

Rock Chalk Rendezvous
Software Development Plan

Version 1.1

Rock Chalk Rendezvous	Version: 1.1
Software Development Plan	Date: 03/21/2024
proj_plan1	

Revision History

Date	Version	Description	Author
02/19/2024	0.1	First draft	Shayna Weinstein, Dylan Kneidel, Delroy Wright, Ben Phillips, Andrew Reyes
02/22/2024	1.0	Draft to turn in	Shayna Weinstein, Dylan Kneidel, Delroy Wright, Ben Phillips, Andrew Reyes
03/15/2024	1.1	Updated Project Plan with more accurate dates	Shayna Weinstein, Dylan Kneidel, Delroy Wright, Ben Phillipd, Andrew Reyes

Rock Chalk Rendezvous	Version: 1.1
Software Development Plan	Date: 03/21/2024
proj_plan1	

Table of Contents

- 1. Introduction.....4
 - 1.1 Purpose..... 4
 - 1.2 Scope.....4
 - 1.3 Definitions, Acronyms, and Abbreviations..... 4
 - 1.4 References..... 4
 - 1.5 Overview..... 5
- 2. Project Overview.....6
 - 2.1 Project Purpose, Scope, and Objectives..... 6
 - 2.2 Assumptions and Constraints..... 6
 - 2.3 Project Deliverables..... 6
 - 2.4 Evolution of the Software Development Plan..... 7
- 3. Project Organization.....8
 - 3.1 Organizational Structure..... 8
 - 3.2 External Interfaces..... 9
 - 3.3 Roles and Responsibilities.....9
- 4. Management Process..... 11
 - 4.1 Project Estimates..... 11
 - 4.2 Project Plan..... 11
 - 4.2.1 Phase Plan..... 11
 - 4.2.2 Iteration Objectives..... 11
 - 4.2.3 Releases..... 12
 - 4.2.4 Project Schedule..... 12
 - 4.2.5 Project Resourcing..... 13
 - 4.3 Project Monitoring and Control..... 13
 - 4.4 Requirements Management..... 15
 - 4.5 Quality Control..... 15
 - 4.6 Reporting and Measurement..... 16
 - 4.7 Risk Management..... 16
 - 4.8 Configuration Management..... 17
- 5. Annexes..... 18

Rock Chalk Rendezvous	Version: 1.1
Software Development Plan	Date: 03/21/2024
proj_plan1	

1. Introduction

1.1 Purpose

The purpose of the *Software Development Plan* is to gather all information necessary to control the project. It describes the approach to the development of the software and is the top-level plan generated and used by managers to direct the development effort.

The following people use the *Software Development Plan*:

- The **project manager** uses it to plan the project schedule and resource needs, and to track progress against the schedule.
- **Project team members** use it to understand what they need to do, when they need to do it, and what other activities they are dependent upon.
- Any **instructor** may use it to keep a running tab on what is currently being developed within this project and the relative dates for which each phase of the program will be released

1.2 Scope

This *Software Development Plan* describes the overall plan to be used by the “Rock Chalk Rendezvous” project, including deployment of the product.

- The details of the individual iterations will be described in the Iteration Plans.
- The plans as outlined in this document are based upon the product requirements as defined in the *Vision Document*.
- The maintenance and management of our project will be based upon the phase plan and project schedule as listed in Section 4.2.4 below.

1.3 Definitions, Acronyms, and Abbreviations

See the Project Glossary.

1.4 References

Rock Chalk Rendezvous	Version: 1.1
Software Development Plan	Date: 03/21/2024
proj_plan1	

[1] Project Vision, version 1.0 from 2/19/2024. Team Strawhacks, accessible via project GitHub documents folder.

[2] Glossary of Terms, version 0.1 from 2/19/2024. Team Strawhacks, accessible via project GitHub documents folder.

[3] Software Requirements Specifications, version 1.0 from 03/15/2024. Team Strawhacks, accessible via project GitHub documents folder.

1.5 Overview

This *Software Development Plan* contains the following information:

Project Overview — provides a description of the project's purpose, scope, and objectives. It also defines the deliverables that the project is expected to deliver.

Project Organization — describes the organizational structure of the project team.

Management Process — explains the estimated schedule, defines the major phases and milestones for the project, and describes how the project will be monitored.

Applicable Plans and Guidelines — provides an overview of the software development process, including methods, tools and techniques to be followed.

Annexes — includes an organized list of all relevant documents and materials pertaining to the reader of the **Software Development Plan** as well as any project technical standards and plans which apply to this project.

