GIT OPEN SPACE

CJ SVENINGSSON, NAIM LATIFI, ERIK LUNDGREN
2015-10-06

CM why and what - Intro

“Now, up to now my plan went all right
'Til we tried to put it all together one night
And that's when we noticed that something was definitely wrong
The transmission was a '53 and the motor turned out to be a '73
And when we tried to put in the bolts all the holes were gone
So we drilled it out so that it would fit
And with a little bit of help with an adapter kit
We had that engine runnin' just like a song
...
You might say I went right up to the factory
And picked it up, it's cheaper that way
Uh, what model is it?
Well, it's a '49, '50, '51, '52, '53, '54, '55, '56 '57, '58' 59' automobile
It's a '60, '61, '62, '63, '64, '65, '66, '67 '68, '69, '70 automobile”

"One Piece at a Time" April 1976
"One Piece at a Time" April 1976

https://www.youtube.com/watch?v=rWHniL8MyMM

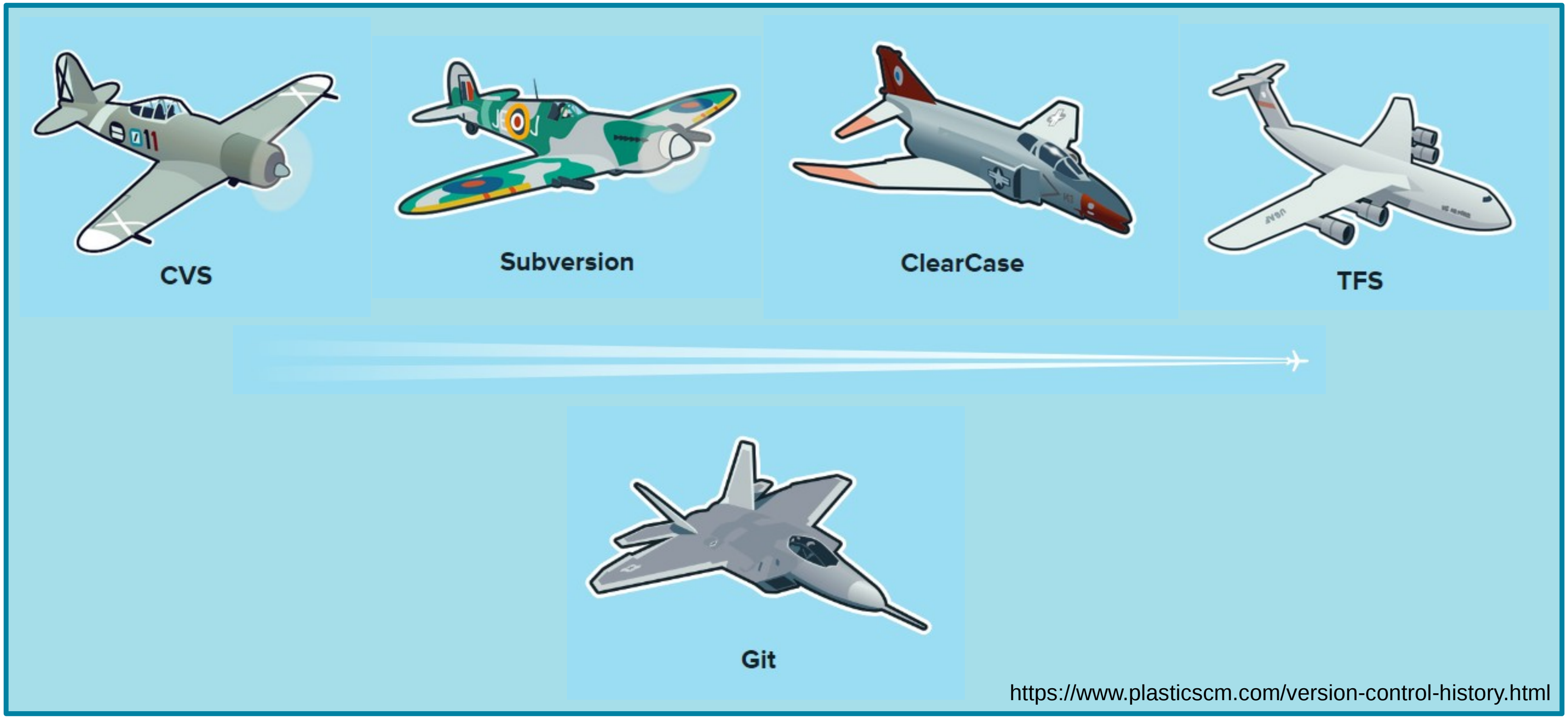
"One Piece At A Time“, Johnny Cash (1976)

