# Project Report

**Team Members:**

* Phenix Gikenyi—Team Lead
* Ryan Munga—SCM lead
* Talak Patel—Lead Dev & QA tester

**Objective:**

To create a simple web page and practice configuration management using Git GUI and TortoiseGit for version control and collaboration.

# Version Control

* + Commit and Push Changes
  + Branching and Merging.
  + Conﬂict Resolution

# Code Review & Software Ǫuality Assurance

* + Pull Requests (PRs)
  + Revert Changes

# Tracking Releases

* + Tagging for Releases
  + Change History and Log

# Backup & Recovery

* + Backup
  + Recovering Old Versions

We did some research on SCM as it relates to our project

Methodologies and Website Development: A Survey of Practice

Taylor, McWilliam, Forsyth, and Wade (2002) conduct a detailed examination of the methodologies applied in website development through an extensive survey of industry practices. The study reveals a notable reliance on informal approaches instead of formalized methodologies. This trend is attributed to the fast-paced nature of the web development industry, which often prioritizes rapid time-to-market over structured processes. The authors highlight that the urgency for swift deployment frequently overshadows the adoption of structured development methodologies (Taylor et al., 2002, pp. 381-391).

The research also underscores the importance of agility in responding to client demands while cautioning against the potential downsides of excessive flexibility. Agile practices are appreciated for their ability to provide speed and adaptability; however, the lack of formal documentation and structured guidelines can lead to inconsistencies and quality issues. The authors argue that while agile methodologies address the need for rapid development, they sometimes fall short in maintaining uniformity across the development lifecycle (Taylor et al., 2002, pp. 381-391).

In the context of programming using website development tools, it was found that "a variety of website development tools were in use across the 25 organizations researched" (Taylor et al., 2002, pp. 381-391). This diverse use of tools underscores the industry's preference for flexible yet efficient development environments.

Additionally, the study emphasizes the necessity of comprehensive website testing, noting that "website testing will typically include testing individual website pages, the links between these pages, and how inputs such as forms behave as well as navigation between pages" (Taylor et al., 2002, pp. 381-391). This thorough approach to testing is crucial for ensuring the functionality and user experience of web applications.

To mitigate the challenges associated with flexibility and lack of structure, the authors propose a hybrid approach that blends formal methods with agile principles. This hybrid methodology is designed to strike a balance between the structured, documented approach of traditional methods and the flexible, adaptive nature of agile practices. The goal is to ensure efficiency and high standards in web development projects. The proposed approach emphasizes maintaining thorough documentation and structured guidelines while allowing for the rapid iteration and adaptability that agile practices offer (Taylor et al., 2002, pp. 381-391).

In their conclusion, Taylor et al. (2002) advocate for the adoption of methodologies tailored to the specific needs of each project and organization.

What Signal Are You Sending? How Website Quality Influences Perceptions of Product Quality and Purchase Intentions

Wells, Valacich, and Hess (2011) analyze how website quality impacts consumer perceptions of product quality and their purchase intentions. They argue that website design serves as a critical signal of a company’s credibility and product value.

The study identifies key elements of website quality, including visual appeal, usability, and technical reliability. It highlights that even small design flaws can lead to negative perceptions. For instance, they note that the "current technological capabilities of eCommerce limit sellers' ability to convey intrinsic product attributes like taste, smell, touch, and fit" (Grewal et al., 2004). This limitation underscores the need for high-quality website design to compensate for the lack of physical product interaction.

Moreover, the study emphasizes that consumers often encounter more unknown retailers online, increasing the need for sellers to differentiate themselves and address consumers' heightened perceptions of risk. The authors write that "sellers using an eCommerce marketing channel must leverage informational cues or signals to facilitate a consumer's ability to make accurate quality assessments" (Pavlou et al., 2007). This approach is essential for building consumer trust and encouraging purchase intentions, especially for products with experiential attributes offered by unknown retailers.

The Common Configuration Scoring System (CCSS): Metrics for Software Security Configuration Vulnerabilities

Scarfone and Mell (2010) introduce the Common Configuration Scoring System (CCSS), a standardized framework for assessing the severity of software configuration vulnerabilities. Designed to complement the Common Vulnerability Scoring System (CVSS), CCSS provides a more detailed evaluation of configuration-specific risks. The CCSS framework comprises three metric groups: base, temporal, and environmental.

A key aspect of the CCSS is that it encourages analysts to "consider the security implications of each option" when a configuration issue arises (Scarfone & Mell, 2010). For example, analysts should evaluate the potential impact of enabling or disabling a setting. Additionally, the authors highlight the importance of understanding the broader security implications of configuration settings. They suggest that in scenarios with large numbers of settings, such as file privileges for users, analysts should "consider the common cases independently" rather than attempting to evaluate all possible combinations (Scarfone & Mell, 2010). This approach ensures that the most likely and impactful security risks are addressed effectively.

# Implementing Git GUI and TortoiseGit

**Project Scope:** The tools should be used by all developers for all the entire project, once Conﬁguration Items have been identiﬁed

# Timeline

1. Install and conﬁgure tools according to local environments
2. Agree on member roles
3. Deﬁne standards & best practices—establish protocols for coordination and conﬂict resolution
4. Training—take time to get familiar with tools
5. Implement the strategy & assess the result
6. Develop the codebase
7. Return to step 3 or 4 as needed based on the assessment of the implementation

**Roles & Responsibilities**

**Team Lead** — coordination of the entire team, ensures deadlines are met

**SCM Lead —** sets standards and manages repositories

**ǪA Lead —** sets quality assurance standards

**Technologies Used**

1. HTML
2. css
3. Javascript

**Tools Used:**

* **Git GUI:** Used for visual management of repository actions (commits, pushes, pulls, etc.).
* **TortoiseGit:** Used for repository management directly from Windows Explorer.

**Branching Strategy:**

* **Main Branch:** Contains stable, production-ready code.
* **Feature Branches:** Separate branches for new features or updates (e.g., feature/add-navbar, feature/update-styles).

**Commit Guidelines:**

* Use descriptive commit messages
* Commit frequently to capture progress.

**Collaboration:**

* Pull the latest changes before starting work.
* Resolve conflicts as necessary using Git GUI or TortoiseGit's merge tools.

**Key Activities**

1. **Repository Initialization:**
   * Created a Git repository for the project and uploaded initial files.
2. **Feature Development:**
   * Developed features on separate branches, such as a navigation bar and footer.
3. **Merging and Conflict Resolution:**
   * Merged feature branches into the main branch after successful testing.
   * Resolved conflicts using TortoiseGit's visual conflict resolution tools.
4. **Version Control:**
   * Used Git GUI to track changes, commit updates, and visualize the repository history.