Rock Chalk Rendezvous	Version: 1.1
Software Development Plan	Date: 03/21/2024
proj_plan1	

2. Project Overview

2.1 Project Purpose, Scope, and Objectives

The objective of our project is to create a platform for scheduling meetings for a group by compositing schedule information from each member of the group. The purpose is to improve the functionality of existing schedule comparison software to reduce repetitive actions and mitigate the need for active responses from group members for each event.

We plan to deliver a desktop application for client users and a back-end application that will run continuously on a central dedicated server and process and respond to internet data received from the client desktop application.

2.2 Assumptions and Constraints

Assumptions:

- All team members will be able to access the development environment.
- All group members actively use and update the calendars that are accessed by this platform.
- Our platform can integrate with existing calendar applications like Google Calendar or Microsoft Outlook.
- A stable network connection will be available for our application.
- We will have access to a dedicated server to process and respond to internet data received.

Constraints:

- The Dedicated server must be continuously running to process data.
- The project must adhere to constraints defined by this Project's Scope, Schedule, and Budget.
- The platform must be able to handle multiple requests by multiple users.
- Real-time or near real-time data synchronization is necessary to ensure all users are viewing up-to-date schedules.
- The system must implement authentication and authorization mechanisms to verify user identities and manage access controls.

2.3 Project Deliverables

Rock Chalk Rendezvous	Version: 1.1
Software Development Plan	Date: 03/21/2024
proj_plan1	

Deliverables for each project phase are identified in the Development Case. Deliverables are delivered towards the end of the iteration, as specified in section 4.2.4 *Project Schedule*.

2.4 Evolution of the Software Development Plan

Proposed Version	Date
0.1	2/19/2024
1.0	2/22/2024
1.1	03/15/2024

Revision Criteria

- The *Software Development Plan* will be revised prior to the start of each Iteration phase.
- The *Software Development Plan* may be revised during each Iteration phase to better fit the scope of the project.
- The *Software Development Plan* will be revised after the end of each Iteration phase.

Rock Chalk Rendezvous	Version: 1.1
Software Development Plan	Date: 03/21/2024
proj_plan1	

3. Project Organization

3.1 Organizational Structure

The project team for this initiative follows a well-defined organizational structure designed to ensure efficient collaboration and effective decision-making. The team consists of various roles, each contributing to different aspects of the project. Here's an overview:

1. Project Manager: Shayna Weinstein

The project manager is at the helm, overseeing the entire project. They are responsible for planning, coordination, and ensuring that the project stays on track in terms of timelines and objectives.

2. Development Team: Everyone

This team comprises software developers, engineers, and programmers responsible for the actual creation and implementation of the software solution. They work closely with other teams to meet project requirements. Everyone is a part of our development team

3. Quality Assurance Team: Andrew Reyes

This team focuses on testing the software to identify and rectify any issues before the final release. Their role is critical in ensuring the product meets high-quality standards.

4. System Architects: Ben Phillips and Delroy Wright

System architects are responsible for designing the overall structure of the software. They make decisions on the system's architecture, ensuring scalability, security, and efficiency.

5. Review Authorities: Dylan Kneidel

In addition to the project manager, there are designated review authorities who provide oversight and guidance. This could include department heads, senior executives, or key stakeholders who review progress, make strategic decisions, and ensure alignment with organizational goals.

Communication Channels:

Rock Chalk Rendezvous	Version: 1.1
Software Development Plan	Date: 03/21/2024
proj_plan1	

Regular communication channels are established to facilitate information flow within the team. This includes team meetings, progress reports, and documentation shared through project management tools moderated through Discord and Github. Regular review and feedback mechanisms are in place to adapt to changing circumstances and ensure the project's success.

3.2 External Interfaces

[Describe how the project interfaces with external groups. For each external group, identify the internal and external contact names. This should include responsibilities related to deployment and acceptance of the product.]

3.3 Roles and Responsibilities

Person	Unified Process for EDUcation Role
Shayna Weinstein shayna.weinstein@ku.edu	Team Administrator
Dylan Kneidel dckneidel@ku.edu	Documentation Engineer
Delroy Wright dcwright@ku.edu	Configuration Management Co-Lead Technical Lead

Rock Chalk Rendezvous	Version: 1.1
Software Development Plan	Date: 03/21/2024
proj_plan1	

Ben Phillips ben.a.phillips@ku.edu	Configuration Management Co-Lead
Andrew Reyes andrewreyes@ku.edu	Quality Assurance Engineer

Rock Chalk Rendezvous	Version: 1.1
Software Development Plan	Date: 03/21/2024
proj_plan1	

4. Management Process

4.1 Project Estimates

[Provide the estimated cost and schedule for the project, as well as the basis for those estimates, and the points and circumstances in the project when re-estimation will occur.]