http://en.wikipedia.org/wiki/One_Piece_at_a_Time

CM why and what - Definitions

- Configuration Management
- Change Management (ITIL)
- Software Configuration Management
- Revision Control / Version Control / Source Control
- Distributed Revision (/Version) Control System
- *List of revision control software*

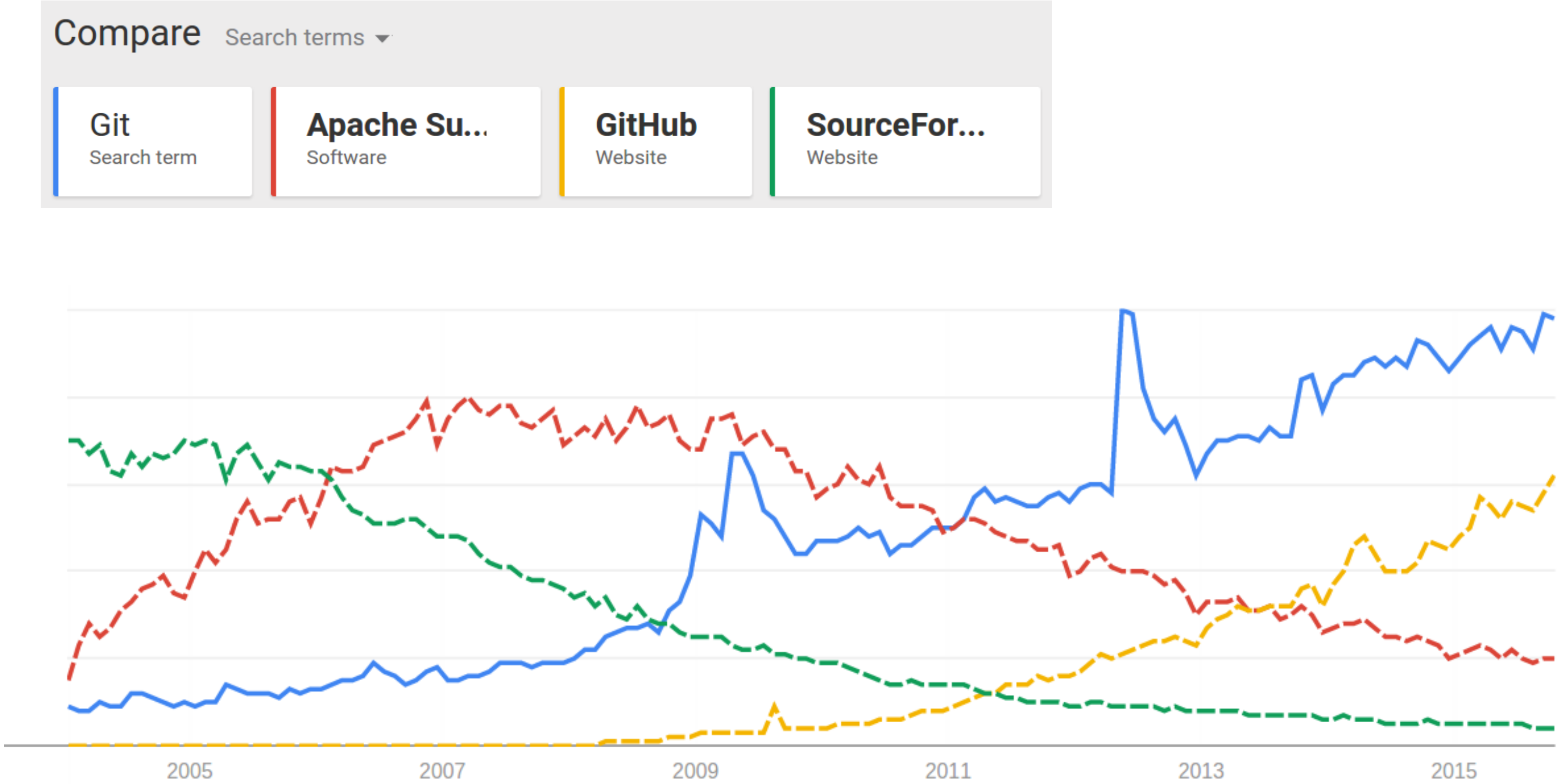
Git is a 5th generation version control system



https://www.plasticscm.com/version-control-history.html

- There were many before Git
- Git was the first to be fast, reliable, flexible and popular with developers (?)
- Git is not a solution in every situation
- Git is not the final solution

