**Scenario 1: Agile Methodology**

1. **How does the Agile approach help manage evolving requirements in this scenario?**  
   Agile allows for iterative development, meaning stakeholders can introduce new features like mobile integration during each cycle. This flexibility helps adjust to changes without significant delays.
2. **How can the team prioritize new features against existing ones in an Agile environment?**  
   The team prioritizes features based on their value and urgency. A Product Backlog is maintained, and each new feature is reviewed and added based on its priority and the project goals.
3. **What role do regular sprint reviews play in managing evolving stakeholder needs?**  
   Sprint reviews allow the team to demonstrate progress, get feedback from stakeholders, and adjust priorities for the next sprint. This ensures that the evolving needs are continuously met.
4. **How can the Agile team adjust their timelines to accommodate new requirements without compromising quality?**  
   The team can adjust timelines by adding new items to the backlog, reassigning tasks, and potentially extending future sprints, ensuring new features are developed without sacrificing quality.
5. **How does iterative feedback help ensure that the system meets stakeholders' needs over time?**  
   Iterative feedback allows the team to refine and adjust the system continuously, ensuring that each iteration brings the product closer to what the stakeholders need.

**Scenario 2: Waterfall Model**

1. **How does the Waterfall model impact the handling of evolving requirements after the project has already started?**  
   Waterfall is rigid, making it difficult to incorporate new requirements once the project has started, as each phase is completed before moving to the next, limiting flexibility.
2. **What challenges does the team face when trying to incorporate new features into a Waterfall-based approach?**  
   The main challenge is the lack of flexibility, as incorporating new features can require going back to earlier phases, delaying progress and increasing costs.
3. **How can the team handle scope changes in the Waterfall model without disrupting the overall project plan?**  
   Changes should be managed through a change control process, carefully assessing the impact on the timeline and resources, and adjusting the schedule and costs as necessary.
4. **What role does detailed upfront documentation play in managing requirement changes in Waterfall?**  
   Detailed documentation helps define clear requirements and minimizes scope changes. However, if changes occur, it provides a basis for understanding the impact and managing them more effectively.
5. **How can the team balance the need for sequential development with the evolving demands of stakeholders?**  
   Balancing this requires careful planning and early stakeholder involvement to ensure that their needs are captured before development begins, thus reducing the need for changes later.

**Scenario 3: Model-Driven Development (MDD)**

1. **How does Model-Driven Development help manage stakeholder feedback in the early stages?**  
   MDD emphasizes creating models first, allowing stakeholders to review and provide feedback before development, ensuring that their needs are reflected early in the process.
2. **How can the team ensure that updated models reflect stakeholder requests without impacting the development timeline?**  
   By using a flexible modeling approach and automating parts of the process, updates can be integrated without significant delays.
3. **What role does automated code generation play in making changes to models based on feedback?**  
   Automated code generation accelerates the process of turning updated models into working code, ensuring that changes based on feedback are quickly reflected in the system.
4. **How can the team validate models before starting the actual coding to ensure they meet stakeholder needs?**  
   Stakeholder reviews and automated validation tools can ensure that models accurately reflect requirements before development begins.
5. **How does MDD help ensure alignment between the final product and stakeholder expectations?**  
   MDD provides continuous validation of the system through models, ensuring that stakeholder requirements are consistently reflected as the project progresses.

**Scenario 4: Agile Methodology (Changing Stakeholder Priorities)**

1. **How can the team incorporate changing stakeholder priorities in an Agile environment?**  
   The team can adjust priorities by reviewing the Product Backlog, and re-prioritizing features based on stakeholder needs and business value.
2. **How do Agile sprints help manage shifting priorities while ensuring progress?**  
   Sprints provide time-boxed periods to focus on specific tasks, allowing the team to adjust priorities while still making progress on the current iteration.
3. **How does the Product Backlog accommodate changing requirements and priorities?**  
   The Product Backlog is a dynamic list that is constantly updated to reflect new requirements and priorities, allowing for quick adjustments.
4. **How can Agile's flexibility support both urgent and new feature requests?**  
   Agile allows for frequent re-prioritization of the backlog, enabling the team to add new features and address urgent requests promptly.
5. **How does the iterative nature of Agile development support the team in revising priorities?**  
   Agile's iterative cycles allow the team to revise priorities at the end of each sprint based on stakeholder feedback, ensuring that the most important tasks are always prioritized.

