.1.5, 4.1.6 -** Minimal Increment Testing ● **5.1.2 -** Minimal Increment Demo Prep ● **3.3 -** Target Increment Development	● **4.1.4, 4.1.5, 4.1.6 -** Minimal Increment Testing ● **5.1.2 -** Minimal Increment Demo Prep
8	14-Apr to 20-Apr	● **3.3 -** Target Increment Development	
9	21-Apr to 27-Apr	● **3.3 -** Target Increment Development	
10	28-Apr to 4-May	● **3.3 -** Target Increment Development	
11	5-May to 11-May	● **3.3 -** Target Increment Development ● **4.1.7, 4.1.8, 4.1.9 -** Target Increment Testing ● **5.1.3 -** Target Increment Demo Prep	● **3.3 -** Target Increment Development ● **4.1.7, 4.1.8, 4.1.9 -** Target Increment Testing ● **5.1.3 -** Target Increment Demo Prep

Below, the required effort for the project is broken down by specific task/week and summarized in terms of total hours of work required per week:

Week-by-Week Effort Breakdown

Week #1

Task #	Task Description	Hour Estimate
1.3.1	Document Stakeholder Vision	2
1.3.2	Document Problem	2
3.1.1	Create GitHub repository, Heroku account	2
3.1.2	Create mLab database	5
4.2.1	Risk identification	3
	Total Hours	**14**

Week #2

Task #	Task Description	Hour Estimate
1.3.3	Identify Stakeholder Needs	2
1.3.4	Define Stakeholder Quality and Value	2
3.1.4	Write code to ensure messaging with Socket.io	5
3.1.3	Create application with create-react-app	10
3.1.5	Add Socket.io, MongoDB, Node.js libraries to application	2
4.2.2	Risk analysis	3
	Total Hours	**24**

Week #3

Task #	Task Description	Hour Estimate
2.1.1	Glossary	1
2.1.2	Software Architecture	1
2.1.3	Information Domain	1
2.1.4	Functional Domain	2
2.1.5	Interfaces	2
2.1.6	Non-functional Requirements	2
2.1.7	Implementation Constraints	2
3.1.6	Add mLab URI to server code	2
4.2.3	Risk planning	3
	Total Hours	**16**

Week #4

Task #	Task Description	Hour Estimate
3.1.7	Push to GitHub and deploy to Heroku	2
4.1.1	Unit testing (skeletal increment)	3
4.1.2	Integration testing (skeletal increment)	3
4.1.3	End to end testing (skeletal increment)	3
5.1.1	Prepare demonstration of skeletal increment	5
3.2.1	Write UI code with core visuals (basic grid, simple HTML table)	15
4.2.4	Risk tracking	2
	Total Hours	**33**

Week #5

Task #	Task Description	Hour Estimate
2.2.1	Architecture	2
2.2.2	Static design	2
3.2.2	Implement player data structure for database	10
3.2.3	Develop client code to handle players and display info relevant to each user	10
4.2.5	Risk resolution	3
	Total Hours	**27**

Week #6

Task #	Task Description	Hour Estimate
2.2.3	Class specifications	2
2.2.4	Dynamic design	2
3.2.4	Implement turns and game state data structure for database	10
3.2.5	Develop a game start/reset on the client	5
	Total Hours	**19**

Week #7

Task #	Task Description	Hour Estimate
3.2.6	Implement core game logic in client and server code	15
4.1.4	Unit testing (minimal increment)	3
4.1.5	Integration testing (minimal increment)	3
4.1.6	End to end testing (minimal increment)	3
5.1.2	Prepare demonstration of minimal increment	5
	Total Hours	**29**

Week #8

Task #	Task Description	Hour Estimate
3.3.1	Develop "game lobby" page	5
	Total Hours	**5**

Week #9

Task #	Task Description	Hour Estimate
3.3.2	Polish UI	10
	Total Hours	**10**

Week #10

Task #	Task Description	Hour Estimate
3.3.3	Ensure all game logic implemented correctly	5
	Total Hours	**5**

Week #11

Task #	Task Description	Hour Estimate
3.3.4	Polish logic to ensure all details are complete	10
4.1.7	Unit testing (target increment)	3
4.1.8	Integration testing (target increment)	3
4.1.9	End to end testing (target increment)	3
5.1.3	Prepare demonstration of target increment	5
	Total Hours	**24**

Project Risks and Contingencies

Drop Table Team Names will routinely review and update the following five activities dynamically throughout the course: risk identification, risk analysis, risk planning, risk tracking and risk resolution. The team will create, update and maintain a table of the most probable sources of project risk, their current and previous priorities, severities and plans of action, as shown below. These risks include uncertainty, knowledge, project concerns and project issues.

With regard to uncertainty, the requirements and project deliverables are well documented. Estimates, however, are not based on valid historical information because this is the first project the team is developing. At the top of the list of knowledge risks for the team is framework knowledge. Not every team member is familiar with the framework that will be used. Learning time could be critical. To minimize this risk, the team decided early on which framework to utilize to allow maximum time for the team members who are unfamiliar with the framework to get up to speed.

Knowledge of the rules of the game is much lower on the list of risks because every team member has become familiar with the rules of the board game, including the classic version as well as the “Clue-less” version. The hard deadlines and fixed timeline of deliverables affect the delivery dates of each of the milestones, which must be prioritized. All of the requirements are clearly defined, minimizing the risk of ambiguous requirements. However, issues may arise regarding team collaboration. Because this is an online course, and teammates are spread across the globe in vastly different time zones, communication between team members could be hampered.

Risk analysis involves analyzing the impact of each risk and the associated loss. Because hard deadlines require delivery on a fixed timeline, schedule overruns should not occur. This could cause diminishment of the work product, and end product quality. The risk exposure and risk severity will be calculated for each deliverable. Risk planning will be utilized to create a plan of action that will reduce the probability of the risk occurring and/or reduce the impact if the risk should materialize. Frequent and regular communication using the tools listed in the Team Charter will be used to reduce the risk of missing a deadline. As discussed above, GitHub will allow us to work on the project collaboratively and implement version control. All team members will use Visual Studio integrated development environment to reduce the risk of complicating environmental variables and system requirements between different teammates.

Risk tracking will be used to monitor the status of known project risks, including time estimates, framework knowledge, delivery deadlines and collaboration. Risk resolution actions will be taken should a risk occur. These include avoiding the risk, risk reduction as discussed above, and accepting the risks, for example if team members drop the course or become too busy with other responsibilities.

Risk Tracking Table

Identified Risk	**Current Priority**	**Previous Priority**	**Action Plan Status**	**Risk Severity**
Framework knowledge	1	N/A	Not every team member is familiar with the framework that will be used. Learning time could be critical. To minimize this risk, the team decided early on which framework to utilize to allow maximum time for the team members who are unfamiliar with the framework to get up to speed.	1
Time estimates	2	N/A	Time estimates are not currently based on valid historical information because this is the first project the team is developing.	2
Team Collaboration	3	N/A	Because this is an online course, and teammates are spread across the globe in vastly different time zones, communication between team members could be hampered.	1
Requirements	4	N/A	Requirements are well documented.	4
Project deliverable uncertainty	5	N/A	Project deliverables are well documented.	4
Knowledge of rules of the game	6	N/A	Every team member has become familiar with the rules of the board game, including the classic version as well as the “Clue-less” version.	5