Why we talk about Git and Github today



Compare Search terms

Git
Search term

Apache Su...
Software

GitHub
Website

SourceFor...
Website

http://www.google.com/trends/explore#q=Git%2C%20%2Fm%2F012ct9%2C%20%2Fm%2F04g0kcw%2C%20%2Fm%2F02p0tdp&cmpt=q&tz=Etc%2FGMT-2

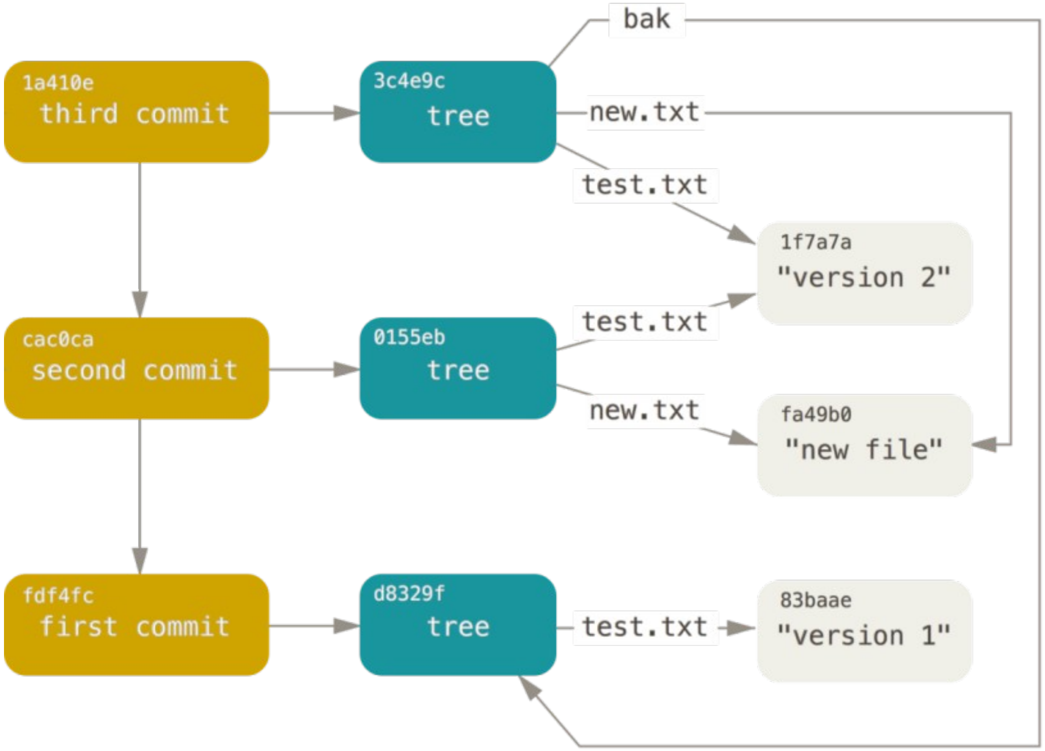
What Git is

- A distributed version control system
- Created in 2005 by Linus Torvalds to move the Linux kernel development away from BitKeeper
- Linus promptly calls Subversion ”the most pointless project ever started” (for wanting to ”do cvs right”)
- GitHub becomes the world’s most popular site for open source collaboration

GIT commits

IMPORTANT!

