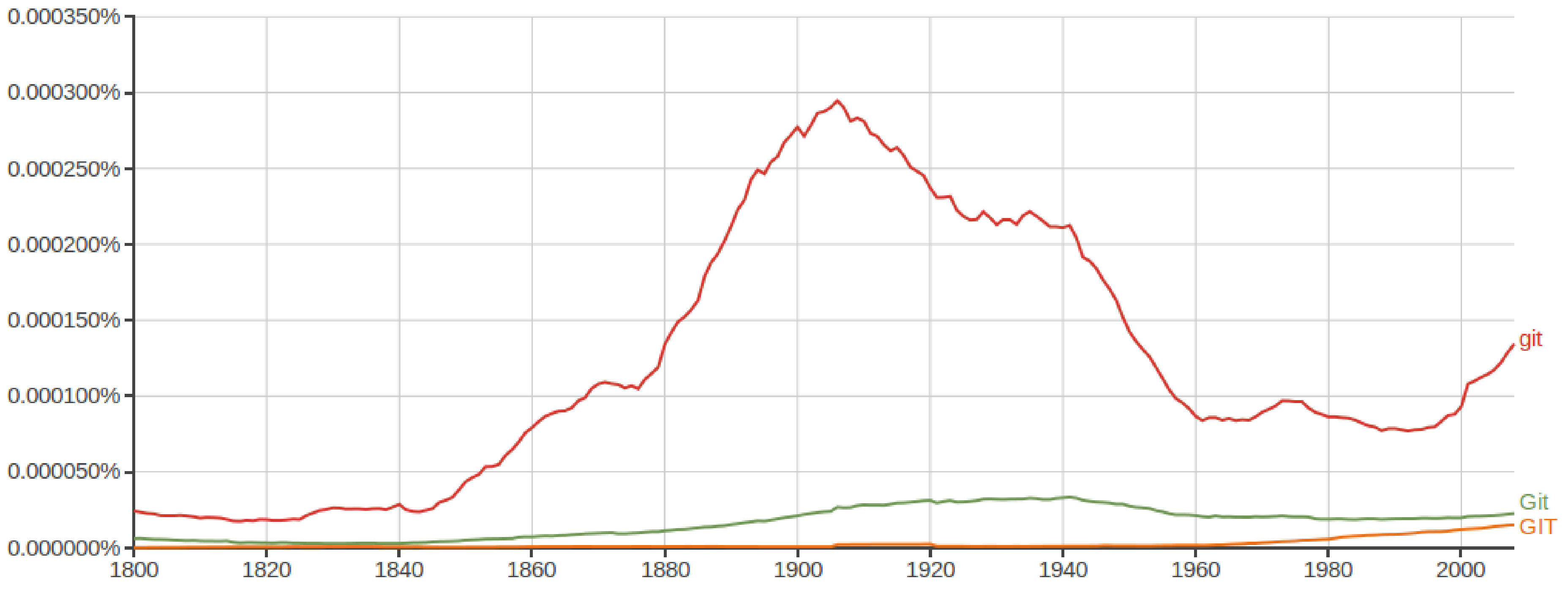
git

An overview

Git: Definition

noun British *informal*
noun: **git**; plural noun: **gits**

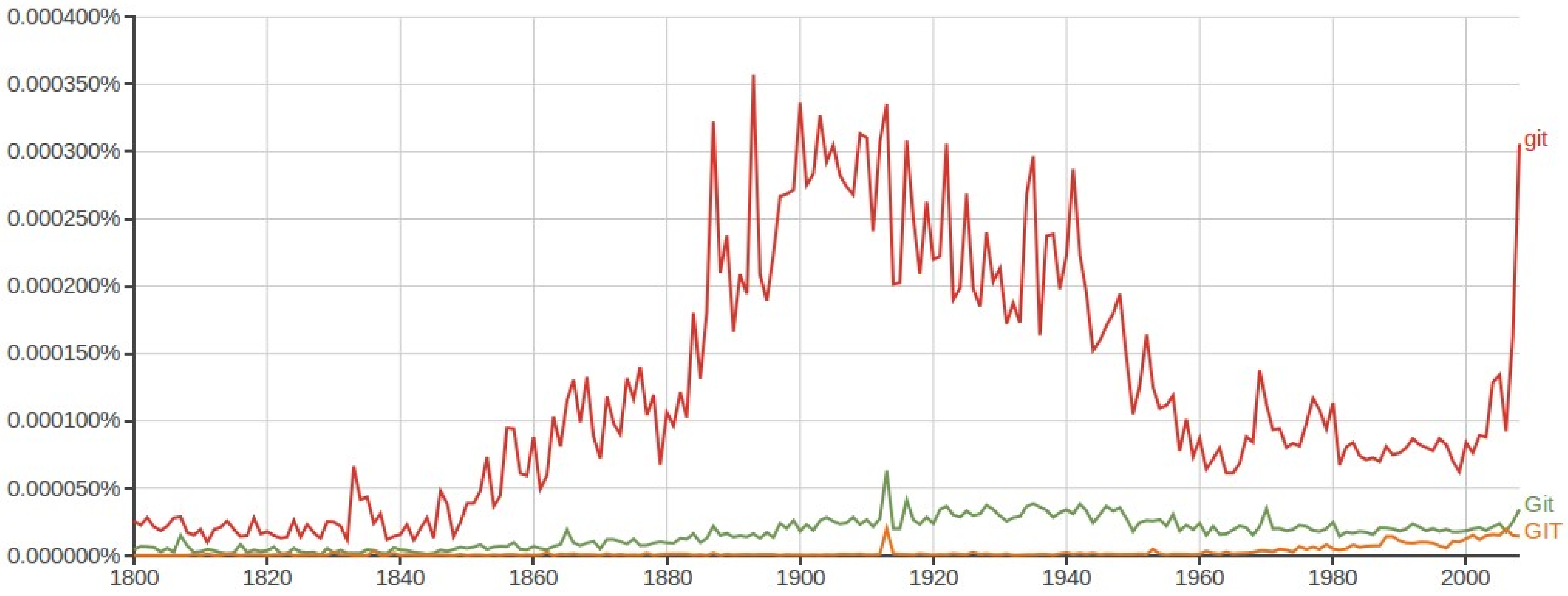
an unpleasant or contemptible person. *Google*



Git: Definition

noun British *informal*
noun: **git**; plural noun: **gits**

an unpleasant or contemptible person. *Google*



Git: Definition

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is easy to learn and has a tiny footprint with lightning fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows.

Git: Definition

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is easy to learn and has a tiny footprint with lightning fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows.

Git: Definition

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is easy to learn and has a tiny footprint with lightning fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows.

Git: Definition

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is easy to learn and has a tiny footprint with lightning fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows.

Git: Definition

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is easy to learn and has a tiny footprint with lightning fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows.

Git: Definition

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is easy to learn and has a tiny footprint with lightning fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows.