**Scenario 5: Waterfall Model (Late-Stage Changes)**

1. **How can early stakeholder engagement prevent late-stage changes in a Waterfall-based project?**  
   Early engagement helps clarify stakeholder needs upfront, reducing the risk of late-stage changes that can disrupt the schedule.
2. **What impact does late-stage requirement modification have on the project in Waterfall?**  
   It can lead to significant delays, increased costs, and rework, as the project must go back to earlier phases for adjustments.
3. **How can thorough requirements analysis in the Waterfall model help avoid changes later on?**  
   Detailed requirements analysis helps identify all needs upfront, reducing the likelihood of changes during later stages.
4. **How can the team better estimate the timeline and resources in Waterfall considering changes in stakeholder needs?**  
   By conducting regular reviews and updating estimates based on stakeholder feedback, the team can adjust the project plan to account for new requirements.
5. **What processes can be introduced to improve the flexibility of the Waterfall model in accommodating changes?**  
   Introducing a formal change management process can help handle new requirements and minimize disruptions.

**Scenario 6: Agile Modeling and User Stories**

1. **How can user stories be used to document new requirements in Agile methodology?**  
   User stories describe the functionality from the user's perspective, capturing new requirements in a clear and understandable format.
2. **How does the iterative process of Agile development support the continuous evolution of user stories?**  
   As feedback is gathered, user stories are continuously refined to reflect the evolving needs of stakeholders.
3. **What role does collaboration between developers and stakeholders play in refining user stories?**  
   Collaboration ensures that user stories are accurately defined and aligned with stakeholder expectations.
4. **How can Agile modeling tools help visualize new features or changes based on stakeholder feedback?**  
   Agile modeling tools help create visual representations of features and processes, making it easier to incorporate feedback and visualize changes.
5. **How can the team manage changes in user stories without disrupting development cycles?**  
   Changes in user stories can be incorporated into the next sprint, allowing the team to adjust priorities without impacting current work.

**Scenario 7: Model-Driven Architecture (MDA)**

1. **How can MDA help the team integrate new features into the system without reworking the entire architecture?**  
   MDA allows for the separation of concerns, enabling changes to be made to models without affecting the overall architecture.
2. **What benefits does MDA provide when incorporating changes into an existing model?**  
   MDA's flexibility allows new features to be integrated easily, without disrupting the existing system, and supports reusability.
3. **How does the separation of concerns in MDA help in handling evolving requirements?**  
   MDA separates different aspects of the system, allowing changes to one part of the system without affecting others, making it easier to adapt to new requirements.
4. **How can the team ensure that model updates reflect new features requested by stakeholders?**  
   Regular stakeholder reviews and model validation ensure that updates align with stakeholder needs.
5. **How can automated testing in MDA ensure that updates align with system functionality?**  
   Automated testing ensures that any changes made to the model are validated against the system’s functionality, preventing errors.

**Scenario 8: Scrum Framework (Handling Scope Changes)**

1. **How does the Scrum framework help the team handle scope changes without impacting timelines?**  
   Scrum uses iterative sprints, where scope changes can be reviewed and prioritized, allowing adjustments without affecting the overall project timeline.
2. **How does the Product Owner's role in Scrum affect how evolving requirements are handled?**  
   The Product Owner is responsible for managing the Product Backlog, ensuring that evolving requirements are prioritized and aligned with business needs.
3. **How can Scrum retrospectives help improve processes for managing changes in scope?**  
   Retrospectives allow the team to reflect on how scope changes were handled, identify improvements, and apply them in future sprints.
4. **How can the Scrum team ensure that new features align with stakeholder needs during each sprint?**  
   The Scrum team works closely with the Product Owner and stakeholders to ensure that new features are well-defined and aligned with business goals during each sprint.
5. **How can Scrum's flexibility support scope adjustments without delaying the final product?**  
   Scrum's flexible structure allows the team to adjust scope in each sprint, balancing new features with the need to meet deadlines.

