

# Will Git Be Around Forever?

A List Of Possible Successors

Hanno Embregts

@hannotify

# Version Control 'By USB Stick'

<https://pxhere.com/en/photo/652221>

# Teaching a course on Git

<https://pxhere.com/en/photo/652221>

prehistory

scm classic times

middle ages

the Renaissance

1982

1990

1992

1994

1995

2000

2005

2010

rcs

cvs

perforce

svn

bitkeeper

bazaar

git

clearcase

accurev

darcs

hg

pvcs

vss

welcome to hell

fear of branching ...

scm redone...

<http://blog.plasticscm.com/2010/11/version-control-timeline.html>

dvcs age

embrace branching,  
merging and  
distributed...

laggards

tfs



# BitMover announces accelerated commercial development strategy and migration plan for Open Source users



SAN FRANCISCO, April 5th 2005 /PRNewswire/ -- BitMover, the technology leader of enterprise development tools, announced today that it will focus exclusively on the development of the commercial version of their industry leading configuration management system, BitKeeper.

"We've experienced explosive growth over the last two years." said Bob Burres, VP of Sales. "We're doubling year over year and our commercial users exceed those using BitKeeper on Open Source projects. Focusing our energy on our cutting edge commercial products will ensure we maintain our competitive advantage and allow us to continue delivering world class development tools."

"BitKeeper is already one of the most advanced SCM products on the market," said Holt Mebane, an architect in HP's Imaging and Personal Systems Group. "BitKeeper has an exciting roadmap, and we recognize that a BitKeeper product with these features will be invaluable to our firmware design efforts."

The commercial version of BitKeeper has an exciting roadmap which includes:

- Integration with major IDEs such as Eclipse
- Revamped GUI tools conforming to Windows standards
- Defect tracking database with a powerful integrated scripting language
- Peer review database system
- Cherry picking across branches
- Scaling to 1,000,000 changesets in one repository
- Scaling to an unlimited number of files
- Module (subtree) support
- Enhanced binary management
- Better license management and control
- Management reporting tools

<http://web.archive.org/web/20050408062838/http://www.bitkeeper.com/press/2005-04-05.html>

BitMover looks forward to implementing our extensive roadmap and delivering advanced SCM technology to a wider market.

As part of this focus, BitMover has replaced the free version of BitKeeper with the recently released open source BitKeeper client. Those developers who desire additional functionality may choose to migrate to the more powerful commercial version of BitKeeper.

Major projects hosted at [www.bkbits.net](http://www.bkbits.net), such as the Linux kernel, will continue to be hosted and updated there.

Our relationship with the Open Source community has been evolving and many of the key developers have already migrated to the commercial version of BitKeeper. Those developers will continue to enjoy the full features of BitKeeper.

prehistory

scm classic times

middle ages

the Renaissance

1982

1990

1992

1994

1995

2000

2005

2010

rcs

cvs

perforce

svn

bitkeeper

bazaar

git

clearcase

accurev

darcs

hg

pvcs

vss

welcome to hell

fear of branching ...

scm redone...

<http://blog.plasticscm.com/2010/11/version-control-timeline.html>

dvcs age

embrace branching,  
merging and  
distributed...

laggards

tfs





"640K ought to be enough for anybody  
for now..."

[https://upload.wikimedia.org/wikipedia/commons/thumb/3/31/Bill\\_Gates\\_2004.jpg/1280px-Bill\\_Gates\\_2004.jpg](https://upload.wikimedia.org/wikipedia/commons/thumb/3/31/Bill_Gates_2004.jpg/1280px-Bill_Gates_2004.jpg)

A black and white photograph of a rugged coastline. In the foreground, dark, jagged rocks protrude from the water where white-capped waves break. The middle ground shows more of the rocky shore and the churning sea. In the background, a range of mountains or hills is visible through a thick, misty atmosphere, creating a sense of depth and time.

Change is the only constant.

Photo credit: Ryan McGuire

Heraclitus



# Why did Git become so popular?

# Main features of Git

- Distributed
- Free & open-source
- Fast
- Optimized for branching

# Some surprising facts on Git

*"I'm an egotistical bastard, so I name all my projects after myself. First Linux, now git."*

<https://www.infoworld.com/article/2669670/after-controversy--torvalds-begins-work-on--git-.html>

# WWCVSND?

If in doubt, make the exact opposite decision.

