

Will Git Be Around Forever?

A List Of Possible Successors

Hanno Embregts

@hannotify



Teaching a course on Git

prehistory

scm classic times

middle ages

the Renaissance

1982

1990

1992

1994

1995

2000

2005

2010

rcs

cvs

perforce

svn

bitkeeper

bazaar
git

dvcs age

clearcase

accurev

darcs
hg

embrace branching,
merging and
distributed...

pvcs

vss

welcome to hell

plastic

laggards

tfs

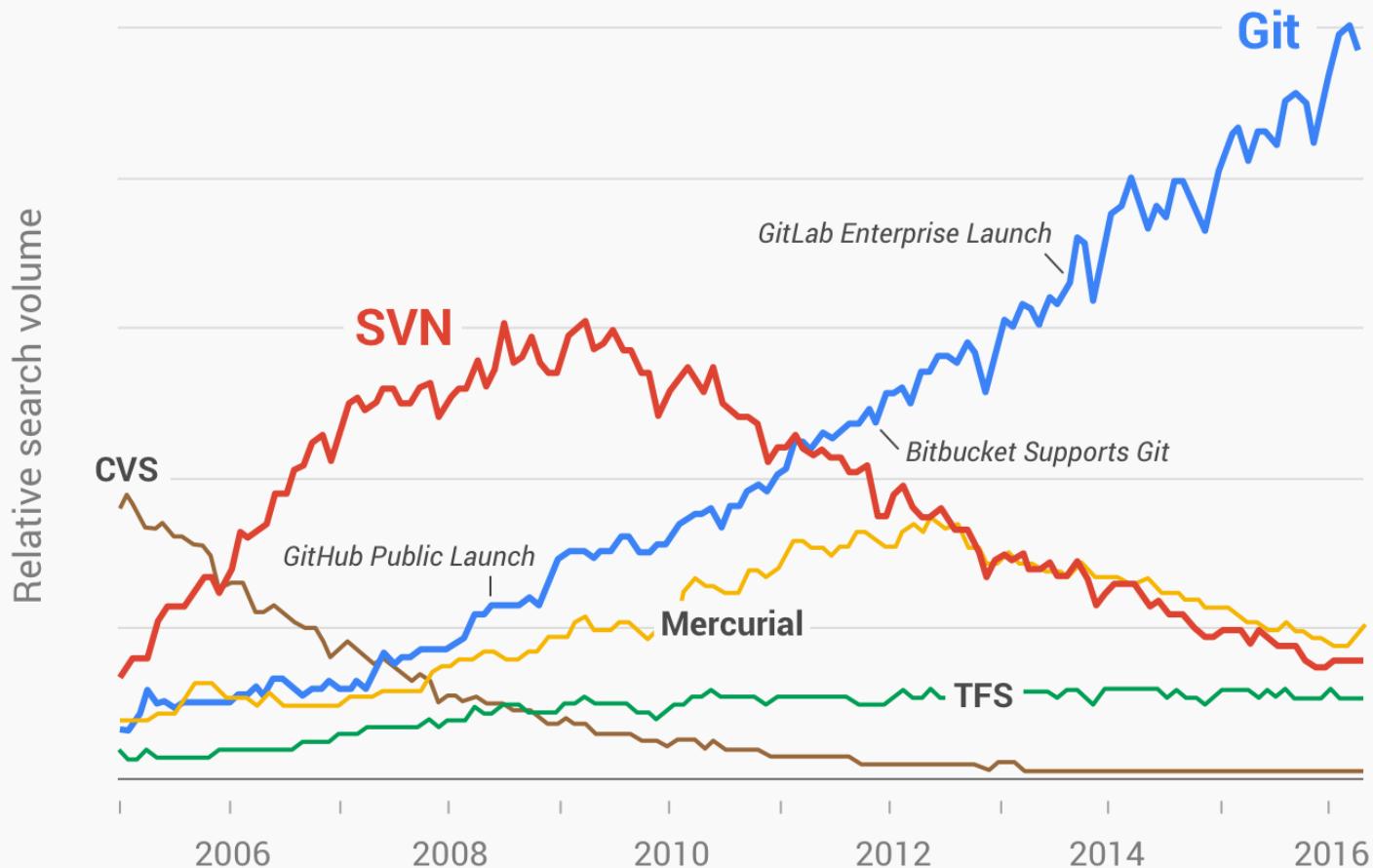




Why did Git become so popular?

How Did SVN Manage to Lose?