Git: Definition

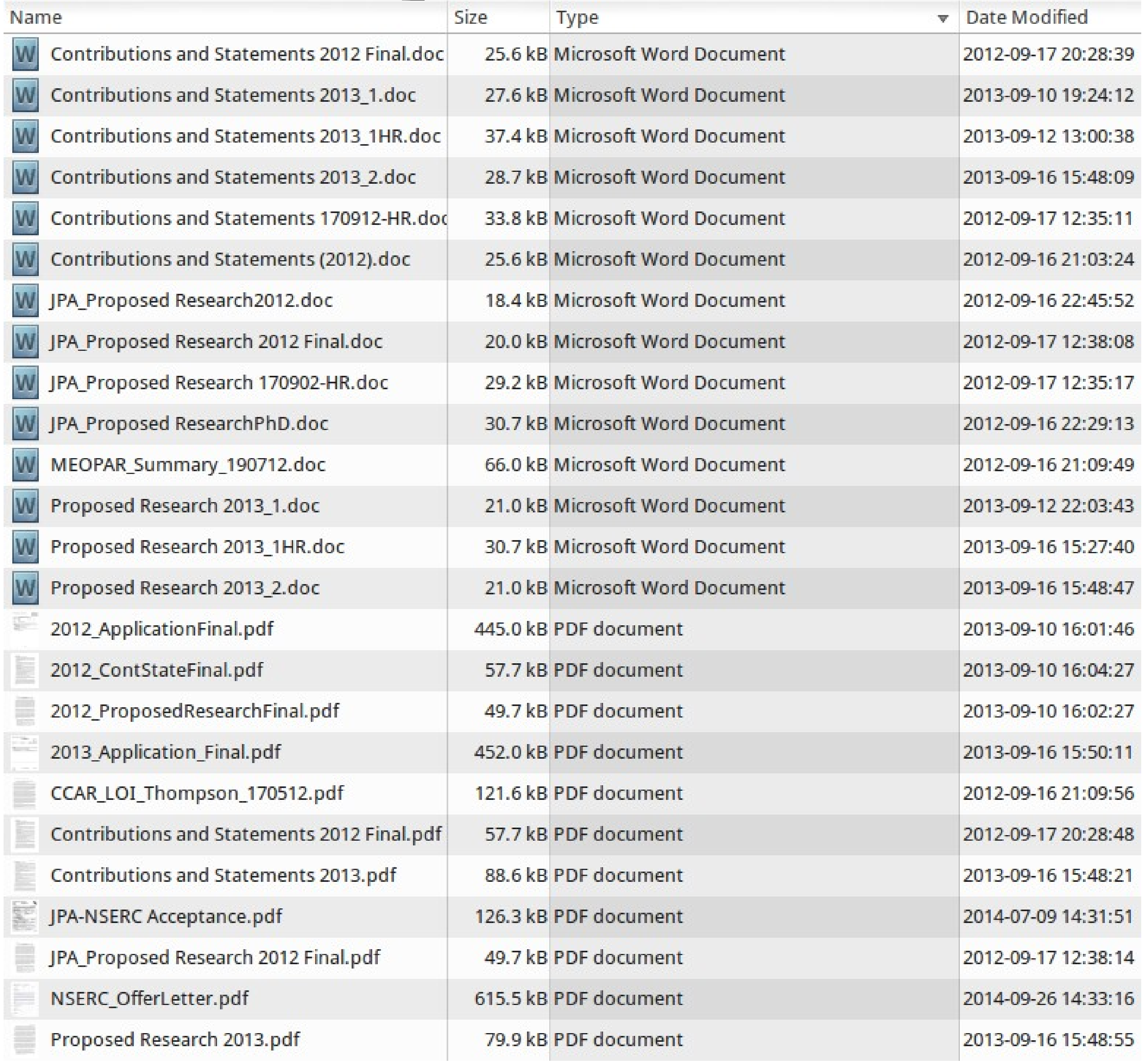
Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is easy to learn and has a tiny footprint with lightning fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows.

Show-off git!

Git: What's it good for?

