

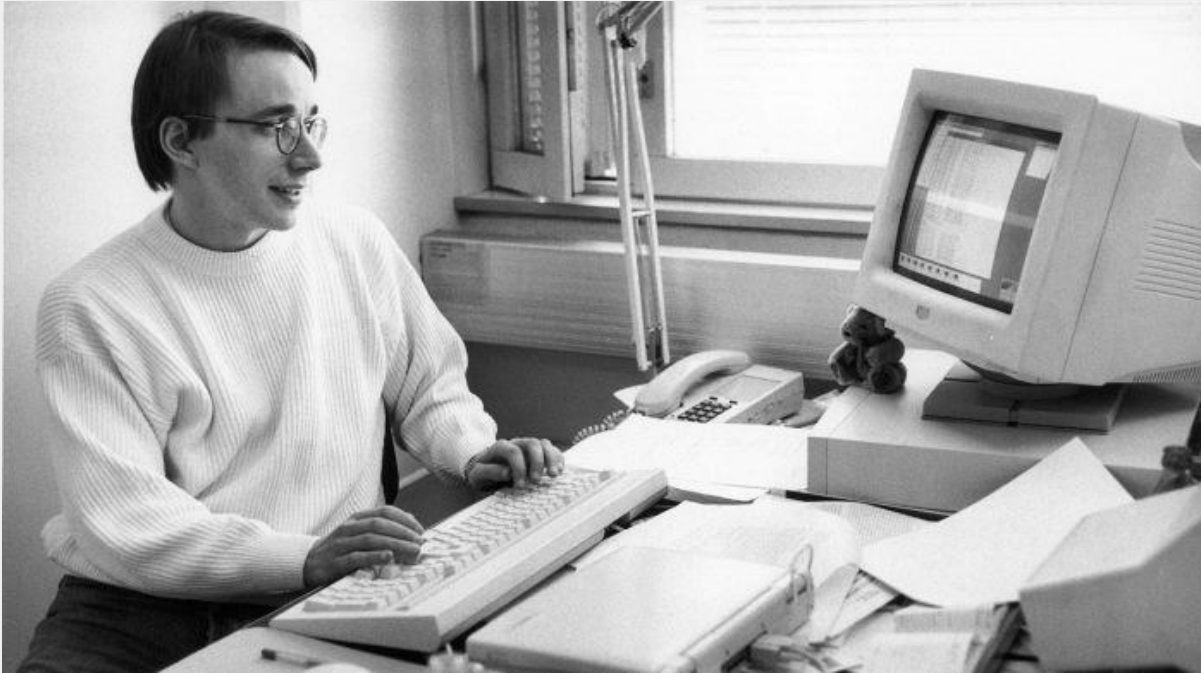
IoT CPP





Understanding Git and GitHub

Mehdi Bigdeli

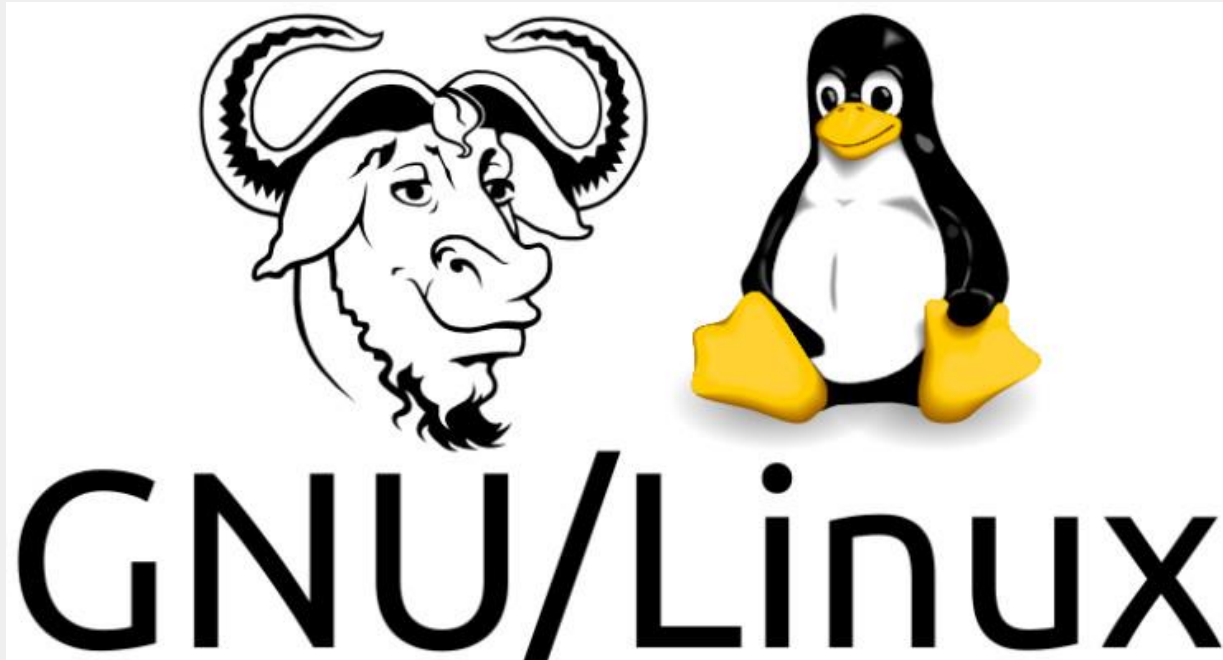


1991

GNU OPERATION SYSTEM

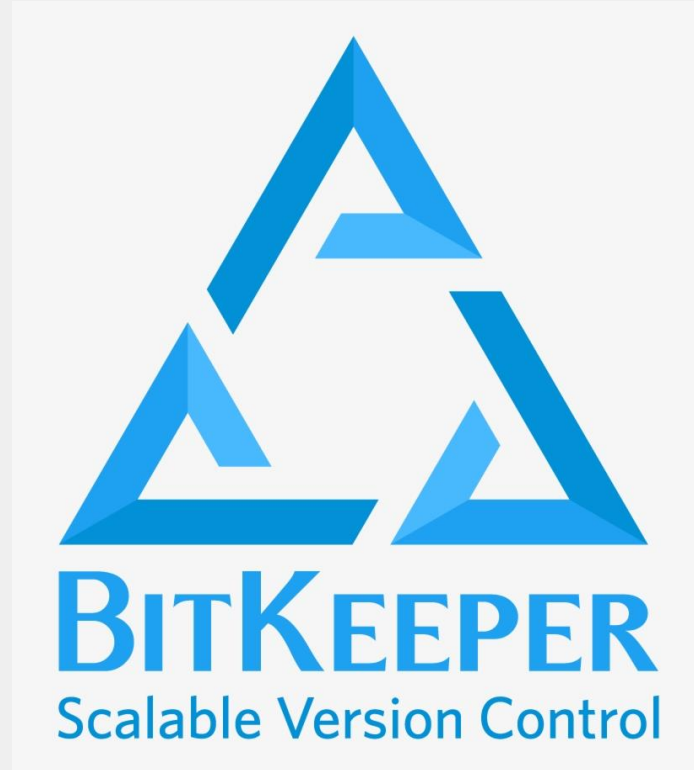


Founder of the GNU : Richard Stallman





Linus Torvalds



Birth of Git



December 2005, Git 1.0 was released.







VERSION CONTROL SYSTEM





Introduction to Version Control Systems

- **Definition and Importance of Version Control:**
