Introduction to Version Control with Git [1]

Khaled Fl-Wazan 1

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



¹khaled_elwazan@alex-sci.edu.eg

Outlines

- Background
- 2 History
- Main and Essential Features
- The First of Many
- The Bibliography

1: Background

Background

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

Background

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

Background

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

2: History

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.

3: Main and Essential Features

The Elusive Features

• Facilitate Distributed Development.

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

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

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

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

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

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

- 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.

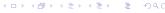
- Facilitate Distributed Development.
- Scale to Handle Thousands of Developers.
- Perform Quickly and Efficiently.
- Maintain Integrity and Trust.
- Enforce Accountability.
- Immutability.
- Atomic Transactions.
- Support and Encourage Branched Development.
- Complete Repositories.



- 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.



- 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.



4: The First of Many

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!



5: The Bibliography

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.