**Scenario 9: Waterfall Model (Scope Creep)**

1. **How can scope creep be prevented during the requirements phase of a Waterfall project?**  
   Clearly defining and agreeing on the project scope early on, and involving stakeholders in detailed requirement gathering, helps prevent scope creep.
2. **How does the rigid structure of the Waterfall model make it harder to accommodate new features?**  
   The Waterfall model is linear and sequential, making it difficult to incorporate new features once development has begun.
3. **What strategies can the project manager use to control scope creep in Waterfall?**  
   Strict change control processes, regular reviews, and stakeholder engagement can help prevent scope creep.
4. **How can the team ensure the original requirements are met before adding new features?**  
   The team must ensure that the baseline requirements are fully implemented and validated before introducing any new features.
5. **How does the Waterfall model’s fixed timeline impact the integration of additional features?**  
   Waterfall’s fixed timeline can make it difficult to integrate additional features, as each phase needs to be completed before moving on, potentially delaying the entire project.

**Scenario 10: Agile Feedback Loops**

1. **How can feedback loops help incorporate stakeholder needs into the development process?**  
   Feedback loops enable continuous stakeholder input, allowing the team to make adjustments in each iteration to ensure the final product meets their needs.
2. **What role does stakeholder engagement during sprint reviews play in gathering feedback?**  
   Sprint reviews are an opportunity for stakeholders to provide feedback on the work completed in each sprint, guiding future development.
3. **How can the team adjust priorities based on feedback received after each iteration?**  
   The team can update the Product Backlog to reflect new priorities and adjust the scope of upcoming sprints.
4. **How does Agile's flexibility in feedback loops ensure that requirements evolve appropriately?**  
   Agile allows feedback to be integrated continuously, ensuring that the system evolves in response to changing requirements.
5. **How can the team ensure that changes based on feedback are manageable and don't derail the project?**  
   Changes should be managed through careful backlog prioritization and ensuring that the team has the capacity to handle adjustments without impacting deadlines.

**Scenario 11: Model-Driven Testing (MDT)**

1. **How does MDT help the team quickly adapt to new requirements during the testing phase?**  
   MDT allows for the rapid generation of tests based on updated models, ensuring that changes in requirements are quickly reflected in the testing phase.
2. **What impact does stakeholder-driven change have on the test models in MDT?**  
   Stakeholder-driven changes can require updates to the test models to ensure that new functionality is correctly validated.
3. **How can automated testing in MDT ensure that new features work without breaking existing functionality?**  
   Automated tests can be re-executed after changes to verify that new features do not disrupt existing functionality, ensuring stability.
4. **How does MDT contribute to continuous integration when requirements change frequently?**  
   MDT enables frequent testing as part of the continuous integration process, allowing for quick identification and resolution of issues related to changing requirements.
5. **How can MDT be used to validate that new features meet stakeholder expectations?**  
   MDT ensures that new features are tested against stakeholder requirements, confirming that they meet the desired outcomes.

GIVE ANSWER

**ChatGPT said:**

ChatGPT

**Scenario 12: Kanban Methodology**

1. **How does Kanban help visualize the impact of new requirements on the project flow?**  
   Kanban helps visualize the impact by providing a board where each task is represented as a card. New requirements are added as new tasks, and the flow of work is visually tracked through different stages like "To Do," "In Progress," and "Done," making the impact of new tasks on existing work clear.
2. **How can the team manage increasing work items on the Kanban board without overloading the team?**  
   The team can limit the work in progress (WIP) for each column to prevent overload. By setting WIP limits, the team can ensure they focus on completing tasks before starting new ones, helping to maintain balance.
3. **How does Kanban's flexibility support ongoing changes in requirements during development?**  
   Kanban's flexibility allows the team to continuously add, adjust, and prioritize tasks as new requirements come in. Tasks can be rearranged or reassigned without disrupting the flow of work, making it easy to accommodate changes.
4. **How can the team prioritize tasks in Kanban to ensure important features are developed first?**  
   The team can prioritize tasks by arranging them in order of importance on the Kanban board. Higher-priority tasks are placed at the top, ensuring that the most critical features are developed first.
5. **How does the continuous flow of tasks in Kanban align with the need for rapid changes in feature requests?**  
   Kanban's continuous flow allows tasks to be worked on and completed as soon as possible, ensuring that new features can be integrated quickly. This rapid, flexible process is ideal for handling frequent changes in feature requests.

