Software Project Management Plan
Team 3

NewsVerse

Software Engineering
Fall 24

CS 673 A1

BOSTON
UNIVERSITY

Damanjit Singh
Abigail Gualda
Stephen Yang
Dewei Wang
Yuhang Wang

Software Project Management Plan

Project Name: News Verse
Date: September 24, 2024
Team Members: Abigail Gualda, Stephen Yang, Damanjit Singh, Dewei Wang, Yuhang Wang

The SPMP defines each team member's roles and responsibilities, establishing clear expectations and standards. It will also cover the development process and configuration management, ensuring that everyone follows the agreed-upon methodologies, processes, and best practices throughout the project. We will ensure that all team members contribute to the project in a trackable, efficient, and effective manner. This document outlines clearly defined roles and responsibilities, transparent project management tools, and establishes consistent control and oversight of the software development lifecycle.

1. Project Scope

The project involves developing an unbiased news aggregation platform that curates content from multiple sources, ensuring diverse perspectives on current events. The app will aggregate content from various APIs, offering comprehensive updates across a range of categories including national news, business, sports, international relations, weather, stocks, technology, and art. With real-time updates and news feeds, the platform will enhance user engagement by allowing users to stay informed on topics of their interest in an impartial manner. We will potentially be utilizing AI semantic analysis, the platform will filter news based on levels of bias, delivering balanced coverage.

2. Feasibility

Technical Feasibility

- **API Integration**: Aggregating news from multiple APIs like NewsAPI, Google News, and Bing News is technically feasible as these services are well-documented and widely used. Real-time updates can be managed by setting intervals (e.g., scraping every 10 minutes). However, handling API rate limits and ensuring reliability may pose challenges.
- **Front-end and Back-end Development**: Developing a scalable architecture using **React.js** (front-end) and containerization tools like **PM2** for back-end microservices is within reach. Using AWS EC2 ensures cloud scalability. However, ensuring seamless integration between the APIs, front-end, and back-end is crucial.

- **Real-time Updates**: Managing real-time updates with short intervals may lead to high server loads or API costs. Careful monitoring and optimization (caching, error handling, API fallback) are necessary.
- **Search and Filtering**: Building robust search and filter functionality is technically feasible with existing search engines or libraries such as Elasticsearch or Algolia. The complexity lies in optimizing performance as the data volume grows.
- **AI Semantic Analysis**: Implementing AI for bias detection is complex but achievable using existing natural language processing (NLP) models such as those from AWS (Comprehend) or open-source libraries (HuggingFace). Training or fine-tuning these models will require significant resources.

Team and Skill Feasibility

- The team has assigned roles that match their expertise (full stack, back-end, front-end development). With a defined project scope, regular sprints, and task tracking through Jira, managing the development lifecycle should be feasible.
- The complexity of tasks like AI integration may require additional expertise, possibly external AI consultants or team upskilling.

Time Feasibility

- Depending on team capacity, a phased approach to deliver an MVP (Minimum Viable Product) within the project timeline is feasible. Starting with core functionalities like API integration, news aggregation, and basic UI/UX before moving on to more complex features like AI analysis and personalization will help manage scope.
- Weekly sprints and task allocation will be critical in staying on schedule. Scope creep and delays in AI implementation could pose risks, so a clear MVP definition will help mitigate this.

3. Analyzing Issue and Risk Management

Technical Debt: Fast-paced development could lead to code maintainability issues.

- ○ Risk type: Technical risks (Uncertainty around the technology, potential bugs, or system failures.)
- ○ Impact: minor delays except change structure for code.
- ○ Cost: less

- Risk Mitigation Plan: Avoid (Adjusting project plans so the risk cannot occur. For instance, if a technical issue is foreseen, changing to a more stable technology or avoiding overly complex features could help.)
- Solution: Implement code reviews and regular refactoring sessions.
- Owner: Dewei Wang
- Risk prioritization:
    - Likelihood: 8
    - Impact: 3
    - Cost of Managing: 3
    - Priority: 8

API Availability and Reliability: Downtime of third-party APIs could affect news data retrieval.

- Risk type: External risks (Third-party dependencies, API failures, or external data sources becoming unavailable.)
- Impact: halt progress entirely.
- Cost: much
- Risk Mitigation Plan: Avoid
- Solution: Identify alternative APIs and implement error handling
- Owner: Stephen Yang
- Risk prioritization:
    - Likelihood: 2
    - Impact: 10
    - Cost of Managing: 8
    - Priority: 10

Scope Creep: The inclusion of AI semantic analysis as a stretch goal could delay the project.

- Risk type: Technical risks
- Impact: a few day to learn how to use AI analysis
- Cost: less
- Risk Mitigation Plan: Conquer (Taking direct action to eliminate the risk. This could involve early testing, setting up backup services, or improving communication to resolve issues before they arise.)
- Solution: Define clear MVP features and manage scope with weekly SCRUM sprints.
- Owner: Damanjit Singh
- Risk prioritization:
    - Likelihood: 8

        - Impact: 4
        - Cost of Managing: 2
        - Priority: 5

Team Coordination: Miscommunication or integration issues may arise with distributed team members.