- A repository is a cryptographically (SHA-1) **directed acyclic graph** of **commits**, **trees** (directories) and **blobs** (files)
- Each commit reflects the entire workspace
- Git enables you to do CM locally (and collaborate however you want)
- *"It is easy to shoot your foot off with git…*
- *…but also easy to revert to a previous foot and merge it with your current leg.“*



http://git-scm.com/book/en/v2/Git-Internals-Git-Objects

The Ultimate Goal

Developer : Product == 1:1

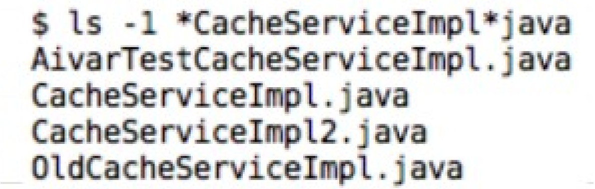
… but our dealings are complex

- Review
- Testing
- Static program analysis
- Release
- Experimental development
- Integration

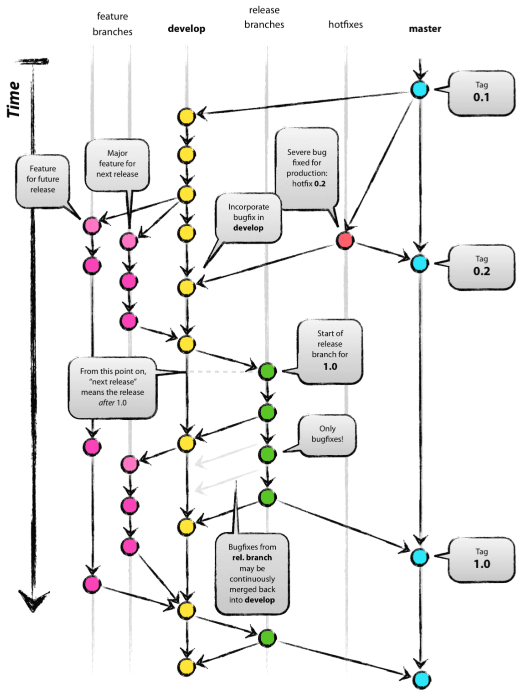
- Build quality in
- Issue tracking
- Update / Hotfix / Service pack
- Continuous integration
- Continuous delivery / DevOps

The Ultimate Goal – Build quality in

- You can have experiments and cruft which you may one day want, but keep it out of the product
- Keep pruduct a pristine environment
- Good CM is essential to enable the lean principles
- Stop the line so the line never stops (and keep the floors clean)
- Branching strategy is usually the first thing teams realize they lack
- But if you can manage the CM, it has proven very powerful, fast and flexible for many



```
$ ls -1 *CacheServiceImpl*java
AivarTestCacheServiceImpl.java
CacheServiceImpl.java
CacheServiceImpl2.java
OldCacheServiceImpl.java
```

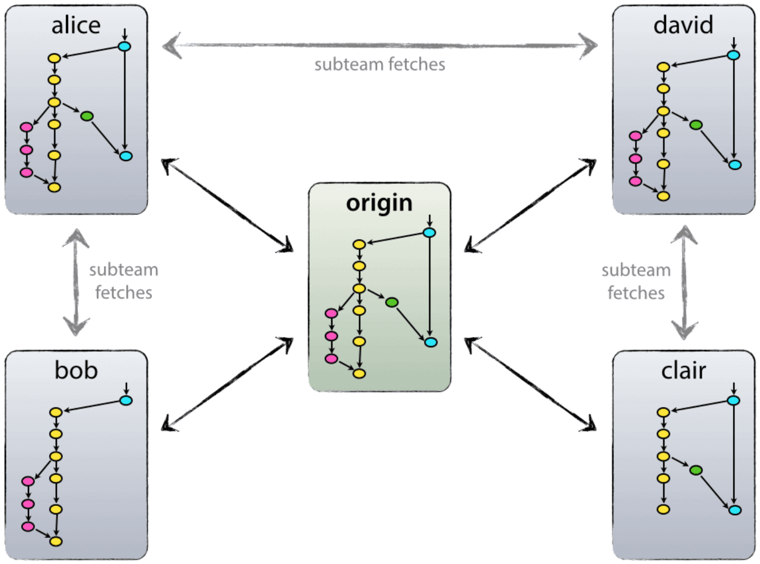

feature branches
develop
release branches
hotfixes
master
Time
Tag 0.1
Major feature for next release
Feature for future release
Severe bug fixed for production: hotfix 0.2
Incorporate bugfix in develop
Tag 0.2
Start of release branch for 1.0
From this point on, "next release" means the release after 1.0
Only bugfixes!
Bugfixes from rel. branch may be continuously merged back into develop
Tag 1.0

