METCS673 Group 6 - Software Configuration Management Plan

TerrierMap

Team Members:

- Ryan Christopher
- Misael Gared
- Jasmine Hughes
- Siddhraj Parmar

Configuration Items:

- Code (Production and Test):
  The code for this project will be stored and maintained using Git and structured within a dedicated Git repository. Code management practices such as version control and branching strategies will be followed. Deployment would be done on either Heroku or Vercel. Automated testing will be integrated into the development workflow.

- Specification Documentation:
  Project-related documents including requirements, design blueprints, development plans, and user manuals will be organized in a centralized Google Drive folder. Changes to documents will be clearly logged to ensure transparency in updates. Resources like user guides and developer instructions will be included to facilitate smooth onboarding and usage.

- User Documentation:
  User documentation will be stored on the GitHub repository as a README file. It will include the current version of the application, lists of ongoing changes, and any other relevant information for the user.

- Supporting Software:
  All supporting software will be documented in the user documentation. It will include the software versions the application currently supports.

Source Code Version Control:

- Git will be used as the primary version control system for managing the project's codebase.
- We will follow a structured branching strategy where features, bug-fixes, and release branches are created as needed.
- All commits will adhere to a standardized format clearly describing changes and associating them with their respective task(s).
- Every change must be submitted via a pull request (PR). The PR will undergo thorough code review by at least one reviewer with feedback provided if necessary.
- The main branch will be protected and no direct pushes will be allowed to it. All updates to this branch will occur through reviewed and approved pull requests.
- The repository owner will be responsible for maintaining repository access permissions and will oversee the code review and merge process.

Change Management:

- Any changes to the project will go through an approval process. This requires submitting a request, undergoing a review, and then getting approved or denied.
- Any change that has successfully gone through the approval process will be documented into a revised plan and timeline.

Progress Tracking:

- GitHub Issues will be utilized to track to-do's, assign tasks for bug fixes/improvements, and track milestones.

Build and Release Management:

- Specific milestones will be set for each build that is released. Each build will undergo testing before being approved by the Project Manager.

Audits and Reviews:

- As new builds are released, a review and an audit will be done on the software configuration items (SCIs) to ensure and maintain SCMP compliance.

Tools and Resources:

- Git: Source code management
- GitHub: Tracking progress on tasks and issues
- Figma: Wireframes, UX/UI, and branding
- Whatsapp: Team communication

Risk Management:

- Development Delays: This can occur due to various circumstances from technical issues to personal matters. To avoid such risks, thorough research into technologies will be completed throughout the development process. To account for any additional delays, overestimating the time it takes to complete tasks will be implemented into the timeline.
- Feature Creep: Additional features might get added during the development process that can lead to delays or an unfinished product. To minimize this risk, features will be added iteratively until completion to ensure a complete project.
- Data Loss: Computer crashes, codebase errors, and other unfortunate accidents occur often enough that data loss is a concern. To minimize lost data, regular backups of code and data will be implemented.
- Lack of Clarity: Unclear expectations from stakeholders or team leads is often an issue. The project manager setting clear expectations for everyone to accomplish or work on during the week will mitigate this risk.

Conclusion:

We are greatly looking forward to building TerrierMap for the BU community. It will allow us to solve an issue that was present in each of our individual academic experiences at BU. It could serve as a useful tool for faculty, staff, and students getting around our unique campus. Our group will focus on clear team communication and thoughtful development in order to deliver an engaging tool to navigate BU.