

Written Submission: Internship Assessment – Touchstone Institute

Authored by Promise Ononokpono

1. SCRUM Roles & Events

Over the course of my journey in Software I have come to understand the impact of implementing SCRUM in software teams. By following this framework teams are able to work more efficiently together to deliver more value and adapt quickly to changes within the project.

Typically, SCRUM teams consist of three core roles:



1. **Scrum Master**, who facilitates the process and removes obstacles
2. **Product Owner**, who represents the voice of the users and prioritizes the backlog
3. **Developers and QA**, who deliver the product increment during each sprint.

As an intern, I would take on the Developer role, eager to contribute meaningfully while learning from the senior developers and mentors, focusing on writing code, testing features, and collaborating with the team to meet sprint goals. I've already experienced this kind of collaboration in Agile environments during my internships at UNB and Global Vision, where I regularly attended SCRUM ceremonies, contributed updates, and participated in team planning. I also have experience assuming the responsibilities of a scrum master and a product owner informally in previous personal and academic projects. I am always ready for a challenge and seeking growth opportunities!

SCRUM Ceremonies & My Contributions:

1. Daily Stand-Ups: I would provide concise updates on my progress, raise any blockers, and stay aligned with team priorities.

2. Sprint Planning: I would help break down tasks into manageable pieces and contribute estimates where possible, asking clarifying questions.

3. Sprint Reviews: I would demo completed features and receive feedback constructively.

4. Sprint Retrospectives: I would share insights on what helped or hindered my productivity, suggest improvements to the process, and celebrate team wins.

5. Backlog Refinement: I would help review user stories with the Product Owner and ask questions to ensure feasibility and testability.

By being engaged in these ceremonies and being proactive in communication, I would support the team not only by contributing to the codebase but by helping keep things on track, consistent, and user-centered.

2. SDLC Planning: Redesigning the Automatic Flagging System

To redesign the automatic flagging system, I would use a structured approach following the SDLC. This would help ensure we build something reliable, maintainable, and genuinely useful to

end users which is my priority as the developer. With my previous internship and technical experiences, I understand that each step is crucial to the success of the overall project.

Requirements: I would start by understanding the current system's strengths and pain points through stakeholder and user interviews and a review of the existing flagging rules. This would help clarify both the functional expectations such as input structure, flag formats, and rule behavior, and the non-functional requirements like acceptable response time, UI responsiveness, and scalability. I would document all requirements, define use cases, and create clear acceptance criteria for each flagging rule. Based on these findings, we could then decide whether to rebuild from scratch or enhance the existing product. As a final-year student, I've followed this process often in recent projects. It is one of the most exciting parts of building something new, especially for someone who enjoys innovation.

Design: The next step would be to draft the data structure specifications, either in JSON or table format, as this forms the foundation of the system. After that, I would plan the backend service architecture, whether that means designing a new system or updating the existing one. I would also create frontend mockups and diagrams to map out how the new system should function, making sure any gaps from the previous version are addressed. To visualize the user flow and logic, I would sketch out both the frontend (user input) and backend (API payload) processes. This step is highly collaborative and involves working closely with both developers and designers. Personally, I often map out my entire idea on paper before I even start writing code. It helps clarify my thinking and spot issues early.

Implementation: Once the design flow and mapping are complete, the development phase can begin. At this stage, the focus is on writing clean, modular, and well-documented code while using version control to track progress. I approach coding as an incremental process, starting with a proof of concept and then progressively building out features. For the backend, I would use Express with Node and TypeScript, creating reusable functions to implement each rule and return a structured list of flags. On the frontend, I would use AngularJS to build a responsive form that handles user input effectively. This phase also includes setting up the development environment for testing to make sure the system is ready for validation and iteration.

Testing: Next, I would write unit and integration tests, simulate real-world scenarios, and conduct user testing to see if we actually improved the system. For the backend, I would use Jest or Mocha to run unit tests. On the frontend, Karma and Jasmine would be used to test AngularJS components. For overall system validation, end-to-end testing can be done with tools like Selenium to ensure everything works smoothly from user input to visual output. This process would be done in collaboration with the QA team if one is available.

Deployment: I would use a staging environment with Docker for containerization and deploying to the cloud, monitoring post-release performance through API monitoring and error logging. Since I'm familiar with the rest of Touchstone's tech stack this would be AWS-hosted.

Maintenance: Although the system is deployed, it is important to then review performance regularly by scheduling periodic testing and updates for dependencies, incorporate user

feedback, and keep documentation up to date. The documentation would clearly include the flagging logic and API specifications.

By approaching the flagging system redesign through a software development life cycle (SDLC) lens, I can contribute meaningfully at every stage. Whether it's gathering insights, assisting with implementation, or participating in user testing. At its core, software development is a lot like math: you break down a complex problem into smaller parts, carefully solve each one, and then piece everything back together into a working solution.

I'm excited about the opportunity to intern with Touchstone Institute, where I can apply these skills to real-world problems while learning in a collaborative and purpose-driven environment. Thank you for your time and consideration.