- **Keeping track of evolving text based documents.**

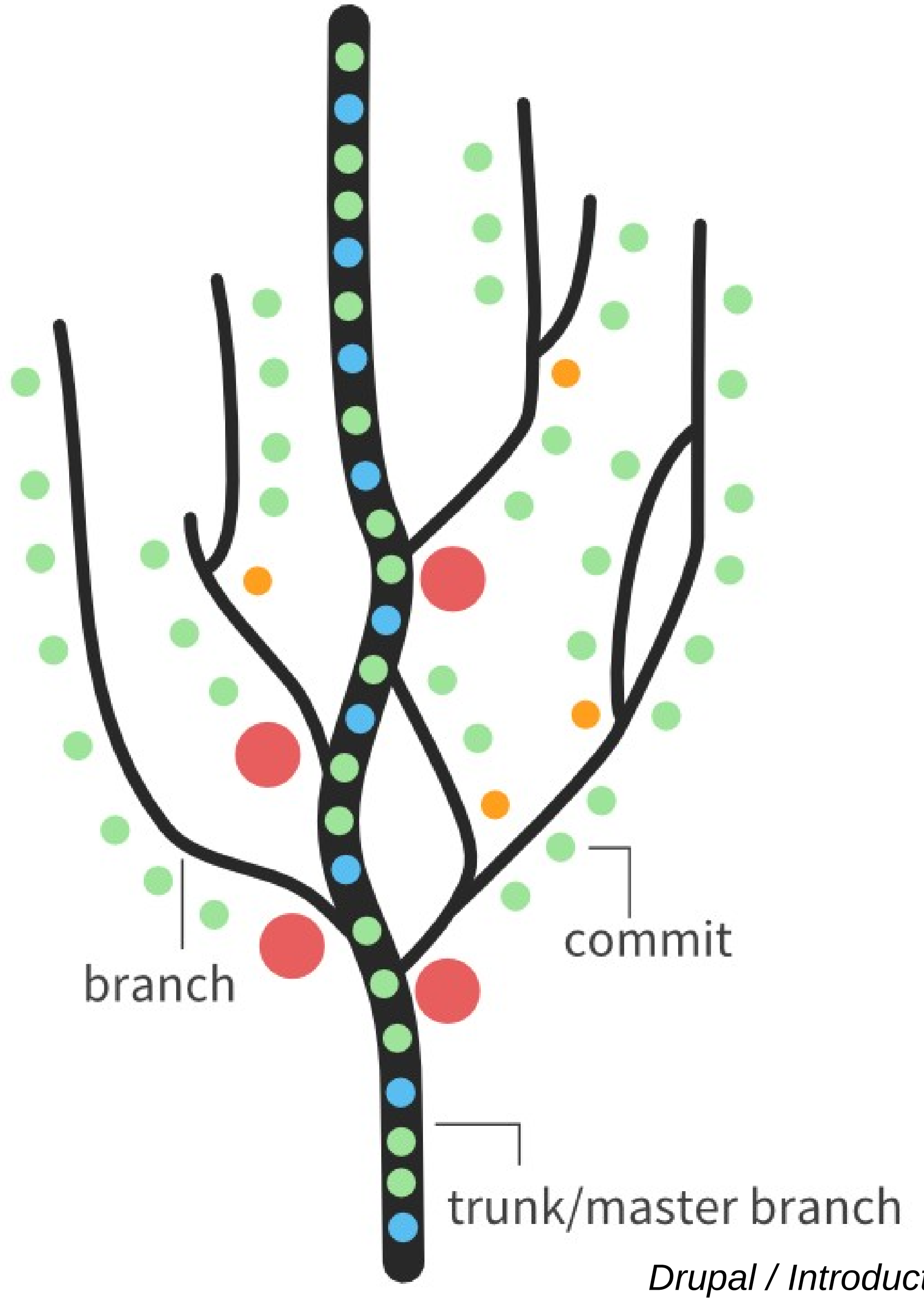


Name	Size	Type	Date Modified
Contributions and Statements 2012 Final.doc	25.6 kB	Microsoft Word Document	2012-09-17 20:28:39
Contributions and Statements 2013_1.doc	27.6 kB	Microsoft Word Document	2013-09-10 19:24:12
Contributions and Statements 2013_1HR.doc	37.4 kB	Microsoft Word Document	2013-09-12 13:00:38
Contributions and Statements 2013_2.doc	28.7 kB	Microsoft Word Document	2013-09-16 15:48:09
Contributions and Statements 170912-HR.doc	33.8 kB	Microsoft Word Document	2012-09-17 12:35:11
Contributions and Statements (2012).doc	25.6 kB	Microsoft Word Document	2012-09-16 21:03:24
JPA_Proposed Research2012.doc	18.4 kB	Microsoft Word Document	2012-09-16 22:45:52
JPA_Proposed Research 2012 Final.doc	20.0 kB	Microsoft Word Document	2012-09-17 12:38:08
JPA_Proposed Research 170902-HR.doc	29.2 kB	Microsoft Word Document	2012-09-17 12:35:17
JPA_Proposed ResearchPhD.doc	30.7 kB	Microsoft Word Document	2012-09-16 22:29:13
MEOPAR_Summary_190712.doc	66.0 kB	Microsoft Word Document	2012-09-16 21:09:49
Proposed Research 2013_1.doc	21.0 kB	Microsoft Word Document	2013-09-12 22:03:43
Proposed Research 2013_1HR.doc	30.7 kB	Microsoft Word Document	2013-09-16 15:27:40
Proposed Research 2013_2.doc	21.0 kB	Microsoft Word Document	2013-09-16 15:48:47
2012_ApplicationFinal.pdf	445.0 kB	PDF document	2013-09-10 16:01:46
2012_ContStateFinal.pdf	57.7 kB	PDF document	2013-09-10 16:04:27
2012_ProposedResearchFinal.pdf	49.7 kB	PDF document	2013-09-10 16:02:27
2013_Application_Final.pdf	452.0 kB	PDF document	2013-09-16 15:50:11
CCAR_LOI_Thompson_170512.pdf	121.6 kB	PDF document	2012-09-16 21:09:56
Contributions and Statements 2012 Final.pdf	57.7 kB	PDF document	2012-09-17 20:28:48
Contributions and Statements 2013.pdf	88.6 kB	PDF document	2013-09-16 15:48:21
JPA-NSERC Acceptance.pdf	126.3 kB	PDF document	2014-07-09 14:31:51
JPA_Proposed Research 2012 Final.pdf	49.7 kB	PDF document	2012-09-17 12:38:14
NSERC_OfferLetter.pdf	615.5 kB	PDF document	2014-09-26 14:33:16
Proposed Research 2013.pdf	79.9 kB	PDF document	2013-09-16 15:48:55

Git: What's it good for?

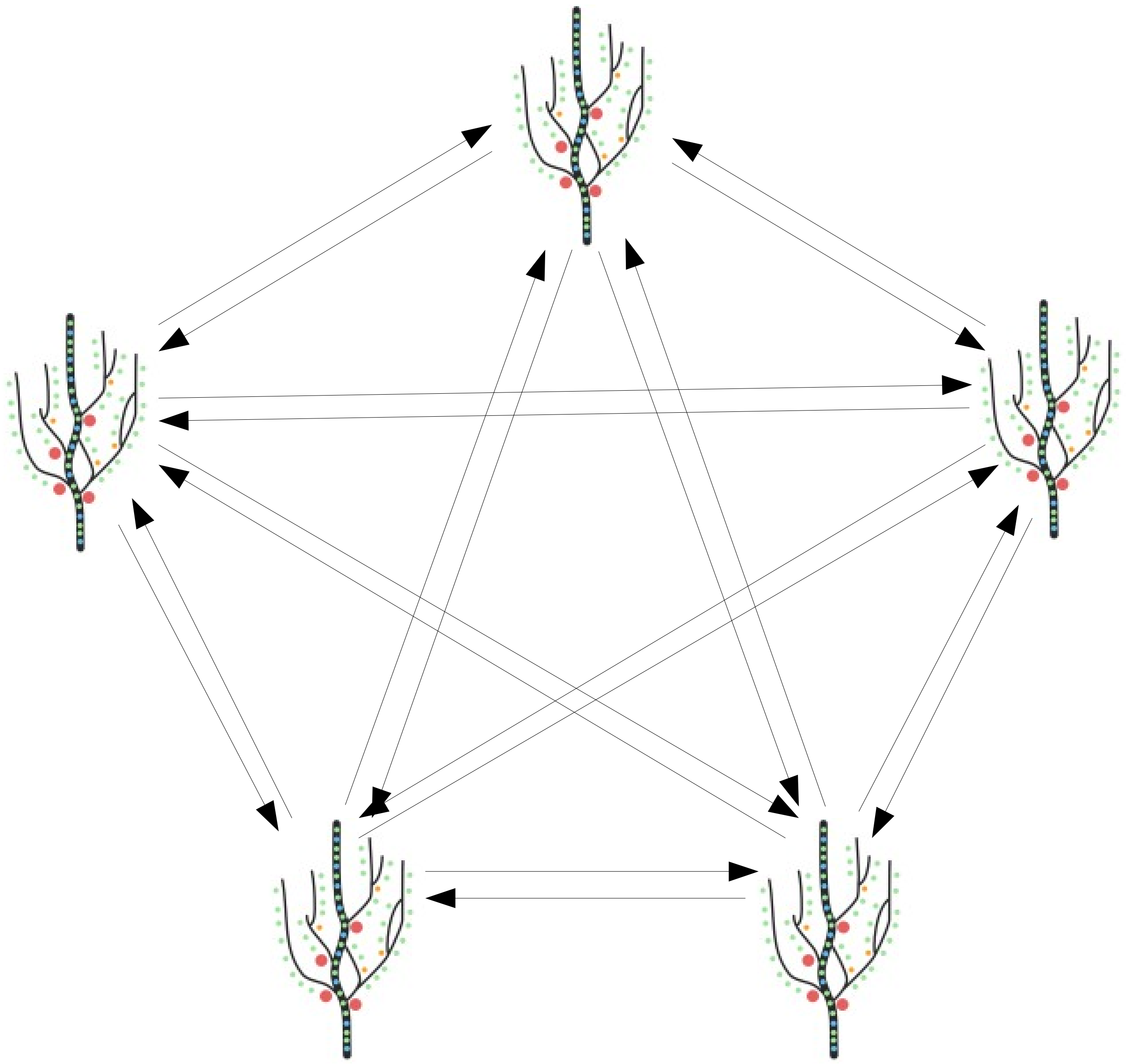
- Keeping track of evolving text based documents.