<https://pxhere.com/en/photo/704019>

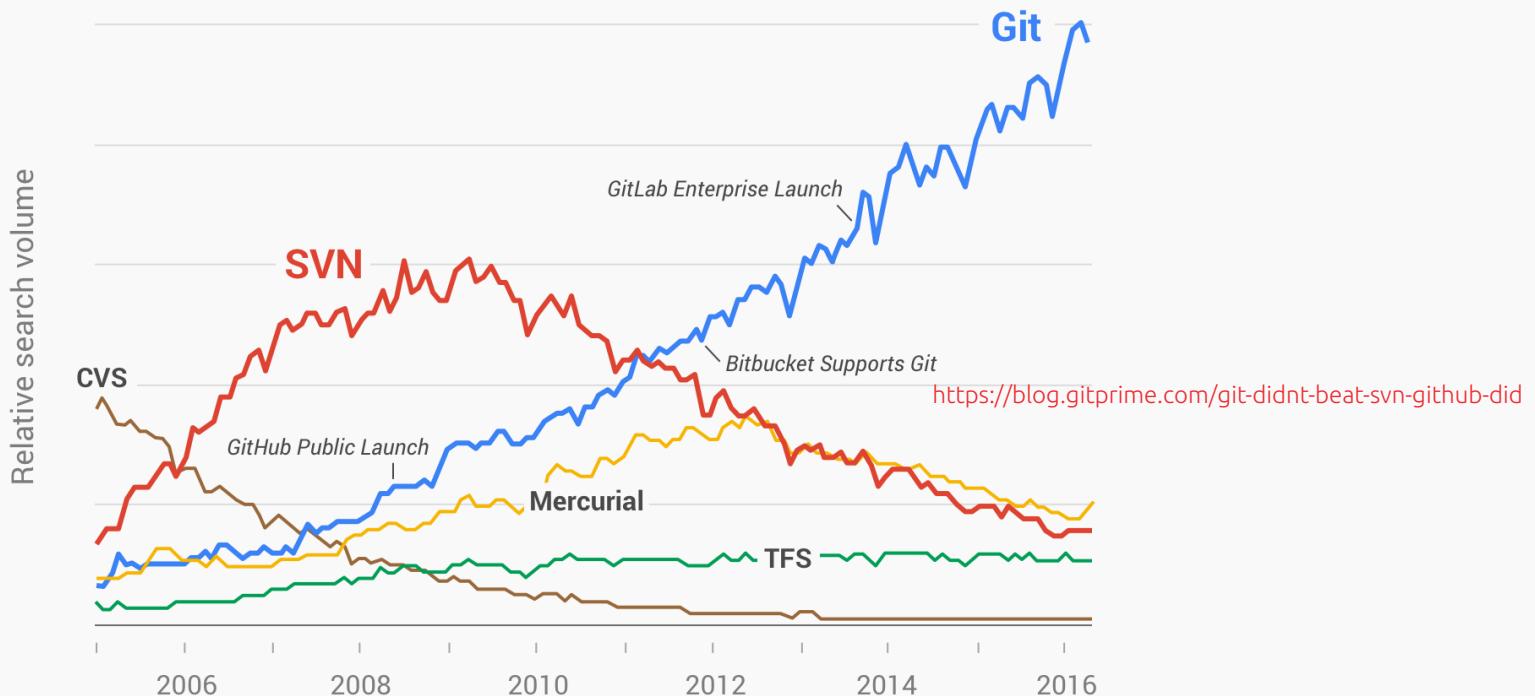
# Killer features.

prediction variable #1

<https://www.pexels.com/photo/photo-displays-person-holding-ball-with-reflection-of-horizon-940880>

# How Did SVN Manage to Lose?

Version control interest over time



# Create a repository

X

Repository type

 Git

✓  Git

 TFVC

Add a README

# Hosting platform support.

prediction variable #2

<https://www.pexels.com/photo/photo-displays-person-holding-ball-with-reflection-of-horizon-940880>

# Open-source community support.

prediction variable #3

<https://www.pexels.com/photo/photo-displays-person-holding-ball-with-reflection-of-horizon-940880>

