

Dzhanybek Zakiriiiev – Part 1: Software Development Lifecycle & SCRUM

1. SCRUM ROLES & EVENTS

As a student intern on a SCRUM-based development team, I contribute by taking initiative in collaborative environments, adapting quickly to feedback, and supporting the team's sprint goals. My background in software development fundamentals and Agile workflows makes me a dependable contributor.

Contribution to a SCRUM Team:

- Take ownership of assigned tasks or user stories and drive them to completion during each sprint.
- Collaborate in sprint planning and reviews to ensure alignment on scope and technical direction.
- Communicate blockers early and provide backend-focused technical input.
- Write clean, maintainable code and participate in code reviews to raise overall quality.
- Utilize ticketing/project tools like Notion, Jira, and Azure DevOps for sprint tracking and visibility.
- Experience as SCRUM master in academic projects—led planning, facilitated retrospectives, and ensured process adherence.

SCRUM Ceremonies Participation:

- *Daily Stand-up*: Report progress, raise blockers, and coordinate daily objectives.
- *Sprint Planning*: Break down user stories, estimate complexity, and assess feasibility.
- *Sprint Review*: Demo completed features (e.g., auto-flagging logic, reviewer dashboards) and gather feedback.
- *Sprint Retrospective*: Reflect on sprint outcomes, suggest improvements, and foster continuous team growth.

Throughout these ceremonies, I stay open to mentorship, promote collaborative learning, and contribute to a supportive Agile culture.

2. SDLC PLANNING – REDESIGNING THE FLAGGING SYSTEM

I apply SDLC principles confidently, reinforced by coursework in System Analysis and Design. I'm comfortable with various documentation techniques and adapt well to both traditional and object-oriented methodologies.

Requirements Phase:

- Conduct domain exploration to identify entities like *Application*, *Flag*, *Reviewer*, *RuleSet*.
- Analyze stakeholder needs via interviews and existing rule documentation.
- Define functional and non-functional requirements.
- Create user stories from perspectives of reviewers and applicants.
- Establish testable acceptance criteria (e.g., "High-severity flag if no Canadian experience.")

Design Phase:

- Use class diagrams to represent system structure and relationships.
- Create use case diagrams to model interactions like flagging, reviewing, and updating.
- Plan modular architecture with REST APIs separating front-end/backend/services.
- Develop UI wireframes with tools like Figma for reviewer interaction screens.
- Design normalized schema for applications, flags, users, and rule configurations.

Implementation Phase:

- Follow a bottom-up approach: database → services → UI.
- Backend with Node.js + TypeScript for business logic.
- Frontend with Angular + PrimeNG for responsive, user-friendly reviewer dashboards.
- Implement dynamic, rule-based flagging with manual override support.

Testing Phase:

- Unit tests for rule logic and thresholds.
- Integration tests to validate frontend-backend flows.
- UAT simulating reviewer experience and interface usability.
- Include robust error handling (e.g., malformed inputs, incomplete submissions).

Deployment Phase:

- Automate builds for dev, staging, and production.
- Set up environment variables and secure configuration.
- Maintain clear documentation for deployment and rollback procedures.

Maintenance Phase:

- Monitor system logs and performance metrics post-release.
- Iterate on reviewer feedback to adjust rule logic or enhance UI.
- Keep technical/user documentation up to date.
- Plan for future enhancements like ML-driven flag prediction.

Conclusion:

While time constraints limited full Agile adoption, I would ideally apply SCRUM workflows with sprint boards to track progress and iterate effectively. My preferred development flow begins from the domain layer, building a robust backend before layering complex UI. This project simulated a real-world development cycle, and I'm confident in applying industry best practices to deliver impactful improvements to the PRO Portal's flagging system.