- Risk type: Operational risks (Issues related to team performance, timelines, or miscommunication.)
- Impact: often week delays.
- Cost: less
- Risk Mitigation Plan: Conquer and communication
- Solution: Hold regular meetings, daily standups, and integration testing after each sprint.
- Owner: Abigail Gualda
- Risk prioritization:
    - Likelihood: 4
    - Impact: 6
    - Cost of Managing: 6
    - Priority: 6

Security Vulnerabilities: Concerns related to data privacy, security, and external API integration.

- Risk type: Technical risks
- Impact: minor delays.
- Cost: less
- Risk Mitigation Plan: Avoid
- Solution: Secure data storage with external API integration
- Owner: Dewei Wang
- Risk prioritization:
    - Likelihood: 2
    - Impact: 8
    - Cost of Managing: 6
    - Priority: 8

4. **Required Resources:**

Human Resources (Project Team Structure and Responsibilities):

In this project, everyone will contribute to the front and backend development at a certain capacity. However, each role is assigned to a specific team member in confidence that they will uphold those responsibilities.

Project Manager (PM): Abigail Gualda
This role is responsible for overall project control, communication, and coordination. This includes scheduling team meetings and managing Jira stories and tasks. I will oversee user experience design, ensure a smooth application development process, and prepare software documentation, including design plans, requirements, and analysis.

Software Development Lead: Damanjit Singh
This role involves managing the development team to ensure standards and timelines are met. He is responsible for designing and developing the product's architecture based on functional requirements, with focus on the UI/UX design and functionality. This includes creating and refining key components, ensuring a cohesive and user-friendly interface, and integrating different components for seamless functionality. He also oversees code reviews to maintain high development standards across the project.

Quality Assurance (QA) Lead & Configuration Management (CM) Lead: Stephen Yang
This role ensures that the product meets quality and testing standards, manages Git version control, and implements robust APIs and functionalities using Python. He is also responsible for database development and storage, as well as overseeing quality assurance and testing to ensure a reliable and high-quality product.

Development Team: Dewei Wang and Yuhang Wang
The development team is responsible for creating a responsive web interface using React, Tailwind, and Material-UI. This includes ensuring that the interface is user-friendly, visually appealing, and adaptable to different screen sizes and devices, while maintaining consistent design and functionality across the application.

Required Software and Environmental Resources (Project Tools):

Github:
We will be using GitHub as our configuration management as it is known for tracking changes in addition to allowing multiple contributors to work on any project simultaneously. We will also be using the branching strategy, code review process and version control to merge our work together in an effective manner. Since we have multiple developers each one focuses on different aspects of development in this project, GitHub will support the developers to work on their own branch and commit the work to

the main branch.

Testing Strategy:
Unit Testing: Test each component of the application including ec2, database and code. We will use Pytest to test back-end python code and Jest with React Testing Library to test the React framework(TailwindCSS, Material-UI) in isolation to ensure its operation.
Integration Testing: Testing the integration of backend including data pipelines that fetch data through APIs to write to the database and read data from the database to display to the interface.
Performance Testing: Validate the app's performance by recording each test.

Code Reviews:
Developers will be responsible for testing their work first then commit to the feature branch and discuss with QA about the changes and its tests. QA Lead will review the code and test in the isolated environment then commit the change to the development branch. At this point, integration testing will confirm the changes to the module are working fine overall.

VSCode:
We will use VSCode as our primary development environment due to its versatility, extensions, and support for multiple languages. It will facilitate smooth collaboration among the developers.

Figma:
We will use Figma as our design platform to create mockups, wireframes, and prototypes for the user interface (UI) and user experience (UX) components. This will allow us to align design and functionality with the project's goals and provide a collaborative space for design reviews.

Python/JavaScript:
The back-end will be developed using Python, with API integrations and data processing. For the front-end, we will use JavaScript with the React framework, ensuring a dynamic and responsive user interface.

Docker:
We will use docker to containerize the application ensuring it is able to run on any isolated environment. In this project, we will deploy the application on ec2 vm to perform back-end tasks that's how docker comes to play.

Jira:
We will use Jira to track all tasks among team members in addition to aiding the team in Sprint Planning, timeline setting, task breakdown, and prioritization of deliverables. This will allow us to stay organized and work efficiently through the following weeks. We are planning on using Agile methodology with SCRUM adoptions and Jira is the perfect platform to help us document this process.

5. Estimated Cost

Milestone	Lines of Code	Time (hours)
Milestone 1: Project Setup and Architecture Design	300	30
Milestone 2: API Integration for News Aggregation	400	35
Milestone 3: Database and Backend Development	1600	120
Milestone 4: Real-Time News Updates	300	20
Milestone 5: AI Semantic Analysis for Bias Detection	1800	160
Milestone 6: UI/UX Design and Front-End Development	2200	180
Milestone 7: Search and Filter Functionality	1000	80
Milestone 8: Testing, Deployment, and MVP Release	800	60

This plan spreads the workload across our available 3 months, giving enough work per week to distribute the load amongst the 5 team members, which is feasible if the team collaborates efficiently. Roughly, each developer will spend 11-12 hours a week working on the project and it is feasible for the team.

