Development Plan
The Crazy Eights

Ruida Chen
Alvin Qian
Ammar, Sharbat
Jiaming Li

September 2025

Table 1: Revision History

Date	**Developer(s)**	**Change**
Date1	Name(s)	Description of changes
Date2	Name(s)	Description of changes
...	...	...

[Put your introductory blurb here. Often the blurb is a brief roadmap of what is contained in the report. —SS]

[Additional information on the development plan can be found in the lecture slides. —SS]

1 Confidential Information?

[State whether your project has confidential information from industry, or not. If there is confidential information, point to the agreement you have in place. —SS]

[For most teams this section will just state that there is no confidential information to protect. —SS]

2 IP to Protect

[State whether there is IP to protect. If there is, point to the agreement. All students who are working on a project that requires an IP agreement are also required to sign the "Intellectual Property Guide Acknowledgement." —SS]

3 Copyright License

[What copyright license is your team adopting. Point to the license in your repo. —SS]

4 Team Meeting Plan

[How often will you meet? where? —SS]

[If the meeting is a physical location (not virtual), out of an abundance of caution for safety reasons you shouldn't put the location online —SS]

[How often will you meet with your industry advisor? when? where? —SS]

[Will meetings be virtual? At least some meetings should likely be in-person. —SS]

[How will the meetings be structured? There should be a chair for all meetings. There should be an agenda for all meetings. —SS]

5 Team Communication Plan

[Issues on GitHub should be part of your communication plan. —SS]

6 Team Member Roles

[You should identify the types of roles you anticipate, like notetaker, leader, meeting chair, reviewer. Assigning specific people to those roles is not necessary at this stage. In a student team the role of the individuals will likely change throughout the year. —SS]

7 Workflow Plan

- How will you be using git, including branches, pull request, etc.?
- How will you be managing issues, including template issues, issue classification, etc.?
- Use of CI/CD

8 Project Decomposition and Scheduling

- Our team will use Github Projects as our core tool for task tracking and project management. Each feature will be decomposed into specific, verifiable small tasks, these tasks will be created in the form of Issues in Github Kanban Board. The Kanban board is divided into 4 stages: Todo, In Progress, Review and Done to visualize the workflow and ensure accountability. Team members will be assigned to different tasks, related pull request will be tied to these issues to maintain tracibility.
- Link to our Kanban project: https://github.com/orgs/The-Crazy-Four-Games/projects
- Decomposed Schedule:
  - Week 03: Team Formed, Project Selected
    - Create Github repo
    - Assign intial role
    - Decide on final project selection
  - Week 04: Problem Statement, POC, Development Plan
    - Draft initial problem statement and development plan
    - Set up CI/CD pipeline and Github branch strategy
    - Discuss the final programming languages and frameworks
  - Week 06: SRS + Hazard Analysis (Rev. 0)
    - Draft System Requirements Specification (SRS)
    - Identify hazards and mitigation strategy
  - Week 08: V&V Plan (Rev. 0)
    - Define V&V strategy

    - Break down into different tests and assigned responsibilities
  - Week 10: Design Document (Rev.-1)
    - Decompose system into major modules (frontend, backend, db, api, etc.)
    - Document architecture diagrams
  - Week 11-12: Proof of Concept Demonstration
    - Implement minimum working prototype
    - Prepare slides and live demo for demo presentation
  - Week 16: Design Document (Rev.0)
    - Refine design based on the feedback of POC Demo
    - Add details for extensibility and scalability
  - Week 18-19: Revision 0 Demonstration
    - Implement key features and workflows
    - Conduct internal testing and bug fixing
    - Prepare live demo
  - Week 22: V&V Report Revision 0
    - Execute test plan and record results
    - Analyze test coverage and tracibility
  - Week 24: Final Demonstration (Rev.1)
    - Finalize all core features
    - Optimize performance
    - Conduct mock presentation and user feedbacks
  - Week 26: EXPO Demonstration
    - Prepare polished project presentation for EXPO
  - Week 26: Final Documentation (Rev.1)
    - Finalize all documents

9 Proof of Concept Demonstration Plan

Main Risk

- Implementation Complexity: Correctly enforcing the gameplay mechanism (e.g. turn-taking, rule variation, base-12)
- Ensuring required libraries and frameworks install and integrate smoothly across team members' environments.

