

SOFTWARE DEVELOPMENT LIFE CYCLE

2.1 Process Model

Process Model Selection:

Considering the unique objectives and multifaceted requirements of the "Anti-Syndicate Reporting and Alert System" app, the **Scrum process model** stands out as the most appropriate development methodology. This decision is supported by several key factors detailed below:

1. Iterative and Incremental Development:

The app aims to serve various user roles, including Shop Owners, Customers, Security Personnel, Police, and Admins. With features such as secure authentication, reporting illegal activities, price verification, and community interaction, the development process must cater to:

- Delivering small, usable increments of the application during each sprint.
- Incorporating user feedback effectively into subsequent iterations.
- Refining features like heatmaps and incident escalation based on real-world data from early releases.

Scrum allows the team to focus on completing specific modules (e.g., login functions or reporting mechanisms) in iterative cycles, ensuring robust functionality at every step.

2. Flexibility and Adaptability:

The socio-political and market environment targeted by the app is dynamic. Issues such as regulatory updates, changes in user behavior, and evolving societal needs require a development methodology that embraces change. Scrum excels in:

- Prioritizing user stories in each sprint to focus on the most critical needs.
- Adapting backlog priorities based on feedback from early adopters, including authorities and shop owners.
- Accommodating unforeseen changes, such as adding a new report category or updating compliance rules.

3. Continuous Improvement:

To ensure the app remains user-friendly and effective, continuous enhancement is vital. Scrum provides tools for:

- Conducting sprint retrospectives to assess successes, failures, and areas of improvement.
- Regularly revising design elements, such as the interface for reporting incidents or accessing price verification tools, to align with user feedback.

- Encouraging team accountability and innovation by fostering a collaborative, self-organized development environment.

4. Risk Management:

Given the app's critical functionalities, such as incident reporting and evidence collection, risk mitigation is a priority. Scrum minimizes risks by:

- Delivering smaller, testable increments to detect and address vulnerabilities early, such as flaws in data encryption or access control mechanisms.
- Regularly testing components like the reward system or transparency reports to ensure reliability.
- Incorporating user feedback to identify overlooked risks or usability challenges.

Why Scrum Over Other Models?

Although various process models offer unique advantages, none align as effectively with the requirements of the "**Anti-Syndicate Reporting and Alert System**" as Scrum does. This section provides a detailed evaluation of other models and explains why Scrum is the most suitable choice for this project.

1. Waterfall Model:

The Waterfall Model follows a sequential and rigid structure, where each phase must be completed before moving to the next.

- **Drawbacks:**
 - It is inflexible, making it unsuitable for projects with dynamic and evolving requirements like this one, where user needs and market conditions can change rapidly.
 - Early testing and feedback are not integrated into the development process. This limitation is critical, as the app relies on iterative user feedback to refine features like secure login, reporting, and incident heatmaps.
 - Delays in identifying and addressing issues can result in misaligned deliverables, particularly for features dependent on real-time usability.

- **Conclusion:**

The Waterfall Model's linear nature is misaligned with the agile, user-focused, and adaptable approach required for this app.

2. V Model (Verification and Validation Model):

The V Model emphasizes verification and validation through structured testing at every stage of development.

- **Drawbacks:**

- Although it integrates testing, the process remains sequential, which limits the flexibility to adapt to emerging requirements or stakeholder feedback.
- Refinements to user-centric features such as reporting mechanisms or multi-language support cannot be accommodated iteratively, potentially leading to a suboptimal user experience.
- Feedback loops are delayed, hindering the ability to promptly adjust to the app's evolving societal and regulatory needs.

- **Conclusion:**

The V Model is overly focused on validation without offering iterative refinements, making it unsuitable for a project requiring high responsiveness and adaptability.

3. RUP (Rational Unified Process):

RUP is an iterative and incremental framework, offering flexibility similar to agile methods but with significant resource demands.

- **Drawbacks:**

- It is resource-intensive, requiring extensive planning, documentation, and team expertise, which may not be feasible for this project's constraints in terms of time and budget.
- The complexity of RUP can lead to longer onboarding times and potentially slower delivery of functional increments, especially for small teams.
- While it provides adaptability, the overhead of managing its multiple phases can detract from focusing on rapid delivery of high-priority features like real-time incident escalation or transparency reports.

- **Conclusion:**

RUP's heavy resource requirements and complexity make it less practical for this project, which demands a streamlined and efficient process.

4. RAD (Rapid Application Development):

RAD emphasizes speed through rapid prototyping and iterative development, often at the expense of long-term quality.

- **Drawbacks:**

- While it can accelerate the initial delivery of the app, RAD often sacrifices robustness and maintainability. This poses risks for an app handling sensitive data and requiring high security, such as the reporting and evidence-collection features.
- RAD focuses on quick fixes rather than creating a scalable and sustainable product, which is a critical need for the long-term success of this app.
- It does not provide a well-defined framework for managing the iterative user feedback and stakeholder collaboration that this project demands.

- **Conclusion:**

RAD's emphasis on speed over sustainability and security makes it unsuitable for a complex, high-impact application like this one.

5. XP (Extreme Programming):

XP focuses on technical excellence, promoting practices like continuous integration, frequent releases, and test-driven development.

