1.A

Here are the steps to create an agile project plan for your new software feature:

1. **Define Clear Objectives:** Start by understanding the purpose of your project. What problem does the software feature solve? Define clear objectives that align with your stakeholders’ needs.
2. **Break Down Objectives into Manageable Tasks:** Divide your objectives into smaller, manageable tasks. These tasks will form the basis of your project plan. Consider using user stories to capture specific functionality from an end-user perspective.
3. **Estimate Effort for Each Task:** Assign estimated effort (usually in story points) to each task. Story points represent the relative complexity or effort required to complete a task. Use techniques like Planning Poker or T-shirt sizing to estimate.
4. **Prioritize Tasks:** Prioritize your tasks based on their value and dependencies. Consider the impact of completing each task on the overall project. High-priority tasks should be addressed early in the project lifecycle.
5. **Create a Backlog:** Compile all the tasks into a product backlog. The backlog serves as a dynamic list of work items that can be adjusted as needed. It includes user stories, bug fixes, enhancements, and technical tasks.
6. **Plan Iteratively:** Agile projects are broken down into iterations (sprints). Each sprint typically lasts 1-4 weeks. During sprint planning, select a subset of tasks from the backlog to work on. These tasks become the sprint backlog.
7. **Daily Stand-up Meetings:** Conduct daily stand-up meetings (also known as daily scrums) during each sprint. In these brief meetings, team members discuss progress, identify roadblocks, and ensure alignment.

**2.**A

Team gathers around for the Daily Stand up meeting. The Product Owner, Scrum Master, and Development Team members are present.

Scrum Master: Good morning, team! Let's kick off our Daily Stand up. Who wants to start?

Developer 1: I can go first. Yesterday, I was able to complete the profile picture upload feature. Today, I'm planning to start working on the bio section. No blockers.

Scrum Master: Great progress, Developer 1! How about you, Developer 2?

Developer 2: Yesterday, I tackled the customization options and made some good headway. Today, I plan to dive deeper into refining the colour scheme feature. No blockers from my end either.

Product Owner: Thanks for the update, Developer 2. How about you, Developer 3?

Developer 3: Yesterday, I started integrating the social media authentication for profile connections, but I hit a snag. The API documentation is a bit unclear about some authentication endpoints, so I'm struggling to proceed.

Scrum Master: That's a common challenge when dealing with new APIs. Developer 3, how much time do you think you'll need to research and resolve this?

Developer 3: I'd say another half-day should be sufficient to troubleshoot and find a solution.

Scrum Master: Alright, let's make sure you have the support you need. If you run into any roadblocks, feel free to pair up with Developer 2 or reach out to me. We can also leverage resources like online forums or documentation. Does that sound good?

Developer 3: Absolutely, thanks, Scrum Master.

Scrum Master: Okay, let's keep the momentum going. Does anyone else have anything to share?

Other team members provide their updates, and the meeting concludes with a plan to address the API documentation challenge.

Scrum Master: Excellent! Thanks, everyone, for the updates and collaboration. Remember, we're a team, and we're here to support each other. Let's tackle this challenge together and keep pushing forward. If anyone needs assistance or encounters any new blockers, don't hesitate to reach out. Have a productive day!

Team members disperse to continue their work.

End of Meeting

In this Daily Stand up simulation, the team effectively communicates their progress and plans while also addressing a common challenge related to API documentation clarity. The Scrum Master ensures Developer 3 has the support needed to overcome the challenge and encourages collaboration within the team. This proactive approach fosters a supportive environment where team members can openly discuss and resolve obstacles together.