http://nvie.com/posts/a-successful-git-branching-model/

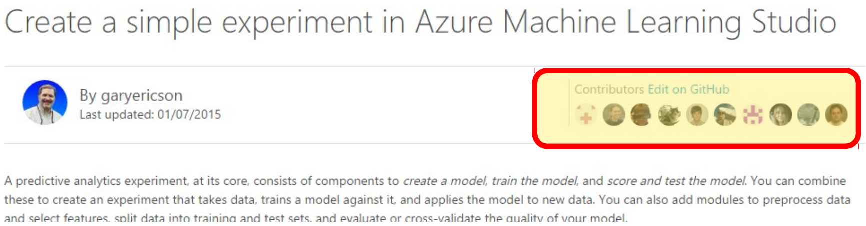
https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow

How to collaborate with Git

- Git doesn’t need the network
- But there are many ways to collaborate:
  - Patches / bundles
  - SSH / network share
  - Gitosis server
  - Github / Bitbucket / other



- The focus to do only CM and do it right and reliably has contributed to the popularity of Git and Github: http://www.wired.com/2012/02/github-2/all/



Create a simple experiment in Azure Machine Learning Studio

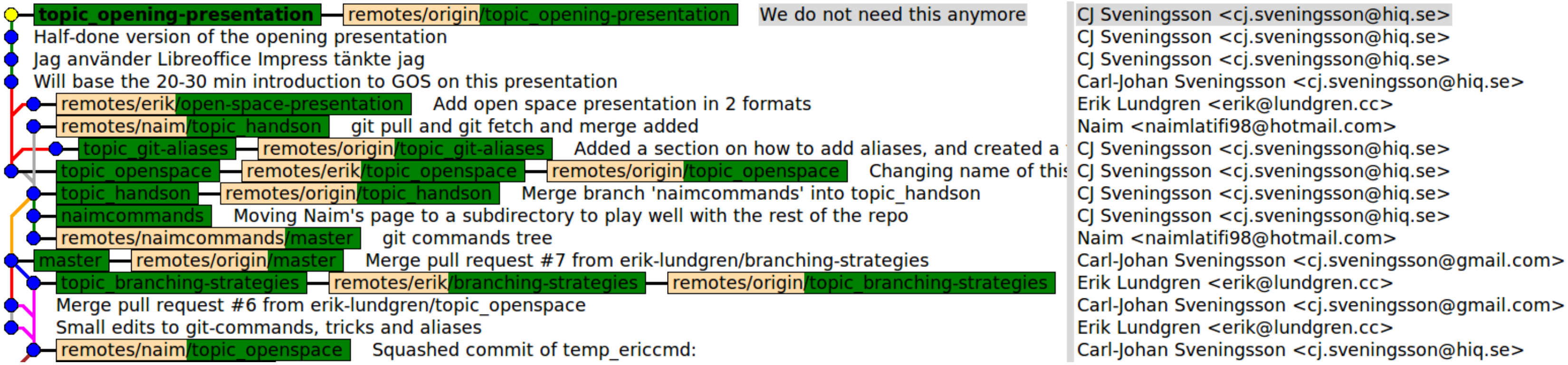
By garyericson
Last updated: 01/07/2015

Contributors Edit on GitHub

A predictive analytics experiment, at its core, consists of components to *create a model, train the model,* and *score and test the model.* You can combine these to create an experiment that takes data, trains a model against it, and applies the model to new data. You can also add modules to preprocess data and select features, split data into training and test sets, and evaluate or cross-validate the quality of your model.