**Scenario 13: Model-Driven Requirements Engineering (MDRE)**

1. **How does MDRE help track changes to requirements throughout the project lifecycle?**  
   MDRE uses models to represent requirements, making it easy to track changes through version control and traceability. The impact of each change can be traced back to the corresponding requirement in the model, ensuring all changes are documented.
2. **How can the team ensure that new requirements are incorporated into the model efficiently?**  
   The team can update the models as new requirements come in, ensuring they reflect the latest changes. Using modular models allows for easier updates without disrupting existing structures.
3. **How does the traceability of requirements in MDRE help with impact analysis?**  
   Traceability allows the team to see how changes to one requirement affect others, making it easier to analyze the impact of new or changed requirements. This helps in understanding how the change might affect the system as a whole.
4. **How can MDRE reduce the risk of missing important features when requirements evolve?**  
   MDRE ensures that each change is tracked and incorporated into the model, reducing the risk of missing important features. Because the models are continuously updated, any new requirements will be included in the system design.
5. **What role does MDRE play in ensuring that the final system meets stakeholder expectations?**  
   MDRE provides clear documentation of requirements, ensuring that every feature and change aligns with stakeholder expectations. The models serve as a basis for validation and verification to confirm that the final system meets the agreed-upon requirements.

**Scenario 14: Waterfall Model with Fixed Budget**

1. **How can the project manager assess the impact of additional features on the budget in Waterfall?**  
   The project manager can assess the impact by comparing the cost of new features against the allocated budget, considering development, testing, and integration costs. This analysis helps in identifying if additional features can be accommodated within the fixed budget.
2. **What strategies can the team use to accommodate new features within a fixed budget?**  
   The team can prioritize critical features, reduce the scope of less essential features, or delay the addition of new features to a future phase. They can also consider optimizing resources or reusing existing components to manage costs.
3. **How can the team identify critical features that need to be prioritized to avoid budget overruns?**  
   The team can conduct a prioritization exercise with stakeholders, focusing on the features that provide the highest value or are essential to the system's core functionality.
4. **How does the linear progression of Waterfall make it difficult to manage changes with a fixed budget?**  
   The rigid, step-by-step approach of Waterfall makes it difficult to accommodate changes once the project has started. Any addition of features or scope requires revisiting earlier phases, which can lead to delays and increased costs.
5. **How can careful documentation during the analysis phase help avoid scope and budget issues?**  
   Thorough documentation during the analysis phase ensures that all requirements are clearly defined and agreed upon by stakeholders. This helps in preventing scope creep and minimizes the risk of unforeseen changes later on.

**Scenario 15: Agile with Feature-Driven Development (FDD)**

1. **How does the team use FDD to ensure that features are prioritized based on stakeholder needs?**  
   In FDD, features are prioritized based on their value to the stakeholders. The team collaborates with stakeholders to identify the most critical features, ensuring that development focuses on what matters most.
2. **How can the Agile team manage multiple new feature requests without derailing the existing timeline?**  
   The team can break down the new feature requests into smaller, manageable tasks and add them to the backlog. By prioritizing them and planning them into future sprints, the team can ensure that they do not disrupt the current sprint's timeline.
3. **How does FDD ensure that features are broken down into manageable chunks for Agile sprints?**  
   FDD involves breaking down large features into smaller, well-defined pieces. These smaller features are then implemented incrementally, making it easier for the team to track progress and stay on schedule.
4. **How can the team handle shifting priorities and keep focus on delivering the most important features first?**  
   The team can regularly revisit the backlog, prioritize features, and ensure that the most valuable and time-sensitive features are delivered first. Frequent sprint reviews allow for reassessment of priorities.
5. **What techniques can be used to ensure that all key features are covered while still addressing stakeholder changes?**  
   Techniques like backlog refinement and feature prioritization can help the team address both new and existing stakeholder needs. The team should continuously review and adjust the backlog based on feedback.

