

Bravo Planning Documents Review

Summary: A major issue observed was the lack of integration in the overall project timeline. While each sub-team created separate timelines, there was no overarching timeline accounting for how each team's work will be integrated together ensuring that their work aligned cohesively which makes it difficult to determine if all sub-teams are on track for meeting shared deadlines (or deadlines for task dependencies). Additionally, the team had poor naming conventions for their documents which created disorganization, they had "roles and responsibilities files" for each sub-team as well as timelines which created even more confusion. Some of those files also contained private information such as names which destroyed anonymity. The absence of a clear process of tracking each team's work may cause delays if one team's part is incomplete.

Overall, while some teams demonstrated strong planning, others lacked essential elements such as resource allocation, task dependencies, and contingency plans. A more cohesive project-wide timeline, improved integration tracking, and better-defined responsibilities would significantly enhance coordination. Moreover, defining accountability measures would help prevent delays and ensure a smoother workflow throughout the project.

Integration Planning Review:

Timeline Issues, Missed Milestones, or Unrealistic Deadlines:

- Demonstration videos mentioned in week 2 when the focus should have been on descriptions and diagrams for use cases and class structure diagrams. There is no planning phase to plan the classes, methods etc. needed.
 - Project is not even finished so why is the focus on the videos right now?
- Final number of use cases different from what is stated in the plan which creates inconsistency.
- There is no time allocated for integrating every sub-team's work together which is the most crucial (and possibly most challenging) aspect of integration.
- The timeline for other requirements of iteration 2 and 3 look doable and realistic.

Suggestions:

- Clarify what will be included in week 2 demo videos since they were mentioned or move them to a later stage. Should have focused on completing descriptions and diagrams first.
- Could change number of uses cases in the planning document to reflect final amount
- Set time to integrate all sub-teams' work together before submission

Roles and Responsibilities Document Evaluation:

- **Risks That Could Impact Completion :** They have mentioned the "extreme" flexibility they can opt for in their timeline:

“The project timeline is meant to be followed with extreme flexibility (After all, there might be instances where one task may take longer than others). In this event, it is the responsibility of the team leader to adjust the timeline based on the delayed task.”

Although flexibility should be allowed, the emphasis on “extreme” flexibility may negatively impact project completion. After all, the integration team is responsible for assembling all components, and the GUI team depends on them to complete their work before they can begin integrating the interface. If the integration team takes significantly longer than expected, it could delay all teams, leaving the GUI team with insufficient time to complete their tasks. While adaptability is necessary, excessive flexibility can lead to poor accountability and missed deadlines.

- **Suggestion:** Certain tasks, such as integrating the authentication team’s work with the leaderboard matchmaking team’s work, should have fixed deadlines to ensure that the GUI team can begin their coding on time. While some flexibility and adaptability can be allowed for other tasks, there should be a clear plan or mechanism in place to track delays, particularly for tasks that are more likely to require extra time, such as integrating leaderboard matchmaking with game logic or game logic with networking. The rest of the team should be informed in advance about potential delays so they can make necessary adjustments (or prepare backup plans beforehand). Like a member from the game logic or networking team could temporarily assist the integration team to get the work done on time, or the GUI team could adjust their deadlines by adding alternative approaches. In short, yes delays may occur, but there needs to be a plan to overcome those delays or “mitigate” their impact and this should not be done at the last moment, this should already be included in the planning document.
- **Task Dependencies:** The planning document for roles and responsibilities does not mention task dependencies. This is primarily because there is no listed responsibility for consistently checking on other teams' progress, even though the integration team's tasks are entirely dependent on the completion of other teams' tasks. As a result, integration tasks may be blocked if prerequisite components are not ready, but there is no clear mention of such issues.
 - **Suggestion:** : Add an additional responsibility for the team leader to maintain constant communication with other team leaders to ensure that everything remains on track. In the event of a missed deadline by another team, the integration team should develop a plan and adjust their own deadlines accordingly.
- **Resource Allocation:** Similarly to task dependencies, there is no mention of resource allocation among team members. The document only specifies that two team members will work on use case descriptions while the other two focus on use case diagrams. However, this allocation is limited to iteration 1, with no planning for task assignments in iteration 3, which involves the programming phase.
 - **Suggestion:** Since integration is a crucial part of the project, with multiple components needing to be assembled into a cohesive unit, there should be a clear plan outlining which team members will be responsible for integrating specific parts. We suggest assigning two members to integrate the authentication team with leaderboard matchmaking, while another two or three members focus on integrating networking with leaderboard matchmaking and subsequently with

game logic. Although the integration team will ultimately work together, assigning specific tasks to different members while having constant communication will prove to be more efficient.