- **Working on many features of one project at the same time.**



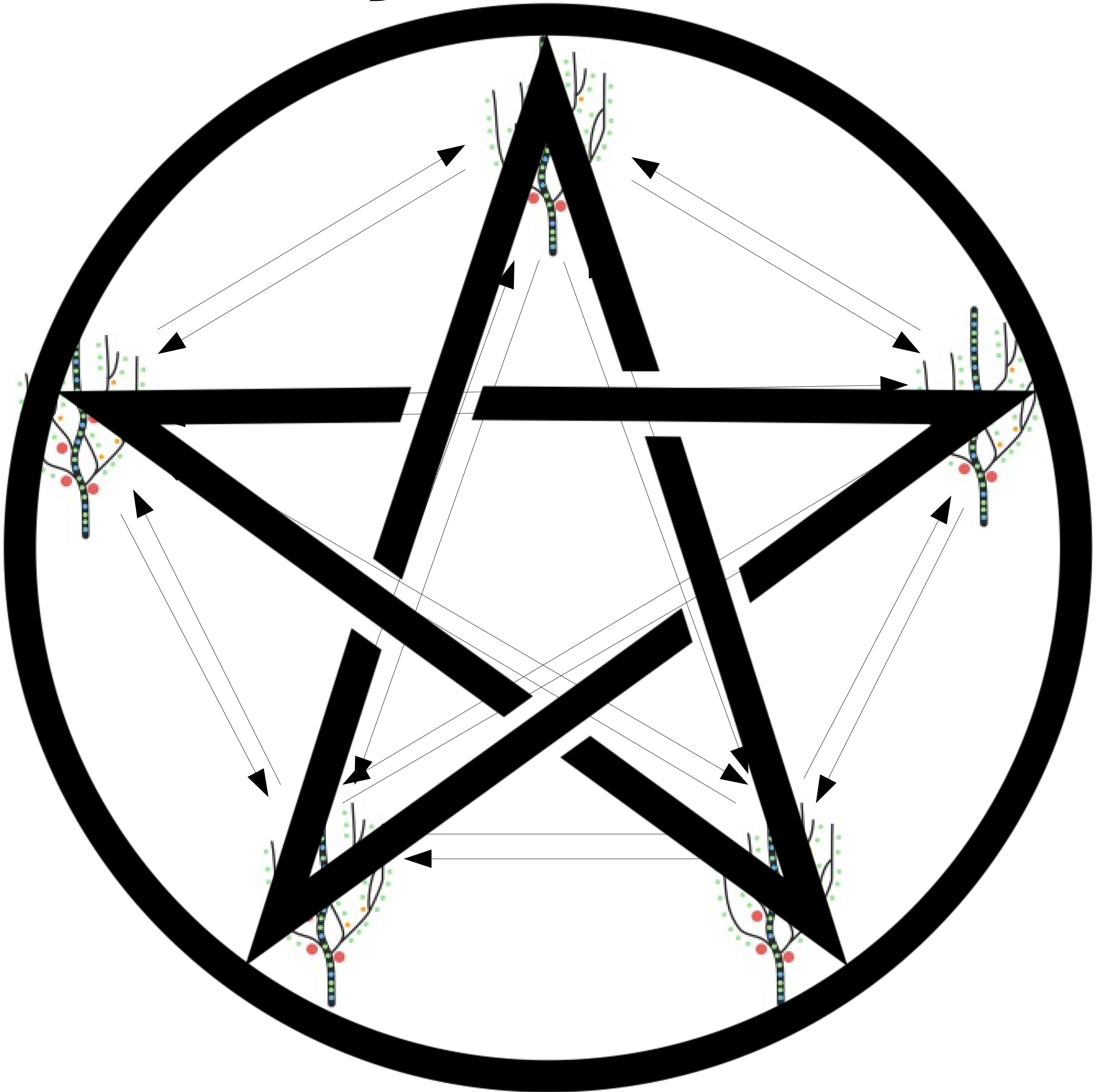
Git: What's it good for?

- Keeping track of evolving text based documents.
- Working on many features of one project at the same time.
- **Working in multiple locations or with multiple people.**



Git: What's it good for?

- Keeping track of evolving text based documents.
- Working on many features of one project at the same time.
- **Working in multiple locations or with multiple people.**

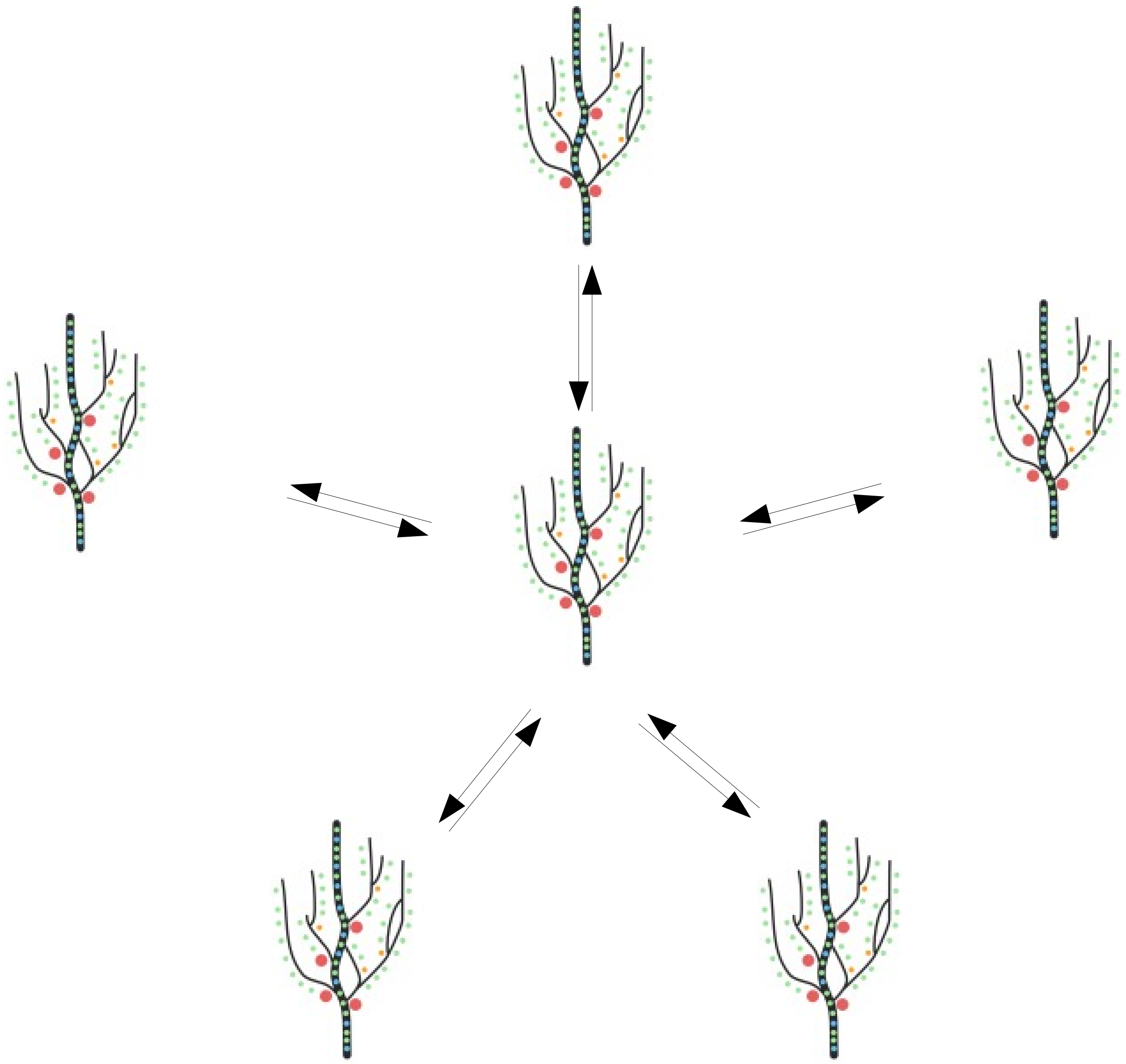


Git: What's it good for?

- Keeping track of evolving text based documents.