**Scenario 16: Agile Development of Library Management System**

1. **How can the team use use cases to define the new "book reservation" feature in Agile?**  
   The team can add a new use case for the book reservation feature to the product backlog. They can collaborate with stakeholders to refine the details and ensure it meets the system's needs.
2. **How does the Agile methodology support the quick incorporation of the new feature into the current sprint?**  
   Agile allows for flexibility in adapting to new requirements, and the team can add the book reservation feature to the current sprint's backlog, adjusting other tasks if needed to accommodate the change.
3. **What changes should be made to the class diagram to accommodate the new "book reservation" functionality?**  
   The class diagram should include a new entity representing reservations, such as a "Reservation" class, along with associations to the "Book" and "User" classes to track which books are reserved and by which users.
4. **How can the team adjust their sequence diagrams to represent the new interactions in the system after the feature change?**  
   The team can update the sequence diagrams to include interactions between the user and the system for making a reservation, as well as the system's interaction with the book and reservation entities.
5. **How can the team ensure that the addition of the new feature doesn’t impact the completion of other features in the current sprint?**  
   The team can carefully plan the sprint backlog to allocate time for the new feature without overburdening the team. They can also break the feature down into smaller tasks to ensure smooth integration.

**Scenario 17: Waterfall and Mobile Integration**

1. **How does the Waterfall methodology handle the integration of a mobile app into the project after the design phase has been completed?**  
   In Waterfall, adding a mobile app after the design phase would require revisiting the design, possibly requiring significant rework. This can cause delays and disrupt the established timeline.
2. **What adjustments need to be made to the use cases to accommodate the new requirement for mobile integration?**  
   The team needs to revise the use cases to include scenarios for mobile interactions, ensuring that the user journey includes mobile-specific tasks such as mobile login and accessing the system from mobile devices.
3. **How should the class diagram be modified to include new mobile app-related entities and interactions?**  
   The class diagram should be updated to include entities related to the mobile app, such as "MobileUser" or "MobileDevice," and their interactions with the core system entities.
4. **How can the team update the activity diagram to reflect the changes in the system’s workflow due to the mobile app integration?**  
   The activity diagram should be updated to include mobile-specific actions, such as mobile user registration and login, and to show how these actions interact with the rest of the system.
5. **How does the lack of flexibility in Waterfall impact the ability to efficiently handle the addition of the mobile app?**  
   Waterfall's rigid structure makes it difficult to accommodate changes like mobile integration once the design phase is completed. This can lead to delays and require significant rework, as every phase is sequential.

**Scenario 18:**

1. **How can the team use sequence diagrams to visualize the interaction between new features like user profiles and existing system components?**
   * The team can update the sequence diagram to show interactions between the new "user profile" feature and existing components like the user interface, database, and authentication system. This helps visualize the flow of data when users create, update, or view profiles.
2. **How can the activity diagram help the team understand the impact of new features on the overall user journey in the system?**
   * The activity diagram can illustrate the steps the user will follow when interacting with the new features, such as creating or updating a profile, recommending books, or browsing recommendations. It helps understand how these new actions fit into the overall user workflow.
3. **How does the iterative nature of Agile allow the team to continuously refine and adapt the class diagram as new features are added?**
   * In Agile, as each iteration progresses, the class diagram is updated to reflect changes. The iterative feedback loop allows the team to adjust class relationships and entities to accommodate new features like user profile management and book recommendations, ensuring the design remains flexible and aligned with evolving requirements.
4. **How can the team ensure that new features are added without breaking the existing functionality using the sequence diagram?**
   * The team should use the sequence diagram to carefully map out interactions for the new features while checking for potential conflicts with existing interactions. Proper testing of sequences in the system will help ensure that the new features integrate smoothly without breaking the existing functionality.
5. **What steps can be taken to manage the evolving requirements in Agile while maintaining a cohesive system architecture?**
   * The team can prioritize requirements in the backlog, review and refine the architecture during each sprint, and ensure consistency between the class and sequence diagrams. Regular code reviews and testing will help maintain a cohesive architecture even as new requirements evolve.

**Scenario 19:**

1. **How can the team use use cases to define the new "book rating" feature within the existing Waterfall process?**
   * The team can create a new use case for the "book rating" feature, defining the steps a user takes to rate a book and the system’s responses. This use case will be incorporated into the existing requirements and documentation.
2. **What changes need to be made to the class diagram to represent the "book rating" feature and its interactions with other entities?**
   * The class diagram should be updated to include a new entity for ratings, which is associated with books and users. The relationships between users, books, and ratings must be clearly defined.
3. **How can the team update the sequence diagram to illustrate the flow of data when users submit ratings for books?**
   * The sequence diagram should show the steps involved when a user submits a rating: the user submits the rating, the system records it, and updates the book's average rating. This flow ensures the feature's integration is clear.
4. **How should the activity diagram be adjusted to show the user journey when interacting with the new "book rating" functionality?**
   * The activity diagram should be updated to include the new decision points and actions related to rating a book. This would include actions like selecting a book, submitting a rating, and viewing the updated rating.
5. **How does the Waterfall model’s structured approach help or hinder the addition of this new feature midway through the development process?**
   * The Waterfall model’s structured nature can make it challenging to add new features midway since the design phase has already been completed. However, the thorough documentation and planning at the start can help minimize the disruption caused by such additions, as long as the necessary changes are carefully managed.