 - **Version control systems (VCS) track changes to files over time.**
 - **Crucial for collaboration, tracking history, and reverting changes.**
 - **Examples: CVS, Subversion, Git.**

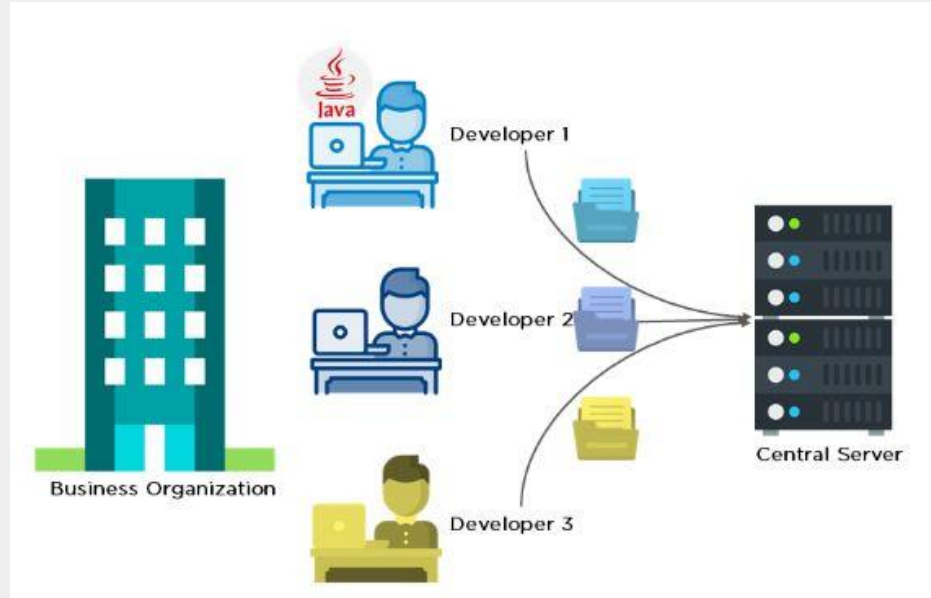


Introduction to Version Control Systems

- Evolution to Distributed VCS:
 - **Traditional (Centralized):** Single repository on a server.
 - **Distributed (Git):** Full copy of repository locally, enables offline work.