https://github.com/Azure/azure-content

Children of the shoe-maker...



```
topic_opening-presentation  remotes/origin/topic_opening-presentation  We do not need this anymore         CJ Sveningsson <cj.sveningsson@hiq.se>
Half-done version of the opening presentation                                                          CJ Sveningsson <cj.sveningsson@hiq.se>
Jag använder Libreoffice Impress tänkte jag                                                            CJ Sveningsson <cj.sveningsson@hiq.se>
Will base the 20-30 min introduction to GOS on this presentation                                       Carl-Johan Sveningsson <cj.sveningsson@hiq.se>
remotes/erik/open-space-presentation  Add open space presentation in 2 formats                         Erik Lundgren <erik@lundgren.cc>
remotes/naim/topic_handson  git pull and git fetch and merge added                                     Naim <naimlatifi98@hotmail.com>
topic_git-aliases  remotes/origin/topic_git-aliases  Added a section on how to add aliases, and created a  CJ Sveningsson <cj.sveningsson@hiq.se>
topic_openspace  remotes/erik/topic_openspace  remotes/origin/topic_openspace  Changing name of this     CJ Sveningsson <cj.sveningsson@hiq.se>
topic_handson  remotes/origin/topic_handson  Merge branch 'naimcommands' into topic_handson             CJ Sveningsson <cj.sveningsson@hiq.se>
naimcommands  Moving Naim's page to a subdirectory to play well with the rest of the repo               CJ Sveningsson <cj.sveningsson@hiq.se>
remotes/naimcommands/master  git commands tree                                                         Naim <naimlatifi98@hotmail.com>
master  remotes/origin/master  Merge pull request #7 from erik-lundgren/branching-strategies           Carl-Johan Sveningsson <cj.sveningsson@gmail.com>
topic_branching-strategies  remotes/erik/branching-strategies  remotes/origin/topic_branching-strategies  Erik Lundgren <erik@lundgren.cc>
Merge pull request #6 from erik-lundgren/topic_openspace                                               Carl-Johan Sveningsson <cj.sveningsson@gmail.com>
Small edits to git-commands, tricks and aliases                                                        Erik Lundgren <erik@lundgren.cc>
remotes/naim/topic_openspace  Squashed commit of temp_ericcmd:                                         Carl-Johan Sveningsson <cj.sveningsson@hiq.se>
```

- Sometimes it's hard to practice what you preach...
- … but with Git, we make it work.
- Looking forward to your contributions!

WHAT ABOUT GIT AT HIQ?

- We don't need no internal infrastructure (?!)
- `https://github.com/orgs/HiQStockholmAB`
- … and some in Bitbucket?
- The competence exist, but is spread out, until we gather it

Things related to Git

- [**Github**](https://github.com/) (probably most popular public hosting)
- [**BitBucket**](https://bitbucket.org/) (another place where you can do private hosting)
- [**Gerrit**](http://code.google.com/p/gerrit/) (code review)
- [**Gerrithub**](http://gerrithub.io/) (Gerrit+Github)
- [**Github pages**](https://pages.github.com/) (publish web pages straight from your repo)
- [**Github dotfiles**](https://dotfiles.github.io/) (save your preferences and share them)
- [**Github gist**](https://gist.github.com/) (CM'ed pastebin service - CM everything!)
- [**Gitorious server**](https://gitorious.org/) (private repository server)
- [**Semantic Merge**](http://www.semanticmerge.com/) (semantically aware merge/diff tool by the authors of Plastic SCM)
- [**Araxis Merge**](http://www.araxis.com/merge/) (Merge office docs etc, Windows & OS X)
- [**Deltawalker**](http://www.deltopia.com/compare-merge-sync/macosx/) ([et al...](http://www.git-tower.com/blog/diff-tools-mac/))
- [**Travis CI**](https://travis-ci.com/), [**BuildHive**](http://en.wikipedia.org/wiki/BuildHive), **Jenkins** (continuous integration servers)
- [**Puppet**](http://en.wikipedia.org/wiki/Puppet_\(software\)) ("Conventional" CM server software)
- [**Jira**](http://en.wikipedia.org/wiki/JIRA) ([Issue tracking](http://en.wikipedia.org/wiki/Comparison_of_issue-tracking_systems) server)

How to do open space?

thiQ

Open space – How does it work

The law of two feet:

- Foot of passion
- Foot of responsibility

Four principles:

- Whoever comes is the right people
- Whatever happens is the only thing that could have
- Whenever it starts is the right time
- When it's over, it's over

Prepare to be surprised!

Open space – Schedule

- **Opening (we did this already)**
- **Marketplace of ideas**
  - Suggest branch
  - Choose time slot
  - Explain branch
- **Break out sessions**
  - Choose your adventure (you can choose multiple adventures!)
  - Utilize the Law of two feet
  - Contribute (to the repository or otherwise)
- **Closing (20:45)**
  - Fill in feedback papers

THANK YOU FOR LISTENING!

Fork the repo!

`https://github.com/UncleCJ/hicollegegit`

LET’S GO G.O.S