4.2 Project Plan

4.2.1 Phase Plan

[Include the following:

- a Gantt chart showing the allocation of time to the project phases (Not necessarily detailed to the activity level; this type of Gantt Chart is providing along with the Iteration Plans themselves; Provide an Overview of the project Timeline with the major miles stones]
- identify **major milestones** with their achievement criteria

Define any important release points and demos.]

4.2.2 Iteration Objectives

1. Planning
 - Write the initial version of project planning documents.
2. Initial Requirements Specification
 - Identify and develop the requirements for the software.
3. First Iteration of Implementation
 - Establish communication protocols
 - Design Data Structures
 - Implement client and server prototypes with core functions
4. Second Iteration of Implementation
 - Implement all required functionality for the MVP server application
 - Develop enough MVP client-side features to properly interface with the completed server
 - UA system
 - Test first iteration implementations
5. User Interface

Rock Chalk Rendezvous	Version: 1.1
Software Development Plan	Date: 03/21/2024
proj_plan1	

- Design and implement the final graphical user interface
-
- 6. Optional Features
 - Features we have not yet implemented as described in SRS Document Section 3 [3]
 - Test user interface and optional features
- 7. Write User Manual
 - Write the complete user manual for the software
 - Begin reviewing all project documents for final revisions
 - Test new optional features
- 8. Final Project Implementation
 - Finalize all project documents
 - Prepare the completed deliverable package

4.2.3 Releases

- First implementation prototype
 - A very basic form of client and server that can communicate with each other and transmit and process meaningful calendar data. This release may use languages other than C++ in order to more dynamically prototype core structures and systems.
- Minimum viable implementation
 - This release will contain all of the features described as “essential” by SRS^[3] section 3. This release will be implemented in C++.
- User interface release
 - This release will include a mature graphical user interface for the client application that fully interacts with the systems established in the minimum viable implementation release.
- Optional feature release
 - This release will add some or all of the features described as “optional” by SRS^[3] section 3 according to what the team agrees is achievable in the time allotted to this iteration phase.
- Final release with user manual
 - This release will include only corrective modifications to the software, and will include final versions of all project documents including a completed user manual.

4.2.4 Project Schedule

Rock Chalk Rendezvous	Version: 1.1
Software Development Plan	Date: 03/21/2024
proj_plan1	

<u>Iteration/phase</u>	<u>Target Date</u>
Planning	02/18/2024 - 02/25/2024
Initial Requirements Specification	02/26/2024 - 03/16/2024
First Iteration of Implementation	03/17/2024 - 03/24/2024
Second Iteration of Implementation	03/25/2024 - 04/07/2024
User Interface	04/08/2024 - 04/21/2024
Optional Features	04/23/2024 - 05/01/2024
Write User Manual	04/23/2024 - 04/25/2024
Final Project Implementation	04/30/2024 - 05/01/2024

4.2.5 Project Resourcing

[Identify the numbers and type of staff required here, including any special skills or experience, scheduled by project phase or iteration.]

List any special training project team members will require, with target dates for when this training should be completed.]

4.3 Project Monitoring and Control

· Quality Control:

Quality will be monitored by using test cases to verify correctness during the development period. During the planning phase, the Quality Assurance engineer will be scrupulous and will verify that the plan is feasible.

Any inadequacies will be addressed on the spot and a GitHub ticket will be created by the Quality Assurance Engineer. These inadequacies will be addressed by developers and then checked by the quality assurance engineer. After corrective action is taken, the ticket will be resolved, and the issue will be removed from the ticket list.

Rock Chalk Rendezvous	Version: 1.1
Software Development Plan	Date: 03/21/2024
proj_plan1	

All artifacts will go through a review process (by the team) before being submitted for release. The review process will include the aforementioned test cases, where all team members will test the system using 1-10 personal tests.

- **Reporting and Measurement:** Describe reports to be generated. Specify which metrics should be collected and why. **OR** if available, refer to the **Project Measurements and Project Measurements** document

- **Risk Management:**

Risks will be identified through a review process carried out by team members. If a risk is identified, a team member will notify the Quality Assurance Engineer. It will then be the quality assurance engineer's duty to carry out the fix. The review process will occur once per release (project deliverable) unless further review is necessary.

Initial identified risk cases:

Risk: A team member becomes sick

Mitigation: If this happens, the team will periodically check in with the ill member. The Team Administrator will work with the ill member to reduce their workload if necessary.

Risk: A team member drops the class

Mitigation: If this happens, the Team Administrator will evenly distribute tasks that were originally delegated to that member.

Risk: A team member is unable to finish assigned tasks before a deadline

Mitigation: If this happens, the Team Administrator will work with the team member to come to some solution. If there are repeated offenses, the Team Administrator will reach out to their lab TA for support in the matter.