Additionally, in cases where a team member does not fully understand another team's code, having different people assigned for integration of different components would be beneficial. Therefore, if an issue arises, they will be responsible for reaching out to the appropriate team for clarification. This approach ensures that knowledge gaps are addressed efficiently without unnecessary delays (where the team members would not know who is responsible for reaching out to the other team). Moreover, by establishing clear points of contact, the other teams (while coding) would also know who to relay the necessary information to, reducing miscommunication and enhancing overall project coordination.

GUI Planning Review:

Timeline Issues, Missed Milestones, or Unrealistic Deadlines:

- Set Week 2: March 10-14 for designing the GUI but did not account for the main purpose of iteration 2 which was to review another team's project
 - Did not allocate time anywhere in the timeline for evaluating another group's work which would push back their entire plan. Would this put them behind schedule? Would this affect the other sub-teams?
- Set all of iteration 1's deliverables (use case diagrams, use case descriptions, class diagrams, planning docs) to be completed in 1 week instead of weeks 1 and 2 which might also be unrealistic
- Did not make a plan to finish early enough to give the integration team time to get everyone's work together

Suggestions:

- Adjust week 2 tasks to allow time for reviewing another team's project.
- Should give more time for Iteration 1 deliverables to avoid rushing
- Plan to finish earlier than the deadline so the integration team can start on time

Roles and Responsibilities Document Evaluation:

- One of things we really liked about GUI's team planning document for roles and responsibilities is the clear breakdown of specific roles like "Game Interface Developer, Chat Interface Developer etc." with a clear mention of all the tasks needed to be completed in each role. It would give the team members assigned to that role a clear vision of what is expected from them, and would help them choose the roles accordingly. In short, the team has done a great job at resource allocation.
- Along with each role, the team explicitly mentions what other teams they will be working with which shows that GUI team has adopted a structured approach during their planning stage where they are actually thinking about the programming part of the project.
- Something they went above and beyond with is planning the basic rough drafts of GUI interfaces → it already gives them a head start for the programming part.

- Just a side note, it would have better if they combined their timeline with the planning document because the planning document mentions all the tasks that need to be completed and integrating it with timeline would make it easier for the team members to refer back to the planning documents and be aware of when each member is expected to complete exactly what task.
- **Suggestion:** Develop alternative plans or approaches in case a team member is unable to fulfill their assigned tasks. The document outlines that each member is responsible for two roles, but it does not specify what happens if one or more members are unable to complete their tasks. In such cases, responsibility should be reassigned to the team member whose existing tasks are most closely related to the uncompleted work. This ensures a smoother transition and minimizes disruption to the workflow.

Leaderboard and Matchmaking Planning Review:

- Unlike other teams, having one document that explains the timeline and roles and responsibilities is a good approach which makes the planning approach structures.
- It is noteworthy how they have clearly defined the overall goal of their team (before listing individual responsibilities), and specifically stated the features the team is expected to implement in the system. After all, it is a “group” project. Undoubtedly, every member’s contribution matters but defining the overall group’s objective helps work as a “team”.
- We noticed how the leaderboard and matchmaking team has specified the time period when they set up the repository and communication channel which indicates that they have been actively communicating since the beginning of the first iteration.
- We really like how instead of having fixed deadlines, they have a time period that a certain task needs to be completed by. It allows flexibility in the tasks which proves to be a positive factor in influencing the psychology of the team members but not to an extent which would impact completion.
 - **Suggestion:** In the week of March 22-28, they have vaguely mentioned “finish most of the coding”. Adding specific tasks (like they did with other roles and responsibilities) would be beneficial. Decide as a team what part of coding should take priority (probably the parts of the code that other teams might need to use?). The objective of the planning document is that team members can refer back to this document at any stage of the project, so it would be effective to mention/ plan how to split up the coding? Are all 4-5 members working in the same class, if not then how are you guys splitting up the tasks? Have a clear **resource allocation** for iteration 2 and 3.
 - Consider emphasizing the need for continuous communication with other teams throughout the programming process, not just at the end when integrating the code. They do mention collaborating with the integration team at the end, however they do not address the importance of ongoing discussions to prevent complications later. For example, maintaining communication with the authentication team about storing player’s statistics would ensure seamless integration, as they are responsible for displaying those statistics. Additionally, it would be helpful to outline task dependencies and a plan for managing them to avoid potential roadblocks.