**Scenario 20:**

1. **How can the team update the class diagram to include entities related to overdue book reports?**
   * The class diagram should include entities for overdue book reports, potentially a report class and a relationship to the user and book entities. It could include attributes like due date, fine, and report generation time.
2. **How will the activity diagram need to change to incorporate the new report-generation process?**
   * The activity diagram will need to show the steps for generating the overdue report, such as identifying overdue books, calculating fines, and generating the report. The user's actions of generating and viewing the report should also be represented.
3. **What impact will this new feature have on the sequence diagrams, especially the interactions between the user and the system when generating reports?**
   * The sequence diagram will need to be updated to reflect new interactions where the user requests an overdue report, the system checks for overdue books, calculates fines, and presents the report to the user.
4. **How can the Waterfall model’s rigid planning process affect the addition of this new feature midway through the project?**
   * The Waterfall model’s rigid structure may make it difficult to add new features after the design phase. However, by adhering to the existing planning and documentation processes, the team can better manage the integration of this new feature into the established project flow.
5. **What challenges will the team face in making changes to the already-completed design documents, such as the class and activity diagrams?**
   * The challenge lies in updating already-finalized documents without disrupting the entire development flow. This may require detailed rework of documents and careful integration into the existing codebase while maintaining consistency across the system.

**Scenario 21:**

1. **How can the use case for book categorization be integrated into the existing backlog in Agile?**
   * The use case can be added to the backlog as a new user story, specifying the steps for categorizing books, and prioritized alongside existing tasks for the next sprint.
2. **What changes should be made to the class diagram to incorporate book categories as a new entity?**
   * The class diagram should include a "BookCategory" entity, with a relationship to the "Book" entity. Attributes for categories could include name, description, and potentially tags for classification.
3. **How can the sequence diagram be updated to reflect the new interactions between the user and the system when categorizing books?**
   * The sequence diagram should be updated to show the user selecting categories for a book, and how the system processes the book's categorization, stores the data, and reflects changes in the interface.
4. **How does the Agile process ensure that the development of this new feature is efficiently tracked in upcoming sprints?**
   * In Agile, the new feature can be broken down into manageable tasks, tracked in the backlog, and planned into sprints with clear goals. This ensures focused development and regular progress checks.
5. **What potential issues might arise when adding a new feature like book categorization to an already-running Agile project?**
   * Potential issues include scope creep, delays in the current sprint, or technical debt if the new feature doesn’t align well with existing architecture. Close collaboration, re-prioritization, and planning are necessary to mitigate these risks.

**Scenario 22:**

1. **How can the class diagram be modified to accommodate the new feature of allowing multiple books to be borrowed at once?**
   * The class diagram should be updated to allow multiple book instances to be associated with a single loan entity. A "Loan" entity could have a relationship with multiple "Book" entities.
2. **What changes are needed in the activity diagram to reflect the new workflow of borrowing multiple books?**
   * The activity diagram will need to be updated to show the steps for borrowing multiple books, such as selecting several books, checking them out together, and processing them as a group.
3. **How can the sequence diagram illustrate the interactions between the user, the system, and the books during the borrowing process?**
   * The sequence diagram should reflect the user’s interactions with the system to select multiple books and confirm the loan. The system will then check availability, create a loan record, and update the books’ status.
4. **How does Agile’s iterative development approach allow for the quick adaptation of the system’s architecture to include this new feature?**
   * Agile’s flexibility allows the team to refine the architecture quickly during each sprint, incorporating the new feature into the ongoing development without major delays or disruptions.
5. **How can the team ensure that the addition of this feature does not disrupt the progress of the current sprint?**
   * The team can prioritize the new feature in the sprint planning meeting and break it into small tasks that can be completed within the sprint. This ensures that the feature is integrated without interfering with other ongoing tasks.