Name: Nina Jaira Lael B. Cagomoc

**1. How to do Iteration Planning Meeting?**

An iteration planning meeting is a crucial part of Agile software development, where the team plans the work for the upcoming iteration (sprint). Here's a step-by-step breakdown of how to conduct an effective iteration planning meeting:

1. **Provide an update on the previous iteration:** 
   1. Start the meeting by discussing the outcomes of the previous iteration. Review what was accomplished, what wasn't, and any lessons learned.
2. **Ensure team availability:**
   1. Make sure that all necessary team members are present for the meeting, including developers, testers, and any stakeholders.
3. **Prioritize and estimate stories:**
   1. Review the backlog and prioritize user stories based on business value. Estimate the effort required for each story using story points or other estimation techniques.
4. **Set goals:**
   1. Define the goals for the upcoming iteration. What do you aim to achieve? What user stories should be completed?
5. **Elaborate on planning goal:**
   1. Discuss the details of each prioritized user story. Break down the tasks needed to complete them and estimate the time required for each task.
6. **Share new information:** 
   1. If there are any changes or new information that impacts the plan, share it with the team. It could be new requirements, technical constraints, or dependencies.
7. **Discuss issues or concerns:**
   1. Allow team members to express any concerns or issues they foresee during the iteration. Address these concerns and find solutions collaboratively.

By the end of the iteration planning meeting, the team should have a clear understanding of the goals, tasks, and priorities for the upcoming sprint.

**2. How to do Kick-off (Dev Kick-off with BA and QA)?**

A project kick-off, especially involving developers, business analysts (BA), and quality assurance (QA), sets the tone for successful collaboration. Here's a detailed breakdown of conducting a kick-off:

1. **Introduce the process:**
   1. Start by introducing the client to the company's development process. Explain the workflow, roles, and responsibilities of each team member.
2. **Prepare a digital product roadmap:**
   1. Outline the project timeline with clear deadlines. Discuss key milestones, releases, regression testing, sprint durations, and any error margins.
3. **Know your product:**
   1. Gather insights from the client about the most important features, expected outcomes, target users, and user experiences. Ask questions to understand their perspective fully.
4. **Know your scope:**
   1. Clarify your role and responsibilities. Define what aspects of the project you are covering, what you are responsible for, and the team and client expectations from you.
5. **Know your deliverables:**
   1. Set expectations with the client on how you'll be sharing milestones and goals. Discuss report schedules, deadlines, and expected coverage of your work.
6. **Create a proposal:**
   1. Based on the gathered information, create a proposal that outlines the QA plan, objectives, deliverables, and action items. Include details about types of testing, scope, code standards, validation pipeline, and more.
7. **Present your plan:**
   1. Present the proposal to the client, the development team, and other stakeholders. Use this presentation to align visions and gain commitment.
8. **Formalize the plan**:
   1. Get the plan signed by both the team and the client to ensure agreement. This adds formality and commitment to the process.

Remember that open communication is key throughout the process to build confidence and trust among team members and clients.

**3. How to do Desk Check (Dev Desk Check with BA and QA)?**

A desk check is a manual process of reviewing code, logic, and algorithm correctness. Here's how to perform a desk check:

1. **Walkthrough the pseudo-code:**
   1. Manually go through every line of the pseudo-code (code-like description of the algorithm) to identify logic errors, bugs, and anomalies.
2. **Identify bugs in logic:**
   1. Payattention to the logical flow of the algorithm. Check if conditions, loops, and control structures are correctly implemented and producing the expected results.
3. **Algorithm verification:**
   1. Verify that the algorithm's behavior aligns with the intended logic. Ensure inputs and expected outputs match the desired outcomes.
4. **Use tables for tracking:**
   1. Create a table with columns for line number, variable values, conditions (if applicable), input-output pairs, and expected results. Use this table to track your walkthrough and identify discrepancies.

Desk checks help catch logical errors that might not be immediately apparent through automated testing. They play a crucial role in ensuring algorithm correctness and overall code quality.