6. Scheduling Process

Development Methodology: Agile SCRUM

This method works for us because Agile SCRUM focuses on flexibility, transparency and continuous improvement which allows us to manage and deliver complex works for the project efficiently. Within each weekly sprint, we can work together to follow up, discuss progress, difficulties and upcoming tasks.

Project Schedule:

Milestone 1: Project Setup and Architecture Design

- Define the platform's architecture, including front-end, back-end, and API integration plans. Set up GitHub, Jira, and cloud environments (AWS EC2 and Docker).
- Sprint 1: Project discussion, scope definition, prepare documentation. Setup environment, GitHub repository and branching.

Milestone 2: API Integration for News Aggregation

- Integrate third-party APIs (e.g., NewsAPI, Google News API) to aggregate news across categories like national, business, sports, etc. Ensure seamless data flow from APIs into the platform.
- Sprint 1: Develop the data pipelines and database in the cloud environment.

Milestone 3: First Passes at UI/UX Design and Front-End Development

- Run a design sprint with the team where we create a low fidelity wireframe on Figma. Discuss with the team what this will mean for the usability of the web application as well as thinking through how we plan on connecting the backends logic. This conversation will have to include how we plan on integrating our API usage.
- Sprint 1: Brainstorm the front-end design and collect requirements.
- Sprint 2: Integrate API usage into UI/UX design and present functionality of back-end.

Milestone 4: Database and Backend Development

- Implement the database to store and manage news articles. Develop the backend to handle API requests, manage updates, and process news data.
- Sprint 1: Design the workflow of batch processing and develop the operations of the database tables for connecting to the front-end.

Milestone 5: Real-Time News Updates

- Build the infrastructure to fetch and display real-time news updates from multiple sources at regular intervals (e.g., scraping APIs every 10 minutes).
- Sprint 1: Spin up the infrastructure to run the application.
- Sprint 2: Test the performance to ensure it is aligned with non-functional requirements.

Milestone 6: AI Semantic Analysis for Bias Detection

- Integrate AI models to analyze and filter news articles based on bias or impartiality, allowing users to view balanced perspectives.
- Sprint 1: Integrate ChatGPT API into the back-end and test the summary quality.
- Sprint 2: Developing the pipelines triggering the ChatGPT call once the news data are in place.

Milestone 7: Final UI/UX Design and Front-End Development

- Develop a responsive, user-friendly interface using React.js, TailwindCSS, and Material-UI. Focus on clear categorization, multi-source comparisons, and a customizable news feed.
- Sprint 1: Develop front-end interface and test the development and discuss the quality of the application.

Milestone 8: Search and Filter Functionality

- Implement robust search capabilities with filters based on topics, categories, and sources. Enable users to compare articles from multiple sources side-by-side.
- Sprint 1: Develop the functions including searching articles and filtering. Then test and review the quality of the functions.

Milestone 9: Testing, Deployment, and MVP Release

- Conduct thorough testing (QA, security, usability). Deploy the platform on AWS EC2 with GitHub Actions for automated builds. Launch the MVP with core features, and incorporate feedback for future iterations.
- Sprint 1: Implement quality assurance, code review, testing on the application.
- Sprint 2: Deploy the application on the AWS environment, launch the product and review the feedback.

Workflows:

The workflow we will be adopting in this project covers traditional Agile SCRUM and Sprint methodologies. We will be using weekly Sprint Planning, Backlog Grooming, Weekly Standups, the following workflow stages: “To Do,” “In Progress,” “Review,” “Blocked,” and “Done.” As a team we will review the backlog during our weekly team meetings, discuss which tasks are of higher priority, break down large tasks into smaller subtasks, assign actionable items, and team members will be responsible for keeping track of their work completion.

Meeting Cadence:
In class on Tuesdays at 6:00pm we will participate in a daily stand-up to check in with one another and ensure we are all on the same page. As a team we will be holding and attending weekly team meetings on Wednesdays at 6:00pm - 7:15pm. During this meeting we will do a sprint retrospective of reviewing the current backlog, discussing as a team what tasks we would like to prioritize, assign tasks, and discuss our priorities for the week.

Tools:
Slack – https://cs673a1.slack.com/archives/C07L83B3249
Email (if needed)

7. Documentation and Monitoring:

We will be using a combination of tools to ensure effective collaboration and transparency.

Jira will serve as our primary platform for tracking workflow and task assignments, accessible at Jira link. The team will update the backlog weekly, review completed tasks, assign new tasks, and monitor overall progress, ensuring that everyone is aligned on project goals.

We will maintain open communication through Slack (Slack link), providing a constant line of communication and promoting transparency.

For document sharing and storage, we will use Google Drive, where all essential project documents will be stored, including weekly team meeting SCRUM Notes and/meeting minutes. This will ensure that detailed records of discussions, decisions, and team expectations are available for reference throughout the project.

Google Drive | SCRUM Notes

8. 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.