

Planning analysis

Bravo team's workflow planning was fairly well done and demonstrates that the team was mostly communicative and collaborative during their planning process. This team did a good job at creating plans which reflected the expected deliverables of this Project. The sub-teams that stand out as being particularly well organized are the Networking and Authentication teams as they had schedules which were closely aligned and encountered no conflicts with each other. While not as obviously in sync, the Matchmaking/ Statistics team stuck decently close to this schedule as well. The weaker sub-teams in terms of planning were the GUI team - with its rather broad but compliant schedule - and the Game Logic team – with a schedule that unfortunately causes scheduling conflicts with other sub-teams.

Scheduling:

Bravo team has a few major scheduling issues that should be addressed. These issues are composed of unrealistic estimations of workload, conflicts between sub-team tasks, missed milestones, and some potential bottlenecks.

Unrealistic Deadlines:

1. The Game Logic sub-team is scheduled to complete the basic integration of all four of their games as early as March 15th, which is only halfway through Iteration 2.
2. The Game Logic sub-team is also scheduled to work on integration with the Networking and Matchmaking/ Statistics teams as early as March 21st, which is only the end of Iteration 2.

This scheduling is unrealistic in a few ways: It is somewhat unrealistic to expect to complete the basic implementation of 4 entire games in just one week because it is a lot of work that is meant to be completed during the much longer Iteration 3. It is yet more unrealistic to expect that once that is done, the Networking and Matchmaking/ Statistics team will simultaneously be ready for implementation work within that same short time span, compounded by the fact that this integration is expected to be finished by only the end of Iteration 2. These deadlines are most unrealistic in that they are scheduled at the same time as the Iteration 2 deliverables are supposed to be worked on, which would not likely be possible with the expected workload.

Scheduling Conflicts:

1. All sub-teams (excluding the Game Logic team) are scheduled to work on and complete their Use case Descriptions/ Diagrams and Structure Diagrams, which are needed before the Integration team can consolidate those files together by the end of Iteration 2 as they are meant to.
2. The Game Logic team is scheduled to integrate with other sub-teams before the end of Iteration 2 even though those sub-teams are busy with Iteration 2 Deliverables.

These conflicts arise from the Game Logic team being scheduled for tasks which are both unrealistic and not the actual focus of Iteration 2 while the other teams are scheduled to do the appropriate work. The Game Logic team is both unavailable to help their group, and the other sub-teams are unavailable to help the Game Logic team.

3. Once the other sub-teams have a sufficient codebase for integration early on in Iteration 3, the Game Logic team is scheduled to already be done with much of their code when the other teams are trying to integrate.
4. The Integration team is scheduled to consolidate everyone's code into one system to start documentation and testing by the midpoint of Iteration 3, but the Networking, Authentication, and Matchmaking/ Statistics teams are scheduled to conclude their independent documentation and testing by the end of Iteration 3.
5. The Game Logic team is not scheduled to help the rest of their group documenting suggested improvements and changes provided by other groups when all the other sub-teams require that help.

These conflicts are less serious and are more easily addressed, stemming from similar issues of mismatched work priorities between teams when planning their schedules. Because these conflicts are within Iteration 3 it should be trivial to rearrange the plan to account for them.

Scheduling Bottlenecks:

1. The Game Logic team's focus on coding instead of Iteration 2 Deliverables could potentially cause a severe pileup of responsibilities on their team as they juggle too much work at once, which could result in deadlines being missed or requiring time be put into rearranging schedules to alleviate workload.
2. The Networking, Authentication, and Matchmaking/ Statistics teams' late documentation and testing may put more work onto the Integration team as they need to accommodate for new tests and documentation that doesn't follow their consistency guidelines.

These are important to fix before they become a problem. Unfortunately, Iteration 2 already happened so Bravo team may be struggling with the first bottleneck already. Rearranging scheduled workloads to line up with each other's schedules will largely fix the problem.

Milestones:

Bravo team did quite well to prioritize the right goals to ensure efficient project completion and flow, however there were some missed deadlines, some incorrectly prioritized goals, and some goals that we recommend for them to prioritize in Iteration 3.

Missed Deadlines:

1. The Integration team did not consolidate the Iteration 2 deliverables into two files

This issue was potentially caused by scheduling conflicts with the Game Logic team.

Good Priorities:

1. Authentication and Networking Infrastructure in Iteration 1
2. GUI and Backend Integration in Iteration 3
3. Matchmaking and Leaderboard Design in Iteration 1 and 2

These aspects are all well-defined, consistent, and integral to ensuring the project works as is expected.

Incorrect Priorities:

1. Integration happens late into Iteration 3 when each team's systems are largely complete
2. Load testing and error handling are delayed to the end of Iteration 3
3. Networking API integration and testing are delayed to the end of Iteration 3

These aspects are well defined but take place far too late into the project to account for issues and may result in incomplete work by the end of Iteration 3.

Recommended Priorities:

1. Perform integration testing before full integration so there aren't surprises
2. Consistent load testing and performance validation
3. Consistent error handling and logging validation

These aspects will help keep the project flow efficient and eliminate surprises and hangups later in the project when time is precious.

Potential Risks:

Bravo team has some notable risks that should be addressed. These risks are potential and composed of scheduling conflicts, integration and synchronization concerns, inter-sub-team dependencies, technical issues, and user experience difficulties.

Timeline Risks:

- The Game Logic team's timeline to create the basic code for four different games in just one week clashes with every other team's significantly and this may cause blockers for progress.
- The Networking, Authentication, and Matchmaking/ Statistics teams' schedules are lined up well but have conflicts with the Integration team.

Integration Risks:

- The Networking Team identifies the risk of needing to ensure alignment with project goals, which implies a potential risk of misalignment if this is not carefully managed.
- The Project Plan Integration document mentions the need for heavy communication with other teams, indicating that integration could be a complex process with associated risks if communication is not effective.

Dependency Risks:

- Inter-team dependencies, such as GUI coordination with Networking and Authentication Teams, may cause delays if not managed well.

Technical Risks:

- Performance bottlenecks in networking and potential security vulnerabilities could impact reliability. Solution: Implement strong design principles, robust testing, and proactive security measures.

User Experience Risks:

- Lack of feedback for failed actions (e.g., unsuccessful login) may lead to frustration. Solution: Implement clear error messages and intuitive UI feedback.

Suggested Improvements:

The Game Logic Team's schedule, if it hasn't been already, needs to be adjusted to fit in with every other sub-team's schedule. The expectations are both unrealistic and generate conflict within the workflow of the group's project. Missed milestones need to be rectified during the improvement documentation process. Priorities should be adjusted according to the recommended priorities and incorrect priorities detailed above. To mitigate risks improve planning with communication and added buffer time, implement more frequent error testing, and implement standardized coding practices to ensure easier integration and workflow for all team members. By identifying and addressing risks and conflicts proactively, Bravo team can minimize disruptions and ensure a high-quality product!