- **Configuration Management:**

Configurations and artifacts will be managed and stored using git and Github.com. Bug fix tickets will also be managed via git and Github.com. Whenever a ticket is created, it will be administered and monitored using this service. The Quality Assurance Engineer will work in this

Rock Chalk Rendezvous	Version: 1.1
Software Development Plan	Date: 03/21/2024
proj_plan1	

domain. Program configurations will be updated whenever a significant change is made to the source code. When updated, program comments and documentation will also be updated. All changes will be verified and permitted by the Quality Assurance Engineer. All additions to artifacts and executables will be included in every release.

4.4 Requirements Management

The requirements for this project are described in the Software Requirements Specification document (soft_req_spec1). Changes to the requirements as described in this document are to occur as discussed in group meetings and subjected to review by the team before and after the change has been made.

4.5 Quality Control

1. Code Reviews and Testing:

- Regular code reviews will be conducted to ensure adherence to coding standards, best practices, and to identify potential issues early in the development process.
- Comprehensive testing will be performed, including unit tests, integration tests, and end-to-end tests, to verify the functionality, reliability, and performance of the application.

2. **Bug Tracking and Resolution:**

- A bug tracking system will be implemented to log and prioritize reported issues.
- Timely resolution of bugs will be ensured through a structured process involving triaging, assigning, fixing, and verifying fixes.

3. Performance Monitoring and Optimization:

- Continuous monitoring of application performance will be conducted to identify bottlenecks and areas for optimization.
- Performance optimization efforts will be prioritized to maintain responsiveness and scalability, especially as the user base grows.

4. Release Management

Rock Chalk Rendezvous	Version: 1.1
Software Development Plan	Date: 03/21/2024
proj_plan1	

- Each release will undergo thorough testing and validation before deployment to ensure stability and reliability.
- Rollback procedures will be established in case of unforeseen issues post-deployment, with a focus on minimizing downtime and user impact.

4.6 Reporting and Measurement

Updated schedule estimates, and metrics summary reports, will be generated at the end of each iteration.

The Minimal Set of Metrics, as described in the RUP Guidelines: Metrics will be gathered on a weekly basis. These include:

Earned value for completed tasks. This is used to re-estimate the schedule and budget for the remainder of the project, and/or to identify need for scope changes.

Total defects open and closed – shown as a trend graph. This is used to help estimate the effort remaining to correct defects.

Acceptance test cases passing – shown as a trend graph. This is used to demonstrate progress to stakeholders.

4.7 Risk Management

Risks will be identified in Inception Phase using the steps identified in the RUP for Small Projects activity “Identify and Assess Risks”. Project risk is evaluated at least once per iteration and documented in this table.

Risks:

1. Integration Difficulty - This risks describes any difficulties encountered in building and integrating our project, leading to inaccurate or incomplete versions.
2. Data Synchronization Issues - This risk describes issues encountered in synchronizing data across multiple schedules.

Rock Chalk Rendezvous	Version: 1.1
Software Development Plan	Date: 03/21/2024
proj_plan1	

3. Team Collaboration Breakdown - This risk describes a breakdown in the team's communication and teamwork, leading to inadequate progress.
4. Unexpected Technical Issues - This risk describes unexpected issues in the technology required for our project, this could be server malfunction, lack of internet connection, etc.
5. Poor Tracking and Documentation - This risk describes a lack of adequate documentation and tracking for our project, leading to ambiguity and confusion.
6. Scope and Feature Creep - This risk describes expanding the project scope to include more features than initially planned, potentially causing delays and budget overruns.
7. Poor User Interface - This risk describes a malfunctioning, or incomplete user interface that does not provide users with the necessary functionality.

4.8 Configuration Management

Appropriate tools will be selected which provide a database of Change Requests and a controlled versioned repository of project artifacts.

All source code, test scripts, and data files are included in baselines. Documentation related to the source code is also included in the baseline, such as design documentation. All customer deliverable artifacts are included in the final baseline of the iteration, including executables.

The Change Requests are reviewed and approved by one member of the project, the Change Control Manager role.

Tools:

- Github Repository - Keeps database of changes, versions, and project artifacts.
 - Contains source code, test scripts, and data files.
 - Contains documentation related to source code and design documentation.
 - Contains all customer deliverable artifacts.
- Google Drive
 - Contains working documents like the Project Plan and Project Description that define our Project and its scope.

Rock Chalk Rendezvous	Version: 1.1
Software Development Plan	Date: 03/21/2024
proj_plan1	

5. Annexes

The project will follow the UPEDU process.

Other applicable process plans are listed in the references section, including Programming Guidelines.