Introduction to Version Control Systems

- Evolution to Distributed VCS:
 - Traditional (Centralized):



Introduction to Version Control Systems

- Evolution to Distributed VCS:
 - Distributed (Git):





git





Understanding Git

- **What is Git?**
 - **Distributed version control system developed by Linus Torvalds in 2005 to develop Linux Kernel**
 - **Designed for speed, data integrity, and support for non-linear workflows.**



Understanding Git

- **Basic Concepts:**
 - **Repository**
 - **Commit**
 - **Branch**
 - **Merge**



Understanding Git

- **Git Workflow:**
 - **Initialize a repository: 'git init'**
 - **Add and commit changes: 'git add', 'git commit'**
 - **Branching and merging: 'git branch', 'git merge'**



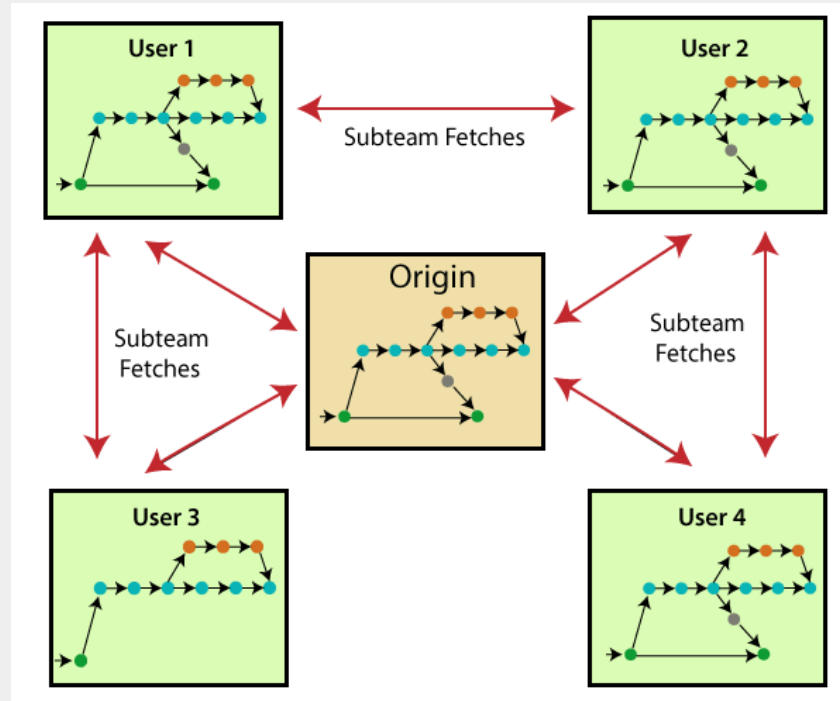


Features of Git

- **Open Source**
- **Scalable**
- **Distributed**
- **Security**
- **Speed**
- **Supports non-linear development**
- **Branching and Merging**
- **Data Assurance**
- **Staging Area**