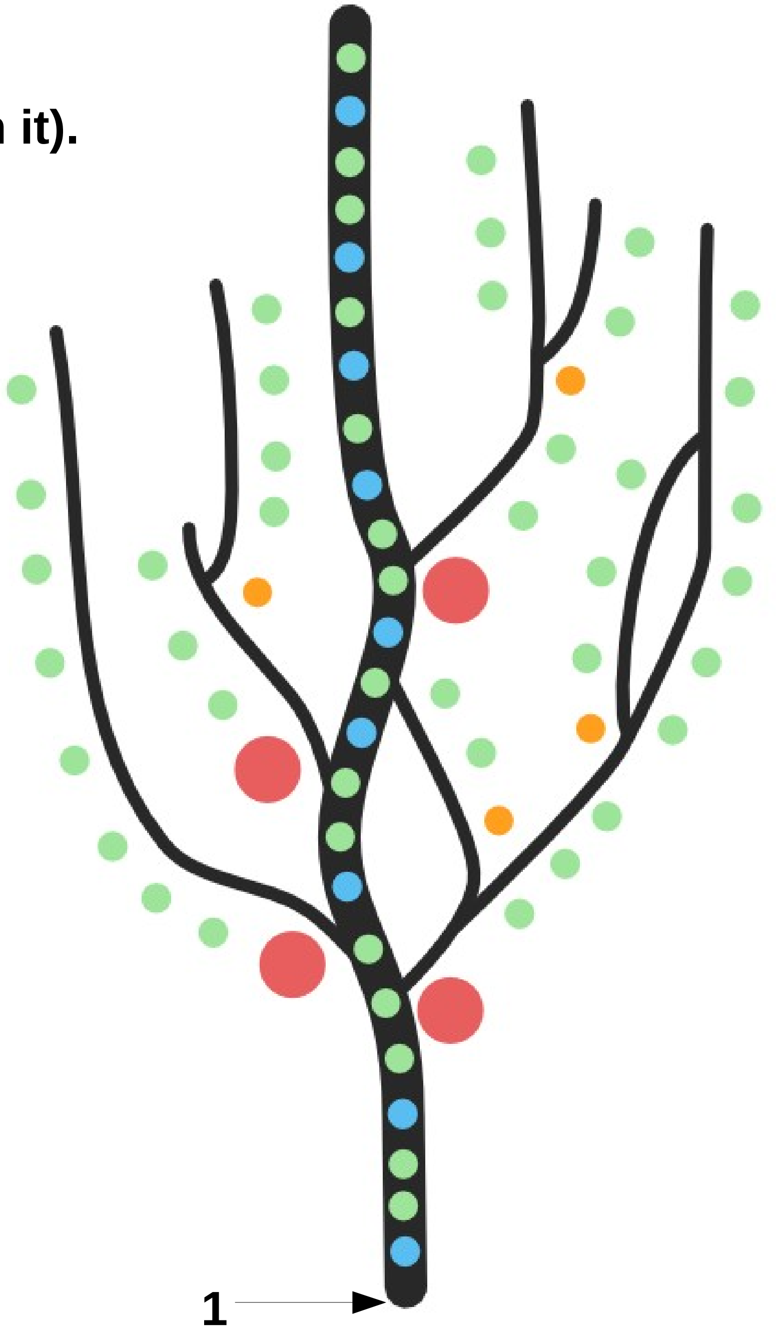
- Working on many features of one project at the same time.

- **Working in multiple locations or with multiple people.**



Git: How to get started?

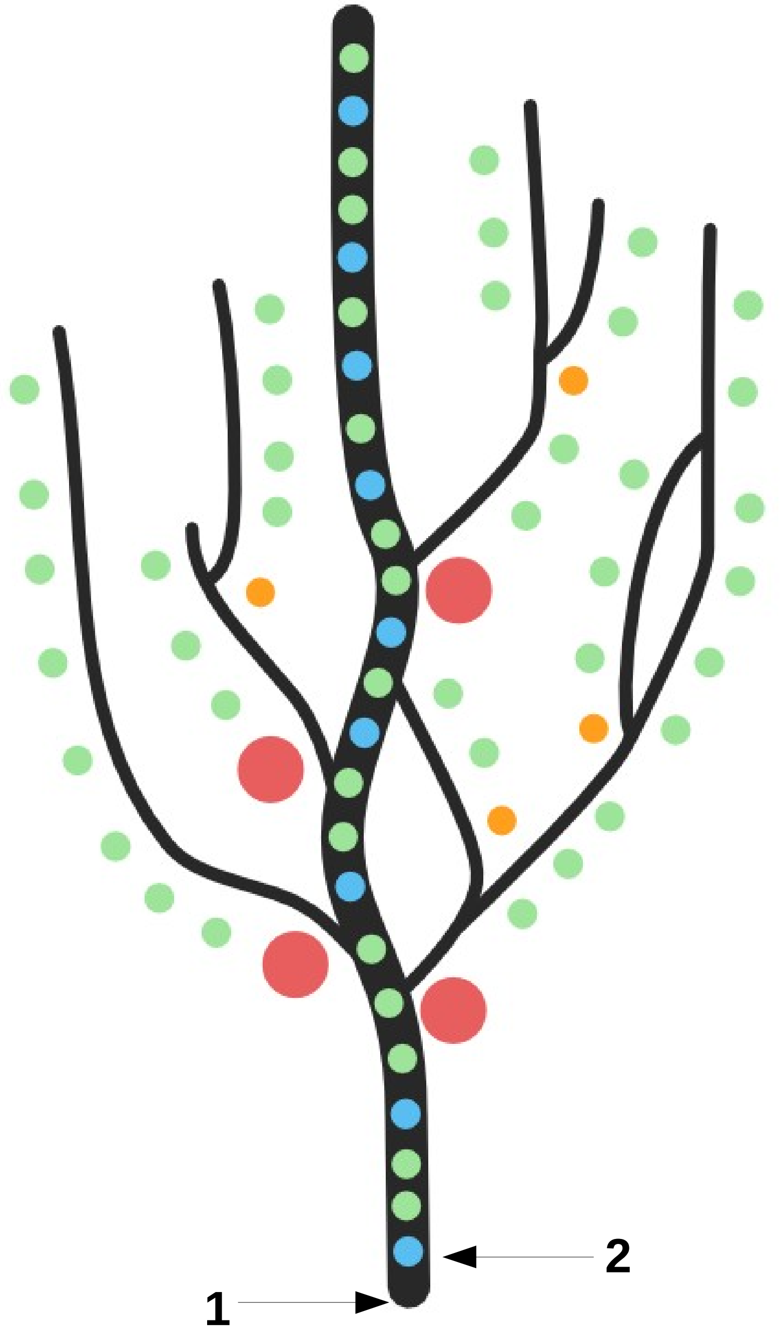
1) Pick a project contained in one directory
   (or create a new directory with the oldest files in it).



Git: How to get started?

1) Pick a project contained in one directory
   (or create a new directory with the oldest files in it).

2) Commit the oldest files without any date/time/version indicators to initialize the history in git.

