**VCS Research Report**

**Introduction**

In the development of the MusoPlan project at Globex Corporation, a version control system (VCS) is a crucial component to ensure collaborative and organized software development. The choice of VCS can significantly impact the project's efficiency and maintainability. After thorough investigation, the following VCS options were considered: Git, Mercurial, and Subversion.

**Different VCS Investigated**

**Git**

Git is a distributed version control system known for its speed, flexibility, and branching capabilities. It has become an industry standard and is widely adopted for various software projects.

**Mercurial**

Mercurial is another distributed version control system, offering a user-friendly interface and simplicity in use. It emphasizes ease of use and is suitable for projects of all sizes.

**Subversion (SVN)**

Subversion is a centralized version control system that tracks changes to files and directories. It is well-suited for projects where a central repository is preferred.

**Benefits of Different VCS**

* **Git:** Git excels in its branching and merging capabilities, making it ideal for parallel development. Its decentralized nature allows for offline work and faster operations.
* **Mercurial:** Mercurial offers an easy-to-use interface, making it suitable for teams with varying levels of expertise. It provides a straightforward workflow and is well-documented.
* **Subversion:** Subversion's centralized model simplifies access control and permissions. It is a solid choice for projects with a clear hierarchy and centralized management.

**How Git Fits with Organisational Requirements**

After careful consideration, Git was chosen as the version control system for the MusoPlan project at Globex Corporation. The decision is based on the following factors:

1. **Decentralized Development:** Git's distributed nature aligns with the collaborative nature of the development team, allowing developers to work independently and merge changes seamlessly.
2. **Branching Model:** Git's robust branching model facilitates parallel development, enabling the team to work on multiple features simultaneously without conflicts.
3. **Industry Standard:** Git is widely adopted in the industry, ensuring that team members, even those new to the project, are likely to be familiar with its workflow.

**Installation Process**

The installation of Git was straightforward, adhering to Globex's organizational requirements. The process involved downloading the Git installer from the official website, running the executable, and following the guided installation steps. No issues were encountered during the installation.

**Pre-Installation Factors**

Before installation, the system's compatibility and the availability of necessary permissions were verified. Additionally, it was ensured that the selected Git hosting service complies with organizational security standards.

**Disruptions Installation May Cause**

Git's installation did not cause any disruptions to ongoing work. Since Git operates locally, team members could continue working on their tasks without interruption during the installation process.

**Configuration Process**

Following installation, the Git configuration included setting the global user.name and user.email to ensure proper attribution of commits. This was done using the **git config** command. Each team member will configure these settings on their local machines to maintain a clear record of contributions.

**Conclusion**

In conclusion, Git was selected as the version control system for the MusoPlan project due to its decentralized nature, robust branching model, and industry-wide acceptance. The installation and configuration processes were smooth, adhering to organizational requirements. This decision lays a solid foundation for collaborative and efficient development throughout the project's lifecycle.