- **Drawbacks:**

- Although XP's principles align with agile values, it lacks a structured framework for project management, which could lead to inconsistencies in the user experience or delivery timelines.
- The project's success heavily depends on team discipline and prior experience with agile methodologies, which might be challenging for less-experienced teams.
- XP is highly developer-centric, often neglecting the broader stakeholder collaboration needed to align with societal and governmental expectations.

- **Conclusion:**

While XP's technical strengths are valuable, its lack of structured project management and emphasis on developer-driven workflows make it less suitable for this project.

6. FDD (Feature-Driven Development):

FDD is focused on delivering well-defined features through incremental development.

- **Drawbacks:**

- While it provides structure, FDD prioritizes feature delivery over iterative refinements and stakeholder collaboration. This limits the app's ability to adapt to dynamic societal and market needs.
- FDD lacks the flexibility to accommodate evolving user feedback, which is essential for a project aiming to serve diverse user groups like shop owners, customers, and law enforcement.
- The method's reliance on upfront planning could hinder the quick integration of new regulatory requirements or user-driven enhancements.

- **Conclusion:**

FDD does not adequately address the need for iterative feedback and collaboration, which are critical for this app's success.

7. DSDM (Dynamic Systems Development Method):

DSDM emphasizes active user involvement and frequent delivery of functional increments.

- **Drawbacks:**

- While it aligns with agile principles, DSDM requires extensive governance and documentation, which can burden teams with limited resources.
- The model's process-heavy nature might slow down the delivery of core functionalities, such as real-time price verification or reward points for reporting violations.
- Its rigid emphasis on formal structures can detract from the lightweight and flexible approach needed for this project.

- **Conclusion:**

DSDM's documentation-heavy framework makes it less practical for a small, agile team working on a resource-constrained project.

Why Scrum Is the Best Choice:

Scrum stands out due to its:

- **Iterative and Incremental Nature:** It allows the app to evolve continuously, integrating user feedback and regulatory changes into each sprint.

- **Flexibility and Responsiveness:** Features can be prioritized and adjusted dynamically, ensuring relevance and user satisfaction.
- **Collaborative Framework:** By involving stakeholders like shop owners, customers, and law enforcement, Scrum ensures the app addresses real-world issues effectively.
- **Lightweight Process:** Its minimal overhead allows the team to focus on delivering value without being bogged down by excessive documentation or rigid phases.

By combining adaptability, collaboration, and iterative improvements, Scrum aligns perfectly with the objectives of the "Anti-Syndicate Reporting and Alert System" and the challenges it seeks to address.

2.2 Project Role Identification and Responsibilities

A successful Scrum project requires clear definition and assignment of roles to ensure smooth communication, accountability, and efficient delivery of increments. Below are the primary roles identified for the development of the project:

1. SCRUM Master:

The Scrum Master plays a critical role in maintaining the structure and principles of Scrum. Their responsibilities include:

- Ensuring the team follows Scrum practices and values throughout the project lifecycle.
- Facilitating Scrum ceremonies like sprint planning, daily stand-ups, sprint reviews, and retrospectives.
- Acting as a liaison between the development team and stakeholders, ensuring smooth communication and addressing roadblocks.
- Monitoring team performance and ensuring adherence to project timelines while fostering a collaborative environment.

2. Product Owner:

The Product Owner represents the stakeholders and ensures the app delivers maximum value by prioritizing the backlog. Their responsibilities include:

- Collaborating with stakeholders (shop owners, customers, law enforcement) to gather and refine requirements.
- Managing and prioritizing the **Product Backlog**, including features like real-time reporting, heatmaps, and multi-language support.

- Defining acceptance criteria for features to ensure they meet user expectations and business goals.
- Reviewing and providing feedback on increments to ensure alignment with objectives.

3. Development Team:

The development team comprises cross-functional members responsible for delivering the project incrementally. Responsibilities include:

- Designing, developing, and testing functionalities such as:
 - Role-specific login systems.
 - Incident reporting and testimony submission features.
 - Interactive heatmaps and transparency reports.
- Collaborating with the Product Owner to break down user stories into tasks and estimating effort for each sprint.
- Regularly demonstrating progress during sprint reviews and implementing feedback in future iterations.

4. Stakeholders:

Stakeholders are external participants whose feedback and involvement shape the project. Key responsibilities:

- Providing critical insights into the challenges faced by shop owners and customers in combating syndicate activities.
- Reviewing app features like price verification and voting systems to ensure they align with societal needs.

5. Administrators:

Administrators play an essential role in ensuring smooth system operations post-deployment. Their responsibilities include:

- Verifying shop compliance and assigning badges to those adhering to government price regulations.
- Generating transparency reports summarizing pricing violations and syndicate activities.
- Managing multi-language settings to ensure accessibility for users across Bangladesh.

6. End Users (Shop Owners, Customers, Security Personnel, Police):

End users interact with the app, contributing reports, feedback, and testimonies. Their roles include:

- **Shop Owners:** Reporting illegal fee collections or extortion cases with evidence.
- **Customers:** Verifying prices, reporting overpricing, and participating in community voting.
- **Security Personnel and Police:** Responding to flagged incidents and taking appropriate action.

This role-based structure ensures that every stakeholder contributes effectively to the project, enabling the development of a robust and impactful solution for addressing syndicate-related issues.