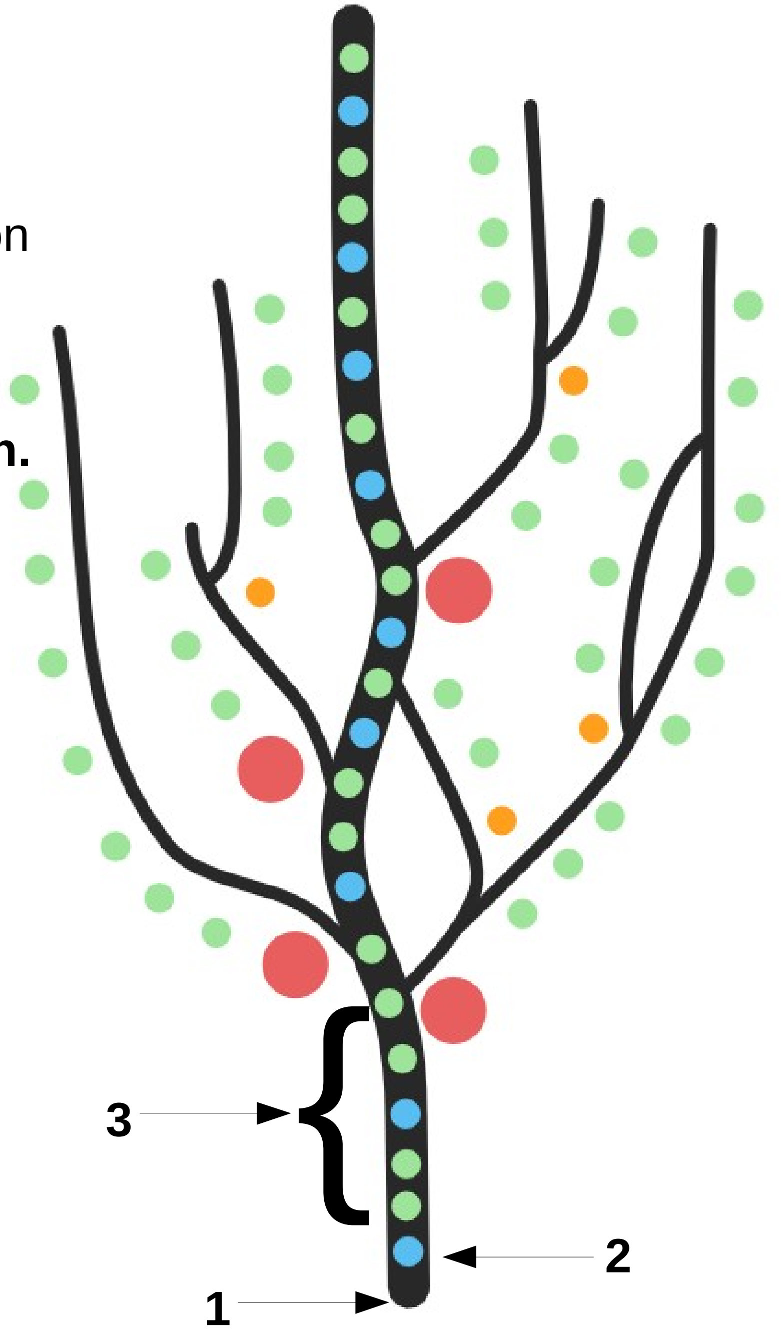


Git: How to get started?

1) Pick a project contained in one directory
(or create a new directory with the oldest files in it).

2) Commit the oldest files without any date/time/version indicators to initialize the history in git.

3) **Overwrite the files with the more recent ones, in chronological order, and commit each version.**



Git: How to get started?

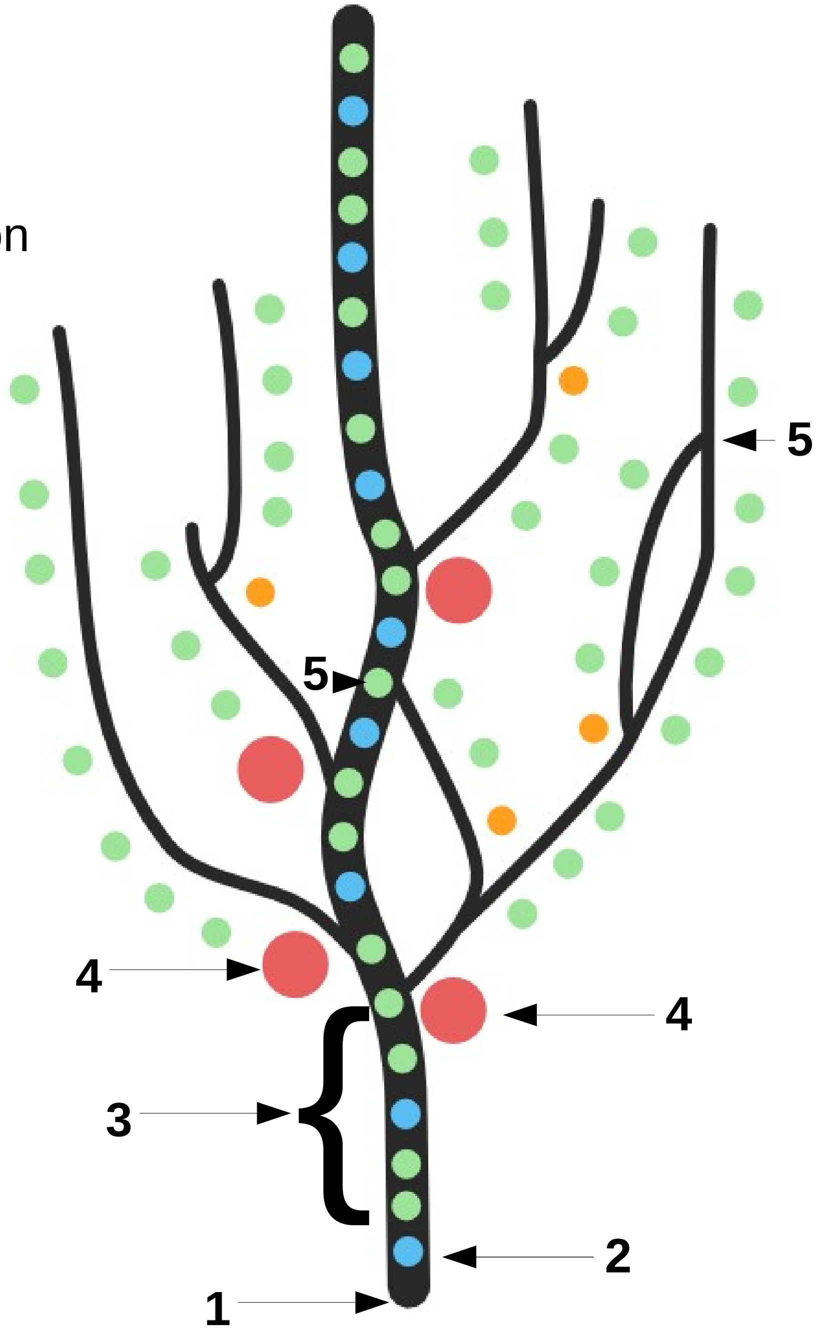
1) Pick a project contained in one directory
   (or create a new directory with the oldest files in it).

2) Commit the oldest files without any date/time/version
   indicators to initialize the history in git.

3) Overwrite the files with the more recent ones,
   in chronological order, and commit each version.

4) Start new branches to work on new features.

**5) Merge back branches when the features are
   completed, to update the main/master branch.**



Git: A specific example

1) Start working on a new paper/thesis/thesis proposal

2) Commit the files to be tracked:
   - tex file of the proper document
   - scripts specific to figures for the document

3) Commit new "milestones" as progress is made towards the first full draft.

4) Start new branches to work with co-authors revisions.

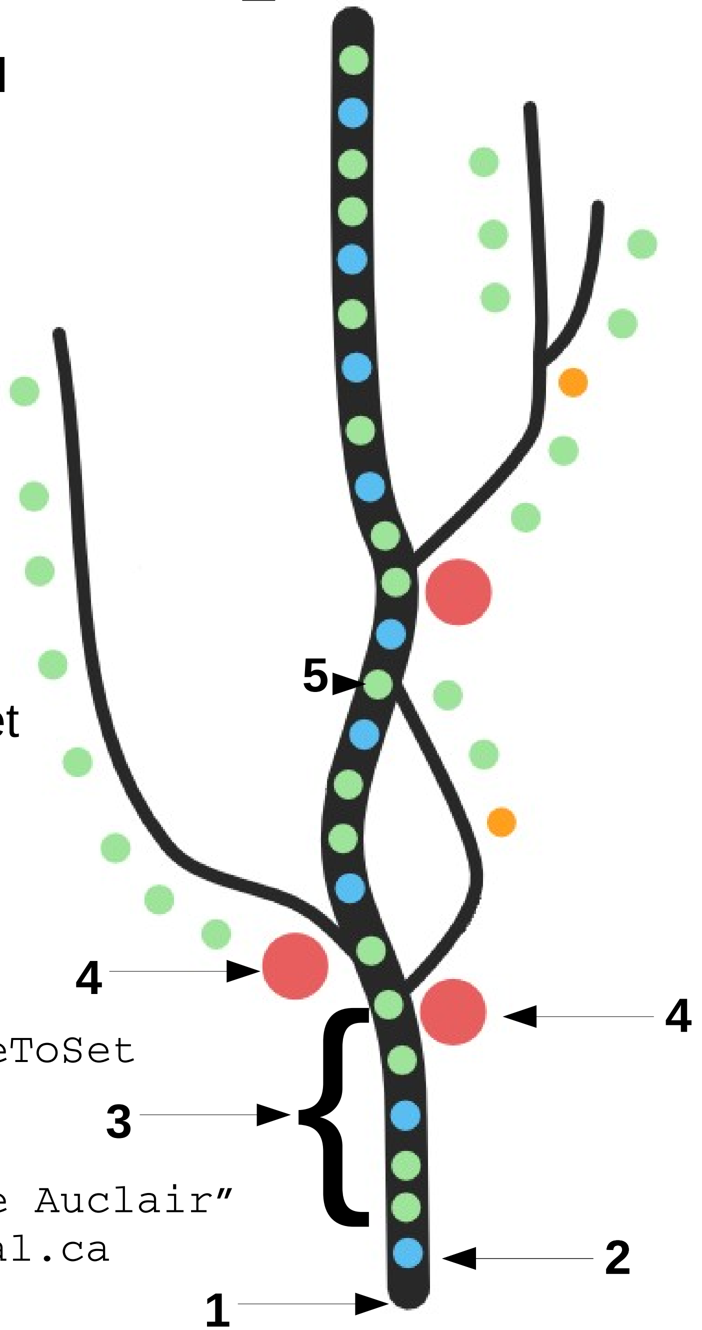
5) Merge comments back into the master branch to get the new version.