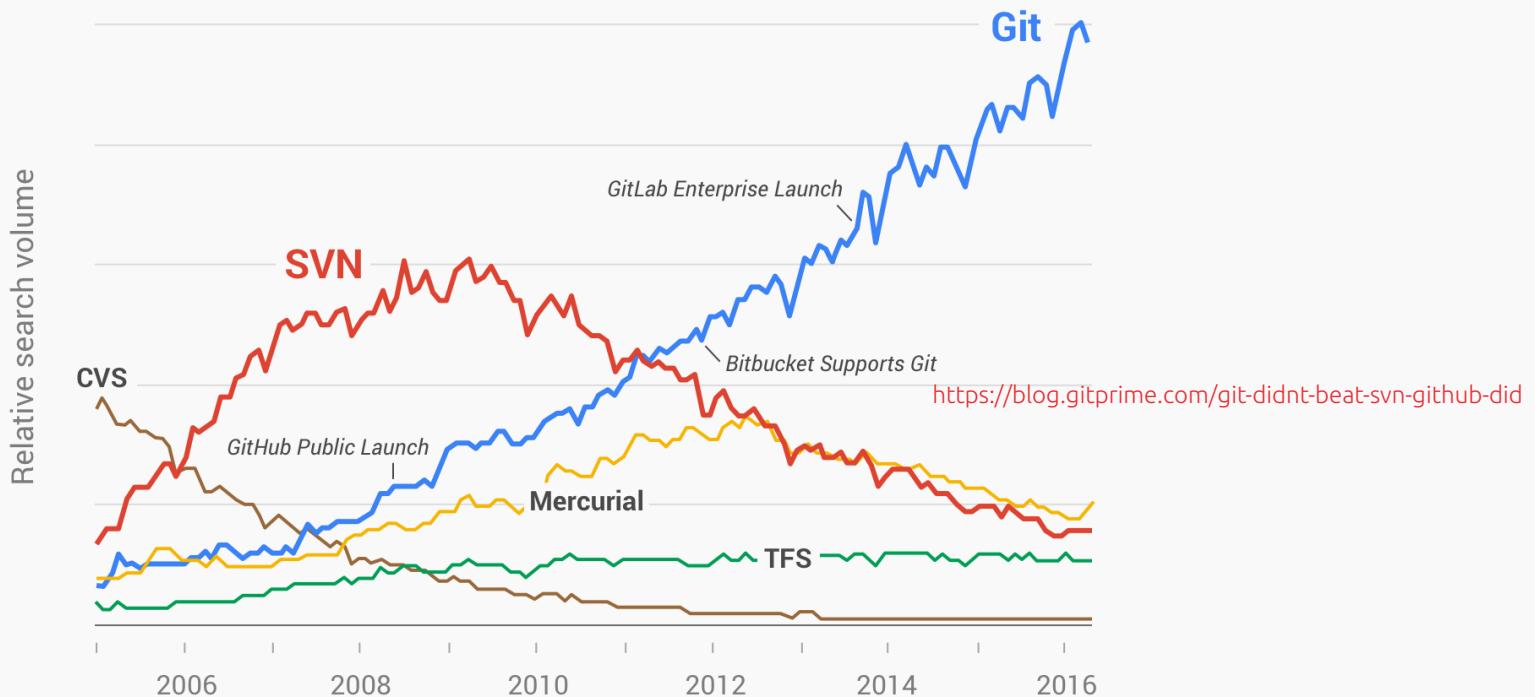
# THERE ARE TWO TYPES OF PEOPLE

<https://bingeprints.com/products/there-are-two-types-of-people-those-who-can-extrapolate-from-incomplete-data-mug-funny-statistics-math-coffee-cup-1>

