

Introduction to Version Control with Git [1]

Khaled El-Wazan ¹

Department of Mathematics and Computer Science, Faculty of Science,
Alexandria University

¹khaled_elwazan@alex-sci.edu.eg

Outlines

Background

- ① A wise man strategies.
- ② Text and code projects need back up plans.
- ③ The main issues with projects.
- ④ Version control to the rescue.

Background

- ① A wise man strategies.
- ② Text and code projects need back up plans.
- ③ The main issues with projects.
- ④ Version control to the rescue.
- ⑤ Develop and maintain a repository of content, provide access to historical editions of each datum, and record all changes in a log.

Background

- ① A wise man strategies.
- ② Text and code projects need back up plans.
- ③ The main issues with projects.
- ④ Version control to the rescue.
- ⑤ Develop and maintain a repository of content, provide access to historical editions of each datum, and record all changes in a log.

Definition

A tool that manages and tracks different versions of software or other content is referred to generically as a version control system (VCS).

Histroy

- Discords between tools and projects.
- The temptation of creating tools on the fly.
- Git, the information manager from hell.
- The genesis of Git.
- BitKeeper, free as in beer.

The Elusive Features

- Facilitate Distributed Development.

The Elusive Features

- Facilitate Distributed Development.
- Scale to Handle Thousands of Developers.

The Elusive Features

- Facilitate Distributed Development.
- Scale to Handle Thousands of Developers.
- Perform Quickly and Efficiently.

The Elusive Features

- Facilitate Distributed Development.
- Scale to Handle Thousands of Developers.
- Perform Quickly and Efficiently.
- Maintain Integrity and Trust.

The Elusive Features

- Facilitate Distributed Development.
- Scale to Handle Thousands of Developers.
- Perform Quickly and Efficiently.
- Maintain Integrity and Trust.
- Enforce Accountability.

The Elusive Features

- Facilitate Distributed Development.
- Scale to Handle Thousands of Developers.
- Perform Quickly and Efficiently.
- Maintain Integrity and Trust.
- Enforce Accountability.
- Immutability.

The Elusive Features

- Facilitate Distributed Development.
- Scale to Handle Thousands of Developers.
- Perform Quickly and Efficiently.
- Maintain Integrity and Trust.
- Enforce Accountability.
- Immutability.
- Atomic Transactions.

The Elusive Features

- Facilitate Distributed Development.
- Scale to Handle Thousands of Developers.
- Perform Quickly and Efficiently.
- Maintain Integrity and Trust.
- Enforce Accountability.
- Immutability.
- Atomic Transactions.
- Support and Encourage Branched Development.

The Elusive Features

- Facilitate Distributed Development.
- Scale to Handle Thousands of Developers.
- Perform Quickly and Efficiently.
- Maintain Integrity and Trust.
- Enforce Accountability.
- Immutability.
- Atomic Transactions.
- Support and Encourage Branched Development.
- Complete Repositories.

The Elusive Features

- Facilitate Distributed Development.
- Scale to Handle Thousands of Developers.
- Perform Quickly and Efficiently.
- Maintain Integrity and Trust.
- Enforce Accountability.
- Immutability.
- Atomic Transactions.
- Support and Encourage Branched Development.
- Complete Repositories.
- A Clean Internal Design.

The Elusive Features

- Facilitate Distributed Development.
- Scale to Handle Thousands of Developers.
- Perform Quickly and Efficiently.
- Maintain Integrity and Trust.
- Enforce Accountability.
- Immutability.
- Atomic Transactions.
- Support and Encourage Branched Development.
- Complete Repositories.
- A Clean Internal Design.
- Be Free, as in Freedom.

The First Commit I

```
-  
commit e83c5163316f89bfbde7d9ab23ca2e25604af29  
Author: Linus Torvalds <torvalds@ppc970.osdl.org>  
Date: Thu Apr 7 15:13:13 2005 -0700
```

Initial revision of "git", the information manager from hell

The First Commit II

```
commit 1da177e4c3f41524e886b7f1b8a0c1fc7321cac2
Author: Linus Torvalds <torvalds@ppc970.osdl.org>
Date:   Sat Apr 16 15:20:36 2005 -0700
```

Linux-2.6.12-rc2

Initial git repository build. I'm not bothering with the full history, even though we have it. We can create a separate "historical" git archive of that later if we want to, and in the meantime it's about 3.2GB when imported into git - space that would just make the early git days unnecessarily complicated, when we don't have a lot of good infrastructure for it.

Let it rip!

References I

- [1] J. Loeliger and M. McCullough, *Version Control with Git: Powerful tools and techniques for collaborative software development.* " O'Reilly Media, Inc.", 2012.