Deliverable 3: Version Control Policy

We decided to use Gitlab for version control. We will organize our repository into a private project where we can manage access and permissions (team members will all have master access). We can confidently and rapidly iterate towards a successful project with this next-generation developer collaboration software.

Consolidating source code: We will be able to wrangle our project into one easy to manage tool. We can control read/write permissions to specific branches which are dedicated to specific task individually. We will keep our documentation within the project using Gitlab's built-in wiki system. We can collect and share reusable codes and use powerful APIs.

Tracking and managing source code: We can leverage GitLab's built-in issue tracker to outline and share ideas. We can view a simple dashboard with a list of to-do's so we know what needs our attention. We can create templates for issues and merge requests. We can integrate with external services like Slack, JIRA etc. We can plan, organize and visualize our last version with issue board (Kanban-style). We can organize issues and merge requests into groups. We will assign due dates to issues to make sure things get done on time. We can also categorize and track issues or merge requests based on descriptive titles. We will be able to spend less time searching and more time building software with powerful search.

Increasing code quality: We can create merge requests and mention team members to review and safely merge our changes. We will be able to view and compare merge request diffs and resolve merge conflicts in the UI. We can leave comments and resolve discussions on specific lines of code. We can test every merge requests, run multiple jobs in parallel and see if build passes with GitLab's built-in CI. We can use the cherry pick button in the UI to isolate specific changes from a merge request. We can increase code-quality in that way.

Following changes and progress: We can view a list of the latest commits, merges, comments and team members on our project. We can set our alert preferences and be notified by email when there are changes to an issue or merge request. Using code analytics, we can check how long it takes to complete our software development lifecycle.

Official backup & naming conventions: We will use USB drive as off-line backup. We decided to use naming convention with version number "M2_ver2.doc", date-time stamp seemed a bit verbose.