1*) If this is your first ever git repository, you should also configure git parameters:

```
git config --global PropertyToChange ValueToSet
```

ex.:

```
git config --global user.name "Jean-Pierre Auclair"
git config --global user.email jn402157@dal.ca
```



Git: The actual commands

1) Start working on a new paper/thesis/thesis proposal

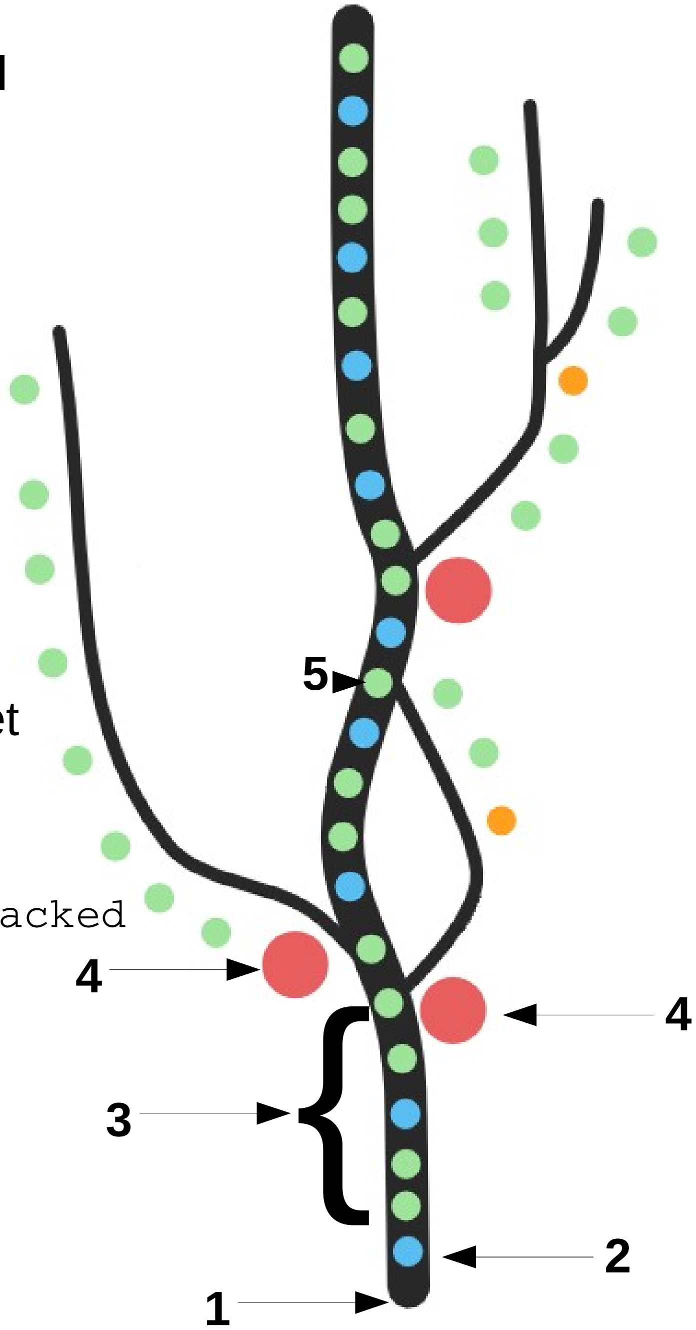
2) Commit the files to be tracked:
   - tex file of the proper document
   - scripts specific to figures for the document

3) Commit new “milestones” as progress is made towards the first full draft.

4) Start new branches to work with co-authors revisions.

5) Merge comments back into the master branch to get the new version.

```
1) git init
2) git add FirstFileToBeTracked ScndFileToBeTracked
2) git commit -m “First commit of ThisProject”
3) git add FileThatHasChanged NewFileToTrack
3) git commit -m “Changed XYZ”
4) git branch NewBranch
4) git checkout NewBranch
5) git checkout master
5) git merge NewBranch
```



Git: The actual commands +

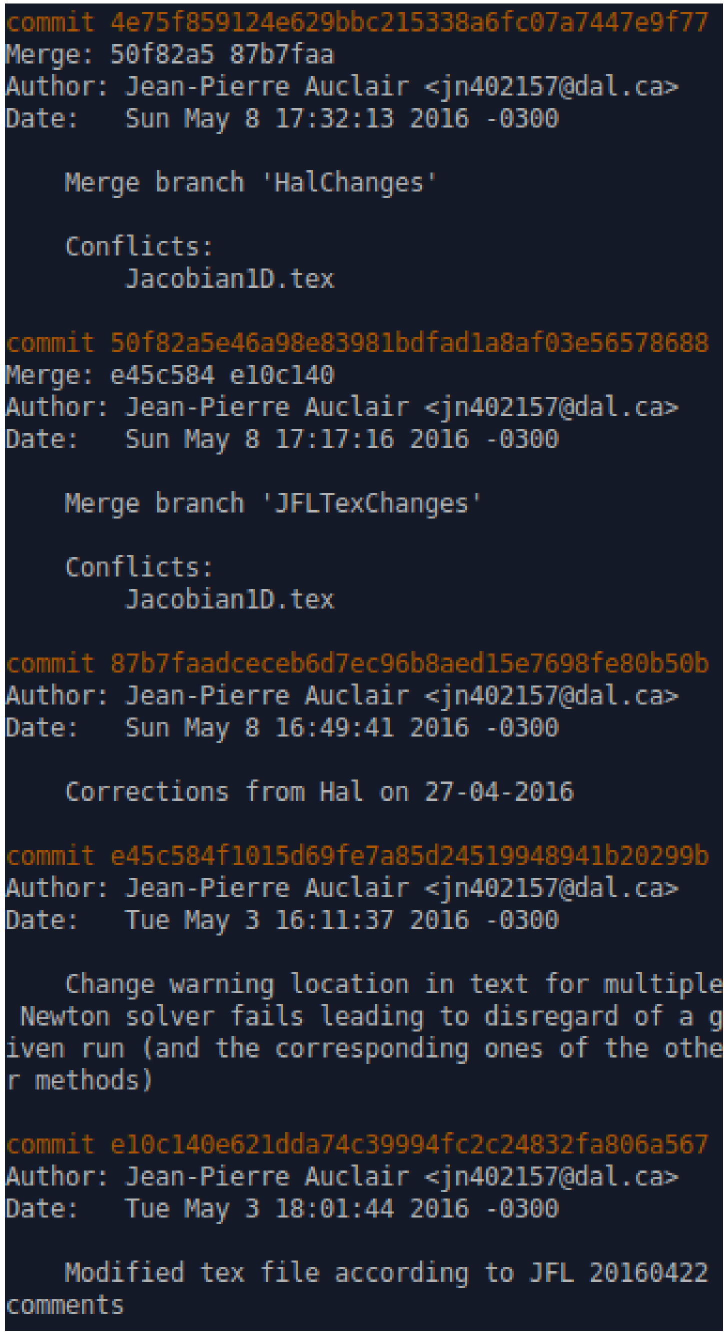
1) Looking where you are:
```
git status
```