PoC Demonstration Goals
In our PoC demonstration, we will address these risks by showing:

- A working prototype of Crazy Eights where players can take turns, match cards by suit/rank, and play an "8" as a special
- Successful integration between frontend and backend for real-time gameplay.

10 Expected Technology

- JavaScript, TypeScript
- PostgreSQL
- Github, git, Github projects
- Node.js, React
- Jest

11 Coding Standard

- Quality-Oriented Development: Code should be clear, maintainable and consistent across the whole project
- Requirement and Specification-Based: All implementation will be tied to the requirements and specifications
- Defensive Programming: Follow practices that reduce errors and improve robustness

Appendix — Reflection

[Not required for CAS 741 —SS]

1. Why is it important to create a development plan prior to starting the project?
   A clear development plan provides the team with structure and direction, it ensures that all team members have a common understanding of the project's goals, responsibilities and timelines. It can also reduce uncertainty, help the team identify risks early, and make it easier to track progress.

2. In your opinion, what are the advantages and disadvantages of using CI/CD?
   Advantages: CI/CD automates testing and deployment, it reduces human errors and improves code quality. Since it encourages frequent integration, problems are detected early.
   Disadvantages: Setting up CI/CD pipelines can be time-consuming, especially for small teams, it may introduce overhead if the project scope is small or team members are not familiar with this tool.

3. What disagreements did your group have in this deliverable, if any, and how did you resolve them?
   One disagreement our group had was whether to focus solely on developing the Crazy Eights card game, or to make it into a product line that could support multiple card games/number systems, some members felt that the product line idea would make the project impressive and ambitious, while others were concerned about the limited timeline and feasibility. After a team discussion and seeking advice from the professor and supervisor, we resolved this by agreeing to prioritize Crazy Eights as the core deliverable, ensuring we can deliver a complete and functional game. At the same time, we left the product line concept as a strech goal that could be pursued if time and resources permit. This compromise allowed us to balance ambition with practicality, while keeping the team aligned.

Appendix — Team Charter

[borrows from University of Portland Team Charter —SS]

External Goals

- Deliver a polished product at the EXPO with hope of receiving positive feedback
- Build a project that can be showcased in future interviews and portfolios
- Aim for a strong course grade by following best practices and meeting all deliverable expectations
- Strengthen our knowlegde of modern frameworks so that the project also contributes to our long-term career growth.

Attendance

Expectations

Team members are expected to attend scheduled meetings(in-person or via Discord) on time and stay until the meeting is concluded. Consistent attendance is essential to maintain good communication and progress.

Acceptable Excuse

Acceptable excuses for missing a meeting or a deadline include illness, family emergencies, or unavoidable academic conflicts.Unacceptable excuses include forgetting, oversleeping, etc.

In Case of Emergency

If a team member experiences an emergency, they should notify the team as soon as possible through Discord. They should also provide updates on the status of their assigned tasks and, if neccessary, delegate or share their work so that the team can adjust and continue meeting deadlines.

Accountability and Teamwork

Quality

[What are your team's expectations regarding the quality of team members' preparation for team meetings and the quality of the deliverables that members bring to the team? —SS]

Attitude

[What are your team's expectations regarding team members' ideas, interactions with the team, cooperation, attitudes, and anything else regarding team member contributions? Do you want to introduce a code of conduct? Do you want a conflict resolution plan? Can adopt existing codes of conduct. —SS]

Stay on Track

[What methods will be used to keep the team on track? How will your team ensure that members contribute as expected to the team and that the team performs as expected? How will your team reward members who do well and manage members whose performance is below expectations? What are the consequences for someone not contributing their fair share? —SS]

[You may wish to use the project management metrics collected for the TA and instructor for this. —SS]

[You can set target metrics for attendance, commits, etc. What are the consequences if someone doesn't hit their targets? Do they need to bring the coffee to the next team meeting? Does the team need to make an appointment with their TA, or the instructor? Are there incentives for reaching targets early? —SS]

Team Building

[How will you build team cohesion (fun time, group rituals, etc.)? —SS]

Decision Making

[How will you make decisions in your group? Consensus? Vote? How will you handle disagreements? —SS]