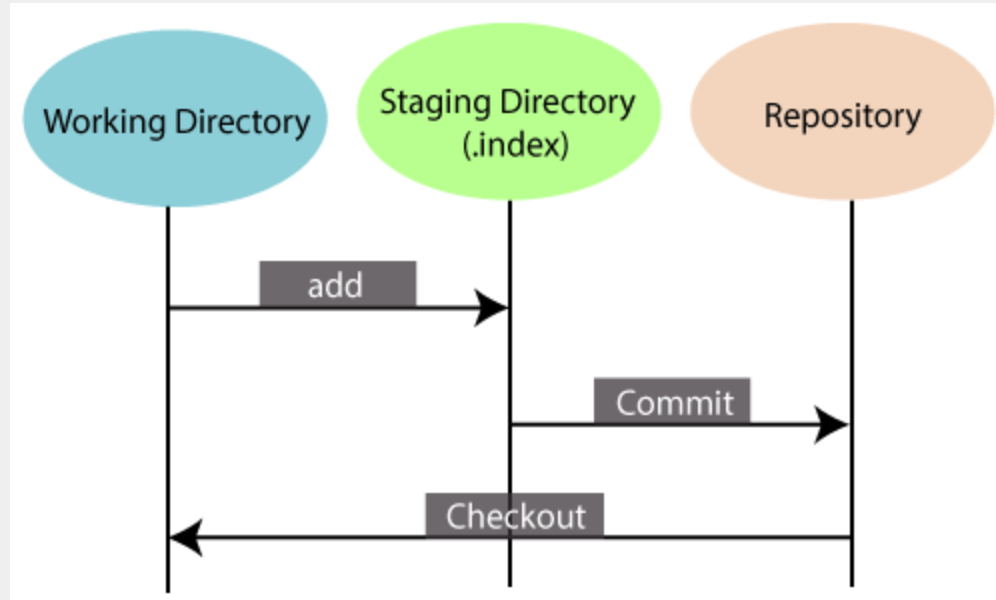
Features of Git

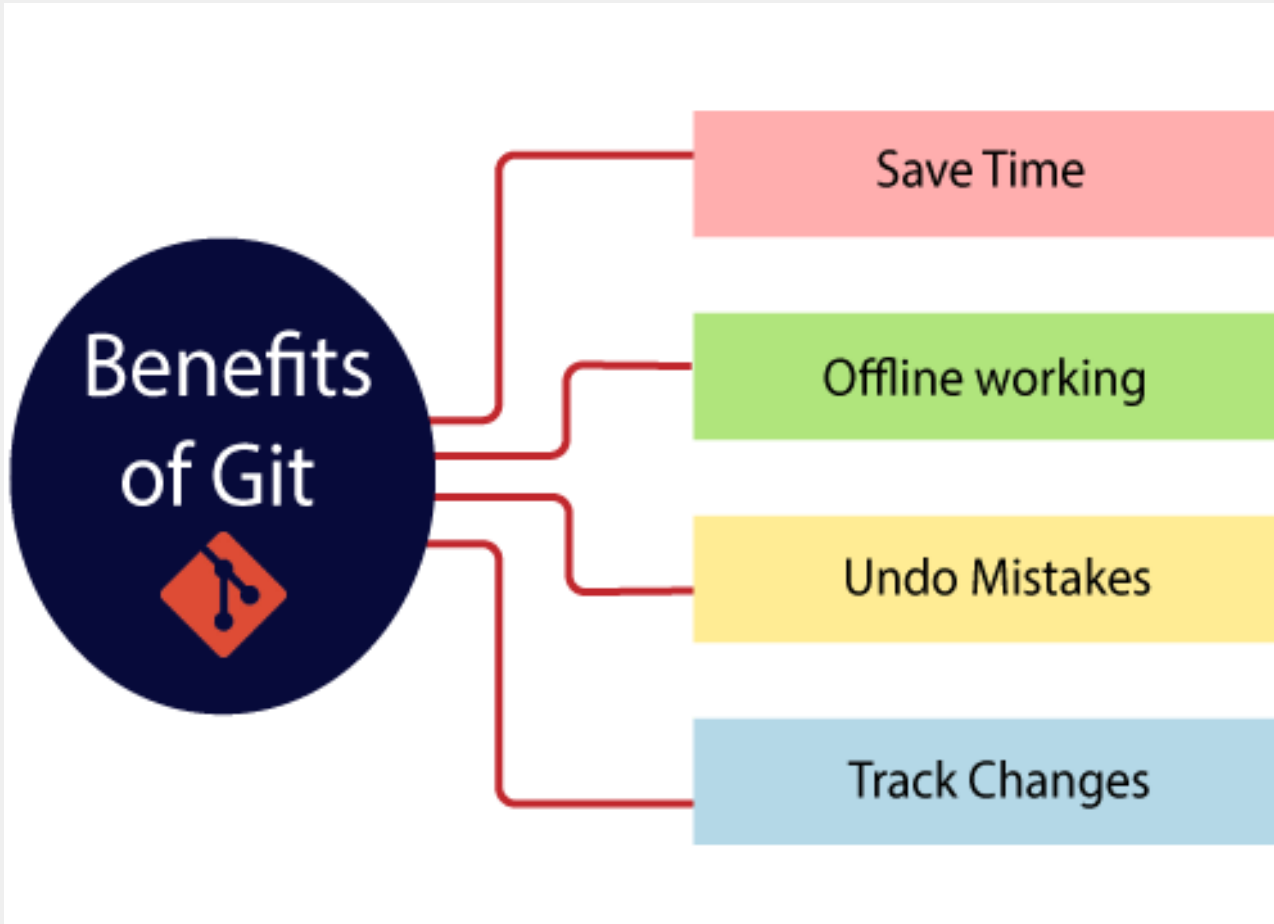
- **Distributed**



Features of Git

- **Staging Area**



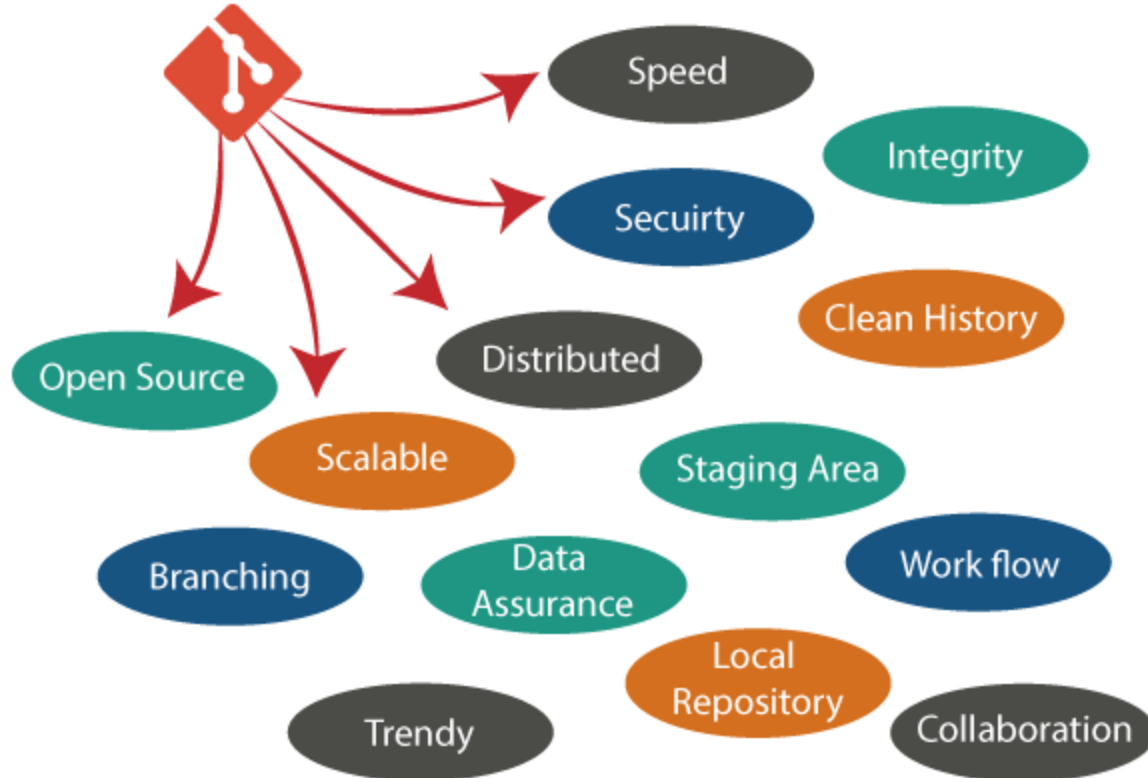




Benefits of Git

- **Benefits for Developers and Teams:**
 - **Concurrent work without conflicts.**
 - **Traceability and accountability with commit history.**
 - **Facilitates collaboration and code review.**

Why Git?





Why Git?

- **Git Integrity**
- **Trendy Version Control System**
- **Everything is Local**
- **Collaborate to Public Projects**



Introduction to GitHub

- What is GitHub?
 - **Web-based platform for hosting Git repositories.**
 - **Adds features like collaboration, code review, and project management.**
- Role in Collaboration and Sharing:
 - **Enables distributed teams to work together.**
 - **Facilitates pull requests for code review and contribution.**



Key Differences Between Git and GitHub

- **Git vs. GitHub: Understanding the Distinction:**
 - **Git: Software for version control, manages repositories locally.**
 - **GitHub: Web-based hosting service for Git repositories, adds collaboration tools.**
- **Git as a Version Control System (VCS):**
 - **Manages source code history and versions locally.**
 - **Supports basic operations like committing, branching, and merging.**



Key Differences Between Git and GitHub

- **GitHub as a Git Repository Hosting Service:**
 - **Provides cloud storage and backup for repositories.**
 - **Adds collaboration features like issues, pull requests, and project management.**



BEST git PRACTICES TO FOLLOW IN TEAMS



Best Practices and Tips

- **Git Best Practices:**
 - **Commit frequently with clear messages.**
 - **Use branches for features and fixes.**
 - **Don't Commit Half-Done Work**
 - **Test Your Code Before You Commit**



Thanks for your attention.