2) Looking back in history:
```
git log
```

3) Comparing commits:
```
git diff BaseCommit NewCommit
```

4) Fixing a commit:
```
git commit --amend
```

5) Tagging a commit:
```
git tag TagToBeApplied
git tag -a TagToBeApplied -m TagMessage
```



```
commit 4e75f859124e629bbc215338a6fc07a7447e9f77
Merge: 50f82a5 87b7faa
Author: Jean-Pierre Auclair <jn402157@dal.ca>
Date:   Sun May 8 17:32:13 2016 -0300

    Merge branch 'HalChanges'

    Conflicts:
        Jacobian1D.tex

commit 50f82a5e46a98e83981bdfad1a8af03e56578688
Merge: e45c584 e10c140
Author: Jean-Pierre Auclair <jn402157@dal.ca>
Date:   Sun May 8 17:17:16 2016 -0300

    Merge branch 'JFLTexChanges'

    Conflicts:
        Jacobian1D.tex

commit 87b7faadceceb6d7ec96b8aed15e7698fe80b50b
Author: Jean-Pierre Auclair <jn402157@dal.ca>
Date:   Sun May 8 16:49:41 2016 -0300

    Corrections from Hal on 27-04-2016

commit e45c584f1015d69fe7a85d24519948941b20299b
Author: Jean-Pierre Auclair <jn402157@dal.ca>
Date:   Tue May 3 16:11:37 2016 -0300

    Change warning location in text for multiple
 Newton solver fails leading to disregard of a g
iven run (and the corresponding ones of the othe
r methods)

commit e10c140e621dda74c39994fc2c24832fa806a567
Author: Jean-Pierre Auclair <jn402157@dal.ca>
Date:   Tue May 3 18:01:44 2016 -0300

    Modified tex file according to JFL 20160422
comments
```

Git: The actual commands ++

1) Looking where you are:
```
git status
```

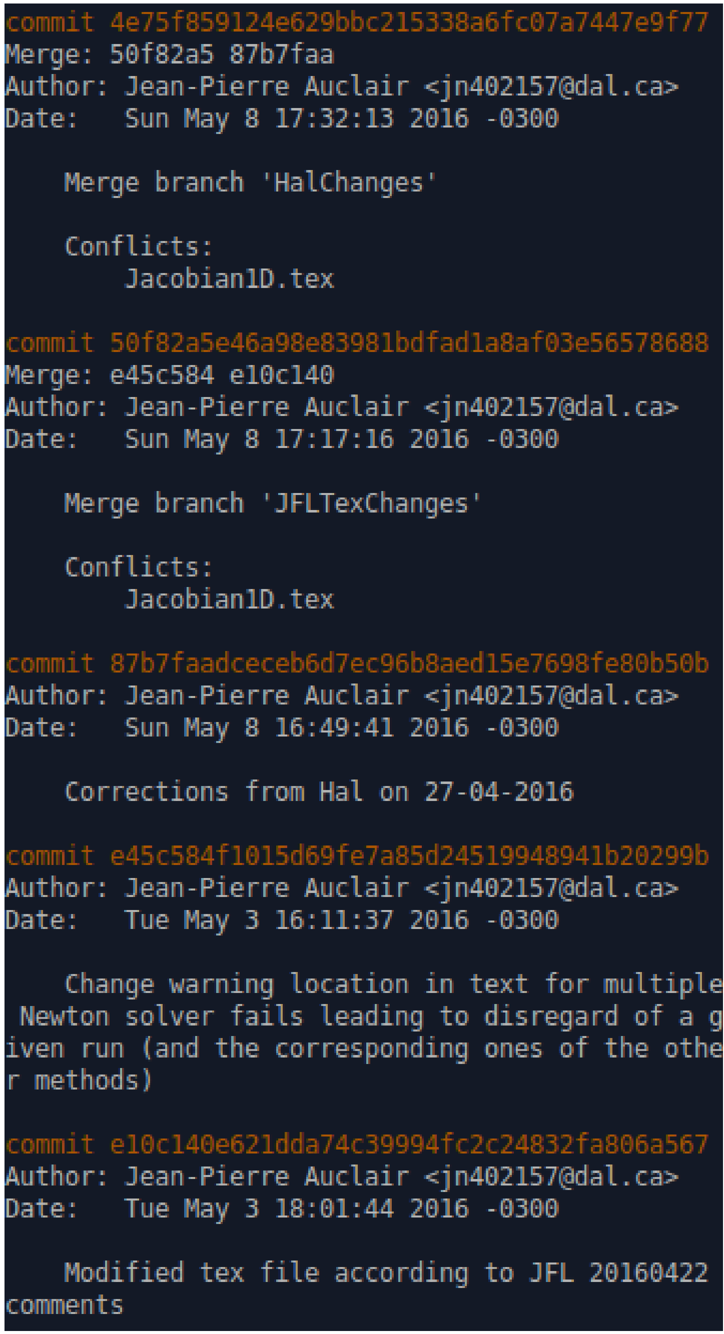
2) Looking back in history:
```
git log
```

3) Going back in time:
```
git checkout OldCommit
```

4) Starting an alternate history:
```
git checkout -b NewBranchName
```

5) Coming back to the present:
```
git checkout master
```

6) Deleting history:
```
git reset --hard CommitToRevertTo
```



```
commit 4e75f859124e629bbc215338a6fc07a7447e9f77
Merge: 50f82a5 87b7faa
Author: Jean-Pierre Auclair <jn402157@dal.ca>
Date:   Sun May 8 17:32:13 2016 -0300

    Merge branch 'HalChanges'

    Conflicts:
        Jacobian1D.tex

commit 50f82a5e46a98e83981bdfad1a8af03e56578688
Merge: e45c584 e10c140
Author: Jean-Pierre Auclair <jn402157@dal.ca>
Date:   Sun May 8 17:17:16 2016 -0300

    Merge branch 'JFLTexChanges'

    Conflicts:
        Jacobian1D.tex

commit 87b7faadceceb6d7ec96b8aed15e7698fe80b50b
Author: Jean-Pierre Auclair <jn402157@dal.ca>
Date:   Sun May 8 16:49:41 2016 -0300

    Corrections from Hal on 27-04-2016

commit e45c584f1015d69fe7a85d24519948941b20299b
Author: Jean-Pierre Auclair <jn402157@dal.ca>
Date:   Tue May 3 16:11:37 2016 -0300

    Change warning location in text for multiple
 Newton solver fails leading to disregard of a g
iven run (and the corresponding ones of the othe
r methods)

commit e10c140e621dda74c39994fc2c24832fa806a567
Author: Jean-Pierre Auclair <jn402157@dal.ca>
Date:   Tue May 3 18:01:44 2016 -0300

    Modified tex file according to JFL 20160422
comments
```

Git: Extra resources:

1) Try-git: https://try.github.io/levels/1/challenges/1
2) Online git cheatsheets
3) Asking Google
4) My (VERY WIP) wiki: http://www.phys.ocean.dal.ca/~jpaucl/index.php/Main/Git
5) Your git using peers!

Git: Extra resources:

1) Try-git: https://try.github.io/levels/1/challenges/1
2) Online git cheatsheets
3) Asking Google
4) My (VERY WIP) wiki:
   http://www.phys.ocean.dal.ca/~jpaucl/index.php/Main/Git
5) Your git using peers
6) **Me right now!**

Questions?

git?!?