Authentication Planning Review:

Timeline Issues, Missed Milestones, or Unrealistic Deadlines:

- Final number of use cases different from what is stated in the plan.
- Did not mention class structure diagrams.
- Did not make a plan to finish early enough to give the integration team time to get everyone's work together.

Suggestions:

- Could change the number of use cases in the planning document to reflect the final amount.
- Could include class structure diagrams to make the coding easier for later.
- Plan to finish earlier than the deadline so the integration team can work.

Roles and Responsibilities Document Evaluation:

Positive Aspects:

- No notable positive aspects.

Areas for Improvement:

- The document lacks proper planning; it only lists general requirements without outlining an approach for future iterations.
- Roles and responsibilities of team members are not defined.
- There is a task breakdown in the timeline, but it does not clarify individual contributions.
- No mention of the group leader's role in keeping the team aligned.

Suggestions:

- Divide tasks among different group members to ensure efficient resource allocation.
- Identify dependencies with other teams and maintain regular communication.
- Develop a structured approach, including contingency plans and alternatives for missed deadlines.
- Integrate roles and responsibilities within the timeline to ensure clarity.

Game Logics and Mechanics Planning Review:

Timeline Issues, Missed Milestones, or Unrealistic Deadlines:

- Did not allocate time for creating the planning documents.
- Mentioned diagrams but did not mention use case descriptions therefore not allocating time for them in the iteration 1 deliverables.
- Stated that both weeks dedicated to iteration 2 would be used for defining game mechanics and coding for the games.
 - Did not allocate time anywhere in the timeline for evaluating another group's work which was the entire purpose of iteration 2 and could push back their entire plan. Would this put them behind schedule? Would this affect the other sub-teams?

- Did not make a plan to finish early enough to give the integration team time to get everyone's work together,

Suggestions:

- List all planning tasks in the document with the timeline
- Add use case descriptions so they don't get skipped
- Adjust iteration 2 to account for reviewing another team's work
- Plan to finish earlier than the deadline so the integration team can work

Roles and Responsibilities Document Evaluation:

Positive Aspects:

- Well-defined milestones with a clear breakdown of tasks for iteration 3.

Areas for Improvement:

- While the timeline is well-structured, there is no planning for its execution.
- Group members appear to be working on structure diagrams individually without prior team discussions.
- Roles and responsibilities of each team member are not clearly explained.
- The first iteration should have included a more detailed breakdown of individual tasks rather than a general task list.
- The statement "Team works collaboratively, following the agreed-upon roles and guidelines" is vague and does not specify the collaborative work being done.

Suggestions:

- Improve resource allocation by clearly defining each member's responsibilities.
- Ensure structured discussions before working on diagrams to maintain consistency.
- Provide a breakdown of individual tasks for each iteration to enhance accountability.
- Specify what each team member is expected to contribute each week.

Networking Planning Review:

Timeline Issues, Missed Milestones, or Unrealistic Deadlines:

- Set goals to evaluate other teams' work and start drafting networking functionality all within week 2 which might be unrealistic (since it's a heavy workload)
- Did not mention anything about class structure diagrams in their planning
- Final number of use cases different from what is stated in the plan
- Did not make a plan to finish early enough to give the integration team time to get everyone's work together

Suggestions:

- Add class structure diagrams to avoid confusion
- Could change number of uses cases in the planning document to reflect final amount
- Plan to finish earlier than the deadline so the integration team can work

Roles and Responsibilities Document Evaluation:

Positive Aspects:

- Requirements are directly derived from the project description.
- Detailed task breakdowns clearly define what each component needs, aiding in resource allocation.
- Roles and responsibilities are assigned explicitly.
- The team has established communication with other teams to prevent excessive workload on the integration team.
- An Integration Liaison has been assigned to ensure efficiency.
- Accountability measures are well-defined.
- No fixed deadlines, but time periods provide flexibility, benefiting team psychology.
- Expectations are clearly stated.

Areas for Improvement:

- No clear approach for handling situations where a team member fails to complete their tasks.
- No contingency plan for task redistribution in case of non-compliance.

Suggestions:

- Establish a protocol for addressing incomplete tasks, including accountability measures.
- Define who would take responsibility for unfinished work and how it will be managed.
- Continue maintaining clear task expectations while incorporating a backup plan for contingencies.