Version control interest over time



Reasons for Git's popularity

- Killer features
- Hosting platform support
- Open-source community support

Prediction variables

- Killer features
- Hosting platform support
- Open-source community support
- The handicap of the head start

Putting the graph data into a table

	2010	2021
Subversion	44%	8%
Git	25%	74%
Mercurial	19%	12%
TFVC	7%	6%
CVS	5%	<1%

A photograph of a man sitting on a grey wooden park bench. He is wearing a black t-shirt and grey jeans. His head is buried in his hands, which are clasped behind his head, suggesting distress or exhaustion. The background shows a bridge and some trees.

Don't forget!

Include some new products!

Fossil

SanDisk

leefco.com
Designed in California

EMTEC

Transcend
8GB

Features that Fossil promises

- distributed version control
- integrated bug tracking, wiki, forum and technotes
- built-in web interface
- both autosync and manual merge mode
- show descendants of a check-in
- supports operations on multiple repositories

Quick facts

- Written in C and SQL
- Is the VCS of choice for the SQLite project
- Free code hosting at:
 - Chisel - <http://chiselapp.com/>
- Or simply host it yourself using an SQLite database and a web server

Uses a relational database

- a repository is stored in a single SQLite database file
- contains relations between check-ins to be able to produce both ancestors and descendants of a check-in



Quick demo

- Up and running
- Using the web interface (`fossil ui`)
- Retrieving descendants of a checkin

Git vs. Fossil

	Git	Fossil
Sample project	Linux kernel	SQLite
Contributors	5,000+	4
Engagement	Global	Personal
Workflow	Hierarchy	Trust

Pijul

SanDisk

EMTEC

Transcend
8GB

pi · jul

or *crotophaga sulcirostris*, a bird
known to do collaborative nest
building.

Features that Pijul promises

- distributed version control
- simple, because of its basis on a sound theory of patches
- fast, because it aims to fix the Darcs performance issues
- interactive recording

Quick facts

- written in Rust
- bootstrapped since April 2017
- free code hosting at <https://nest.pijul.com>

Patch-oriented design

- A patch is an intuitive atomic unit of work.
- It focuses on *changes*, instead of *differences between snapshots* (i.e. Git commits).
- Applying or unapplying a patch *doesn't change* its identity.
- The end result of applying several patches is always the same, regardless of the order in which they

GIT

one line
two line
three line

two line
one line
four line

two line
four line
four line

two line
four line
four line

Pijul

+ one line
+ two line
+ three line

- one line
+ one line
- three line
+ four line

- one line
+ four line

two line
four line
four line

If commits were bank transactions

	snapshot	patch
initial balance	100	+100
salary	400	+300
heating	0	-400

Quick demo

- Up and running
- Recording patches
- Reordering patches

Pijul's current status

- v1.0-beta
<https://pijul.org/posts/2022-01-08-beta>

Predictions

Prediction variables

- Killer features.
- Hosting platform support.
- Open-source community support.
- The handicap of a head start.

	Git	Hg	TFVC	Fossil	Pijul
Features	-	-	-	+	++
Hosting	++	+	-	0	0
Open-source	++	++	--	-	+
Dominant	--	-	0	0	0
	+1	+1	-4	0	+3

	2010	2021
Subversion	44%	8%
Git	25%	74%
Mercurial	19%	12%
TFVC	7%	6%
CVS	5%	<1%

	2010	2021	2032
Subversion	44%	8%	3%
Git	25%	74%	77%
Mercurial	19%	12%	8%
TFVC	7%	6%	3%
CVS	5%	<1%	<1%
Pijul		<1%	6%
Fossil	<1%	<1%	3%

	2010	2021	2032
Git	25%	74%	77%
Mercurial	19%	12%	8%
Pijul		<1%	6%
Fossil	<1%	<1%	3%
Subversion	44%	8%	3%
TFVC	7%	6%	3%
CVS	5%	<1%	<1%

Further reading

- "Beyond Git" (by Paweł Świątkowski)
<http://katafrakt.me/2017/05/27/beyond-git/>
- "Fossil vs. Git" (by the Fossil team)
<https://www.fossil-scm.org/home/doc/trunk/www/fossil-v-git.wiki>
- "Merging, patches and Pijul" (by Joe Neeman)
<https://jneem.github.io/pijul/>

Now what?

- A lot of projects are nothing like the Linux kernel.
So try Fossil for a change!
- Git's snapshotting might be too technical for the average user. **So try Pijul for a change!**

**Any
questions?**

Thank you! ☺

nest.pijul.com/hannotify/will-git-be-around-forever

hanno.codes

@hannotify

hanno.embregts@
infosupport.com