- ① THOSE WHO CAN EXTRAPOLATE FROM INCOMPLETE DATA

# How Did SVN Manage to Lose?

Version control interest over time



# First Prediction

2010

---

**Subversion** 43%

---

**Git** 19%

---

**Mercurial** 16%

---

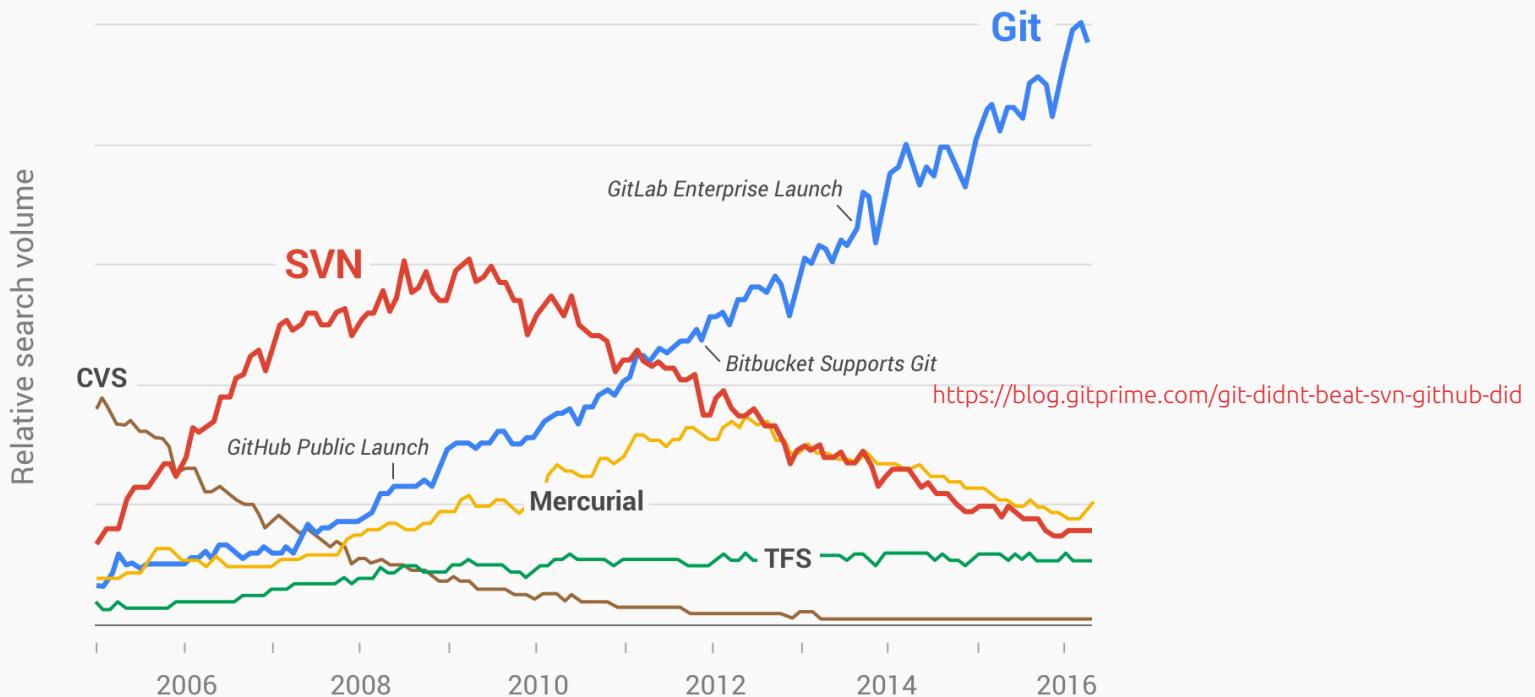
**TFVC** 6%

---

