Software Configuration Management Plan (SCMP)

Team 3

NewsVerse

Software Engineering
Fall 24

CS 673 A1

BOSTON UNIVERSITY

Damanjit Singh
Abigail Gualda
Stephen Yang
Dewei Wang
Yuhang Wang

Software Configuration Management Plan (SCMP):
The Software Configuration Management Plan will ensure that the development process remains organized, versioned, and traceable. The plan includes the following key elements:

1. **Version Control:**
   - GitHub supports version control through Git which will be used as the version control tool in this project, with all code repositories hosted on it. Each team member will maintain a dedicated branch, and all changes will go through push requests to the main branch for review and approval. Regular commits will be required to track progress.
     1. GitHub Repository: https://github.com/stephenyang0215/CS673_Project
2. **Issue Tracking:**
   - Jira will be utilized to manage tasks, stories, and issues. Each task will be assigned to team members based on roles and capabilities, ensuring accountability for deliverables.

   - Each team member will be responsible for tracking the progression of their assigned tasks from the backlog. They are required to update and estimate the time an assigned task will take to complete.
3. **Build and Deployment Management:**
   - The app will be hosted on AWS EC2, with automated deployment pipelines using GitHub Actions. Regular builds will be triggered after each successful merge to the main branch, with automated tests running to ensure quality.
4. **Documentation Management:**
   - All software documentation (design, requirements, analysis) will be stored in a shared google folder for easy access and collaborative editing. Weekly updates will be documented to reflect the project's progress.
     1. (e.g., Google Drive – CS673 Team 3: NewsVerse WebApp)
5. **Environment Configuration:**
   - Development environments will be standardized with Docker to ensure consistency across team members. Separate environments for development, staging, and production will be maintained on AWS to avoid conflicts.
6. **Backup and Recovery:**
   - Regular backups of data in the database hosted on AWS will be scheduled using system tools to ensure recovery in case of any failure or data loss.

Risk Identification:
Several potential risks could impact the development and deployment of the project:

1. **Technical Debt:**
   - There is a risk that rapid development could lead to insufficient attention to long-term code maintainability, creating technical debt that becomes difficult to resolve later.
2. **APIs Availability and Reliability:**
   - The news aggregation depends heavily on third-party APIs. If any of these services experience downtime, data retrieval could be affected.
3. **Scope Creep:**
   - The plan to incorporate AI semantic analysis as a stretch goal could lead to scope creep if not properly managed, potentially delaying the project timeline.
4. **Team Coordination:**
   - With multiple team members working on different components, there is a risk of miscommunication or integration issues between front-end, back-end, and API functionality.

   - With multiple team members working from different parts of the world it can be difficult to connect with each other and allot the work equally and effectively.
5. **Security Vulnerabilities:**
   - Ensuring data privacy and security, especially with user authentication and external API integrations, presents a potential risk.

Risk Mitigation Strategies:

1. **Addressing Technical Debt:**
   - Implement code reviews and regular refactoring sessions. Ensure that tasks related to code quality are added to the Jira backlog to avoid delays in addressing technical debt.
2. **API Reliability:**
   - Identify alternative APIs as backup options and implement robust error handling to ensure graceful degradation of service if a primary API becomes unavailable.
3. **Managing Scope Creep:**
   - Clearly define the scope of work for the MVP (Minimum Viable Product) and prioritize features. The AI semantic analysis feature will only be pursued if the core functionality is completed on time. Weekly SCRUM sprints will help in managing incremental progress and keeping the scope under control.

4. **Improving Team Coordination:**
   - Weekly check-ins with teammates to ensure workflow is consistent and there is no burnout.
   - Regular daily standups, weekly team meetings, and the use of Jira will ensure that all team members are aligned on tasks and deadlines. Additionally, holding integration testing sessions after each major development sprint will help identify and fix integration issues early.

Conclusion:

Our team is ready and excited to get started on this project. We intend to deliver a high-quality unbiased news platform that meets our targeted audiences eyes and interests. We are committed to producing quality work, hone in on transparent communication, provide ongoing support and by any means get a functioning web app produced.