**CVS** 5%

# How Did SVN Manage to Lose?

Version control interest over time

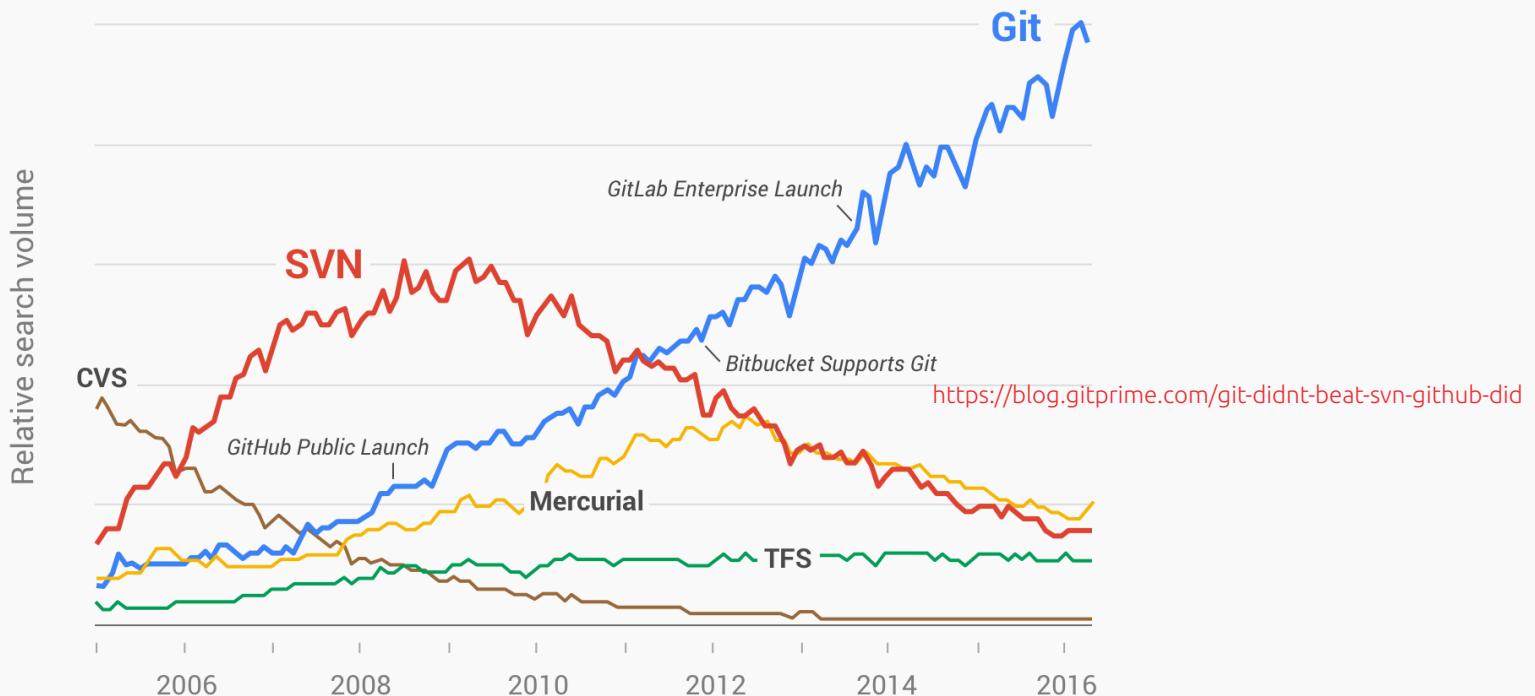


# First Prediction

	2010	2020
<b>Subversion</b>	43%	8%
<b>Git</b>	19%	73%
<b>Mercurial</b>	16%	12%
<b>TFVC</b>	6%	7%
<b>CVS</b>	5%	<1%

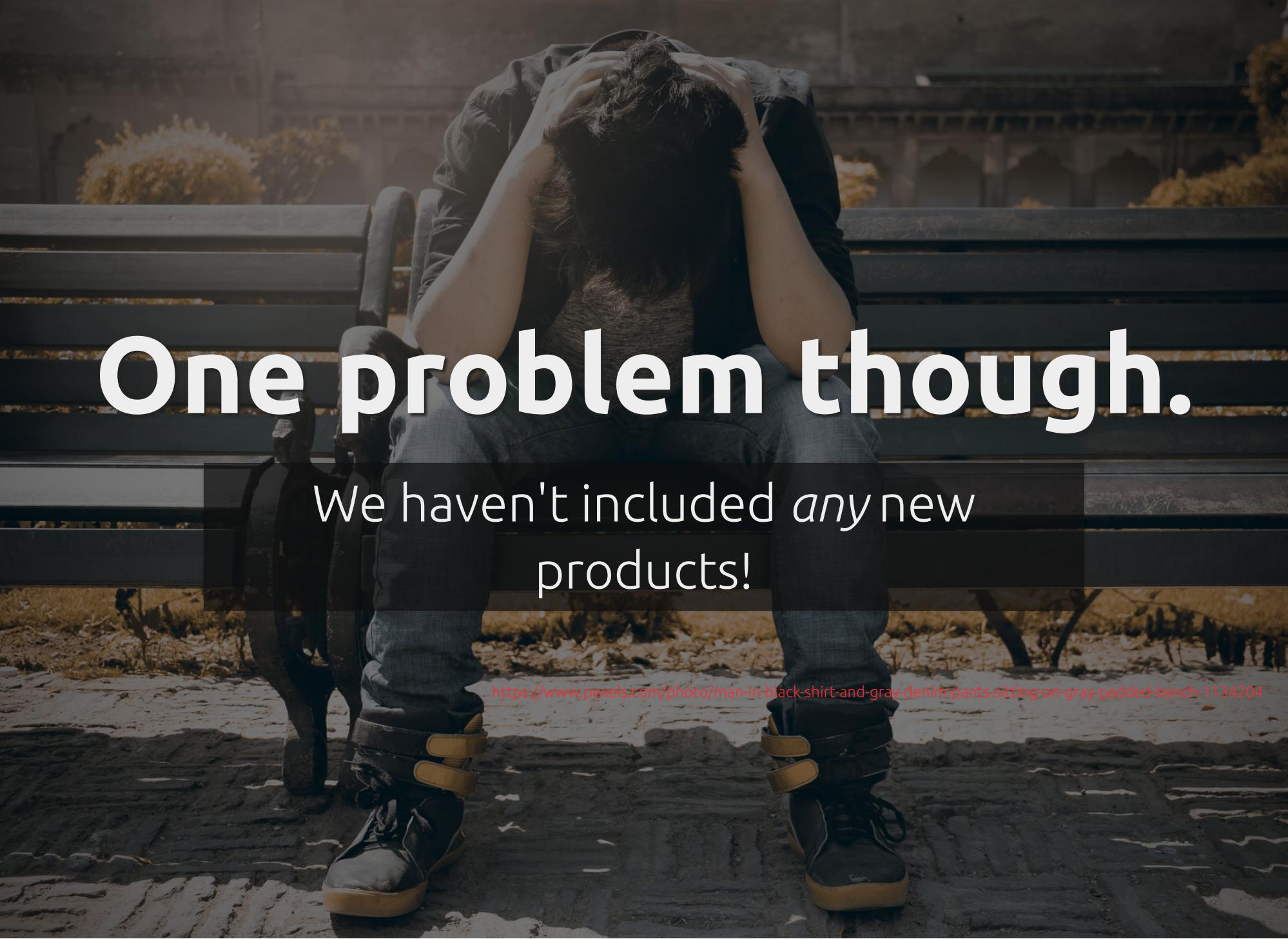
# How Did SVN Manage to Lose?

Version control interest over time



# First Prediction

	2010	2020	2030
<b>Subversion</b>	43%	8%	4%
<b>Git</b>	19%	73%	80%
<b>Mercurial</b>	16%	12%	9%
<b>TFVC</b>	6%	7%	7%
<b>CVS</b>	5%	<1%	<1%

A photograph of a man sitting on a grey metal bench outdoors. He is wearing a black long-sleeved shirt, blue jeans, and brown leather shoes with laces. His head is buried in his hands, which are clasped behind his head, suggesting distress or despair. The background shows a blurred urban environment with trees and a bridge.

# One problem though.

We haven't included *any* new products!

<https://www.pexels.com/photo/man-in-black-shirt-and-gray-denim-pants-sitting-on-gray-padded-bench-1134204>

# The Browser Wars

<https://pxhere.com/en/photo/652221>



### Marktaandeel browsers (januari 2005):

- |                        |        |
|------------------------|--------|
| 1. Microsoft IE 6.0    | 80.95% |
| 2. Mozilla Firefox 1.0 | 4.58%  |
| 3. Microsoft IE 5.0    | 4.18%  |
| 4. Microsoft IE 5.5    | 3.66%  |
| 5. Mozilla 1.x         | 2.77%  |
| 6. Opera 7.x           | 1.29%  |

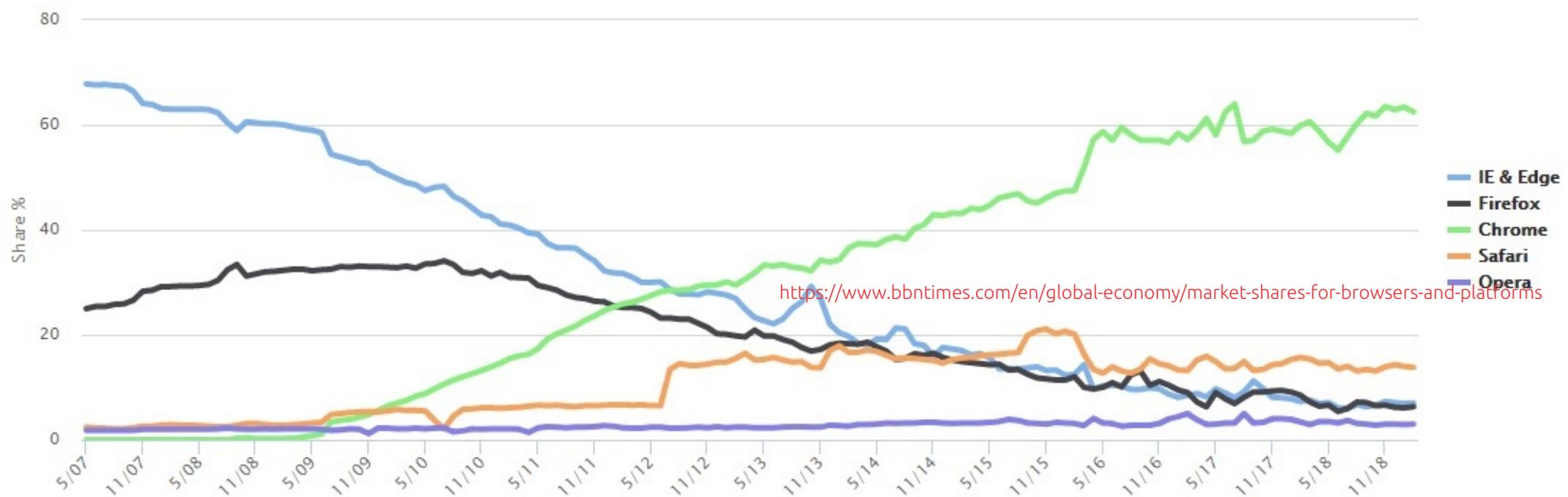
# In ten years time...

1. ...Internet Explorer would be surpassed as the top browser family;
2. ...Mozilla Firefox would surpass Internet Explorer;
3. ...Mozilla Firefox would be the top browser.

# In ten years time...

1. ...Internet Explorer would be surpassed as the top browser family;
2. ...Mozilla Firefox would surpass Internet Explorer;
3. ...~~Mozilla Firefox~~ would be the top browser.

## Browser Family Monthly Usage Share





# The handicap of a head start

You will get a lot of attention.

Jan Romein, "The dialectics of progress", 1937

<https://www.pexels.com/photo/battle-black-blur-board-game-260024>

# The handicap of a head start

prediction variable #4

<https://www.pexels.com/photo/photo-displays-person-holding-ball-with-reflection-of-horizon-940880>

# Three New(er) Kids On The Block

A collection of various USB flash drives (SanDisk, EMTEC, Transcend) scattered on a textured surface.

<https://pxhere.com/en/photo/652221>

# Three new(er) kids on the block

- Plastic (2006)
- Fossil (2006)
- Pijul (2015)

# Plastic

- Fast branching
- Merge previews
- Both central and distributed
- Comes with a lot of GUI stuff



# Focus on artists and designers

- 'Plastic Gluon' is a version control component meant for artists and designers
- Image diffing
- References on their website are exclusively from game studios

Side by side

Differences

Swipe

Show properties



rev:c:\Users\pablo\wkspaces\ironman\_code\ironman\_armor.jpg#cs:3

rev:c:\Users\pablo\wkspaces\ironman\_code\ironman\_armor.jpg#cs:2



<https://www.plasticscm.com/images/gallery/large/image-02-diff-swipe.png>

# But

**Plastic is not going to be 'the Git killer'**

- it's not open-source software
- Git users are not changing their VCS just for some eye candy

# Fossil

<https://pxhere.com/en/photo/652221>

# 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  
`(fossil all)`
- has a 'preserve all history' philosophy

# Quick facts

- Written in C and SQL
- Is the VCS of choice for the SQLite project
- 'Github in a box'
- 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

# 'Github in a box'

- repository UI (`fossil ui`)
- wiki, ticketing & bug tracking
- embedded documentation
- technical notes
- forum

# Quick demo

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

# The Bad

- Dealing with large files
- Dealing with large commits

<https://www.omiyagames.com/blog/2014/02/15/farewell-fossil-version-control>

# Git vs. Fossil

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

<https://www.fossil-scm.org/home/doc/trunk/www/fossil-v-git.wiki>

# Pijul

<https://pxhere.com/en/photo/652221>

# pi • jul

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

[https://commons.wikimedia.org/wiki/File:Museo\\_de\\_la\\_Naturaleza\\_de\\_Cantabria\\_\(208\).jpg](https://commons.wikimedia.org/wiki/File:Museo_de_la_Naturaleza_de_Cantabria_(208).jpg)

8A

Pijul, garrapatero pijuy *Crotophaga sulcirostris* Texas - Chile  
Pijul, garrapatero pijuy *Crotophaga sulcirostris* Texas - Chile

Quiscal o zanate mexicano *Quiscalus mexicanus*  
EEUU - Perú

# 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

<https://www.pijul.com>

# Quick facts

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



Daniel Terhorst-North

@tastapod

...

So where should I host a new open source project?  
[@github](#) seems to want to own everything anyone has  
every done. Is [@gitlab](#) where the kids are moving?

# 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 were applied.
- Pijul keeps track of 'dependent patches'
- No merges or rebases; applying a patch is like `git cherry-pick`.

# 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

<https://www.katacoda.com/ysndr/scenarios/pijul/assets/comparison.png>

# Quick demo

- Up and running
- Recording patches
- Nest demo - [https://nest.pijul.com/pijul\\_org/pijul](https://nest.pijul.com/pijul_org/pijul)

# The Bad

- Usability needs some work  
<https://mivehind.net/2017/04/09/pijul-first-thoughts>
- Complete rewrite in progress for v1.0  
<https://pijul.org/posts/2020-11-07-towards-1.0>

# Pijul towards v1.0

- New change format; 'patches' are now called 'changes'
- 'Branches' are now called 'channels'
- Better support for large files and repositories by compressing changes
- Interactive recording is replaced by a 'change draft screen'
- Version identifiers that don't depend on any order
- Inclusive author names
- Documentation is lagging a bit



<https://www.reuters.com/article/us-france-ovh-fire-idUSKBN2B20NU>

# Predictions

<https://pxhere.com/en/photo/652221>

# Prediction variables

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

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

**2030**

---

**Git** 80%

---

**Mercurial** 9%

---

**TFVC** 7%

---

**Subversion** 4%

---

**CVS** <1%

**2030**    **2030**

---

**Git**    80%    77%

---

**Mercurial**    9%    8%

---

**Pijul**                5%

---

**TFVC**    7%    4%

---

**Subversion**    4%    4%

---

**Fossil**                2%

---

**CVS**    <1%    <1 %

# Further reading

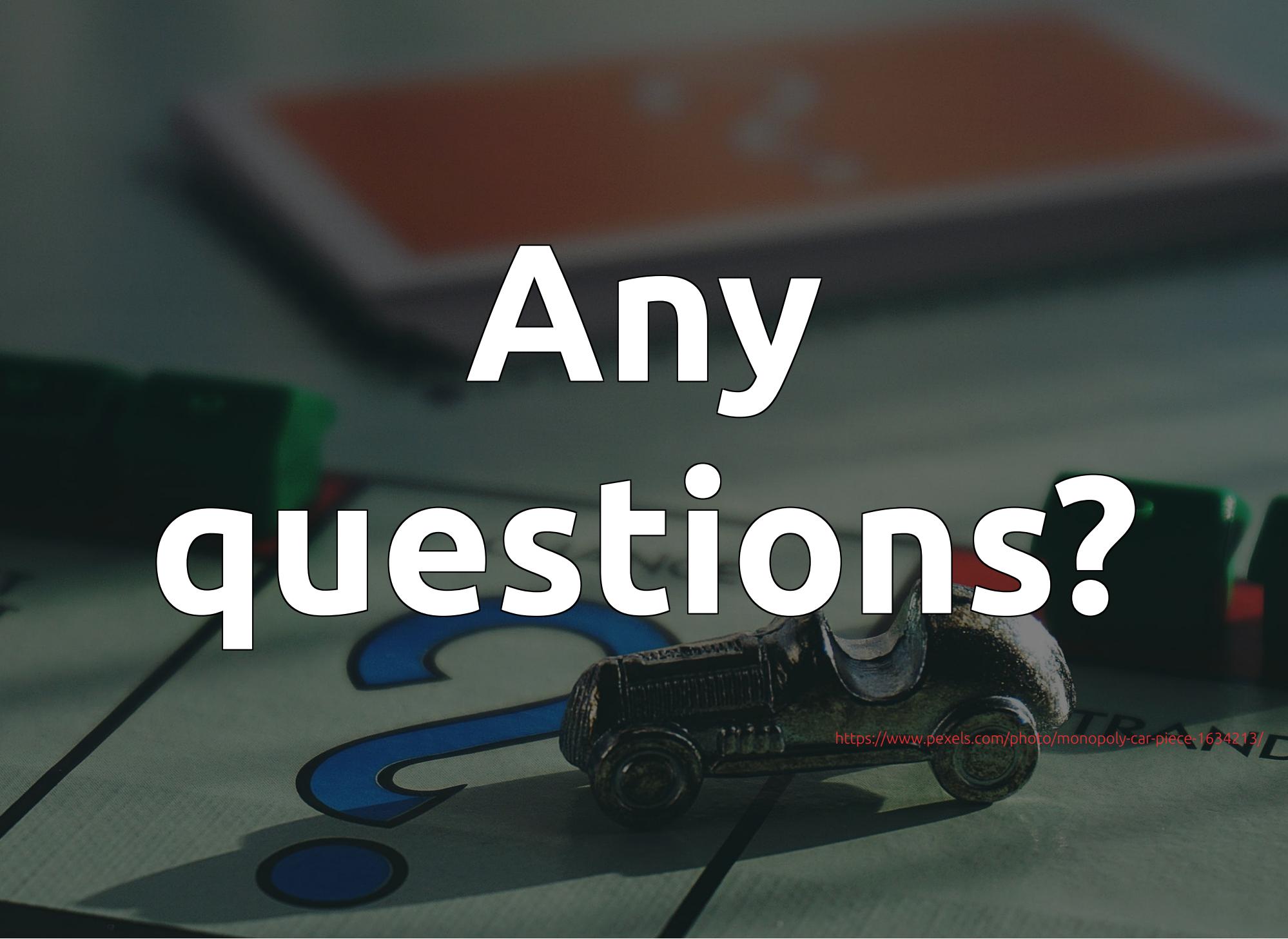
<https://www.pexels.com/photo/white-teddy-bear-reading-book-33196/>

# 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/>
- Try out Pijul @ [katacoda.com](https://www.katacoda.com/ysndr/scenarios/pijul)  
<https://www.katacoda.com/ysndr/scenarios/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 Git user. **So try Pijul for a change!**



# Any questions?

<https://www.pexels.com/photo/monopoly-car-piece-1634213/>

# Thank you! 😊

[bit.do/utrechtjug-successor-of-git](https://bit.do/utrechtjug-successor-of-git)

---

[hanno.codes](http://hanno.codes)

---

[@hannotify](https://twitter.com/hannotify)

---

[hanno.embregts@infosupport.com](mailto:hanno.embregts@infosupport.com)