

# Version Control Systems, Documentation Management & Helpdesk

```
      - -  
    .-. | | |  
    |M|_|A|N|  
    |A|a|.|.|\<  
    |T|r| | | \\  
    |H|t|M|Z|  \\  
    |!| | | |  \>  "Bookshelf" by  
                      David S. Issel  
*****
```

System and Network Administration

Revision 2 (2020/21)

# Table of contents

- ▶ VCS History
- ▶ VCS Operations
- ▶ Hosting VCS UI
- ▶ Documentation Management
- ▶ Bug-Tracking & Helpdesk Systems

# VCS History

*What is it good for?...*

- ▶ dealing with large amount of code
- ▶ dealing with large development team
- ▶ trace back issues and incriminating commits

what content to version control?

- ▶ for code, obviously
- ▶ documentation (Markdown, RST)
- ▶ web design (CSS)
- ▶ (anything text-based)

even binaries with Git Large File Storage (LFS)

-> pointers instead of blobs

## Two bits of history...

- ▶ 1st gen: pessimistic
- ▶ 2nd gen: optimistic
- ▶ 3rd gen: distributed

# Pessimistic

- ▶ Source Code Control System (SCCS)
- ▶ GNU Revision Control System (RCS)

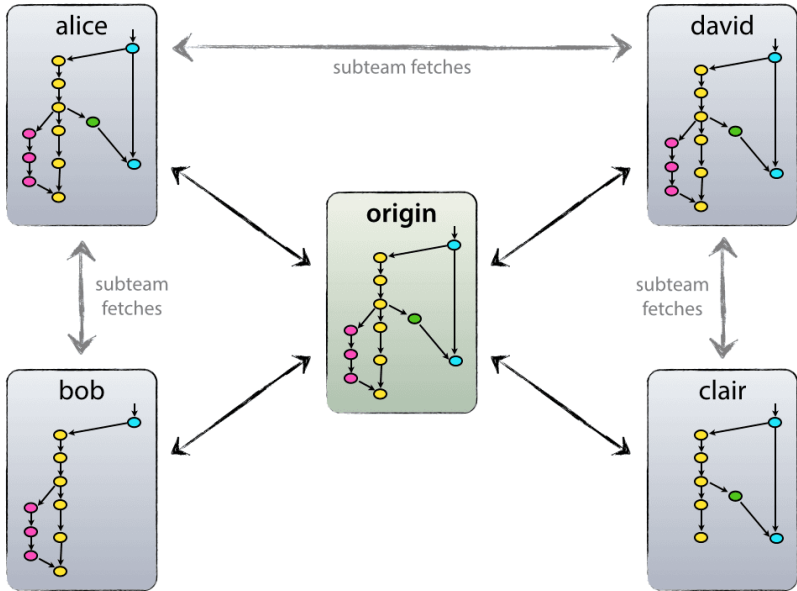
-> file locks

# Optimistic

- ▶ Concurrent Versions System (**CVS**)
- ▶ Subversion (**SVN**)
- ▶ Helix Core (**Perforce**)

-> merging when no conflicts

-> fix conflicts manually





# Distributed

- ▶ BitKeeper (open-sourced 2016)
- ▶ Darcs
- ▶ Mercurial
- ▶ **GIT**
- ▶ GNU Bazaar

-> even branches can merge...

*How to fix conflicts?...*

==> you get to edit the conflicting changes manually during commits (very similarly as with CVS)

## CVS in theory

- ▶ local *copy* of remote repository
- ▶ remote is kind of bare...
- ▶ commit == push your change to the online and centralized repository
- ▶ no staging
- ▶ use tag (branch) to track a (rather large) bunch of changes together
- ▶ centralized but highly-available (mirrors)
- ▶ current/dev == trunk (no branch)

## GIT in theory

- ▶ local repository
- ▶ staging
- ▶ commit locally (implicit tag)
- ▶ push to remote
- ▶ commits are more frequent than CVS
- ▶ current/dev == master branch

## BitKeeper & Linux History

- ▶ Linus Torvalds (really not a zealot)
- ▶ Richard Stallman, Alan Cox (бородач)
- ▶ Andrew Tridgell (communication protocol nerd who did not agree on the license)

## ESR's book makes sense

- ▶ the cathedral vs the bazar
- ▶ → CVS (BSDs) vs Linux (GIT)

# VCS Operations

(\  
\'\  
\'\  
/ '| ( ) \_\_\_\_\_)  
\' / \ ~~~~~ \  
\  
==) . \ \_\_\_\_\_ \  
( ) ( ) \_\_\_\_\_)

# CVS operations

```
export CVSROOT="anoncvs@anoncvs.MIRROR.netbsd.org:/cvsroot"
```

```
export CVS_RSH="ssh"
```

```
cvs checkout -P src
```

```
##-r netbsd-9
```

```
cd src/
```

```
cvs update -dP
```



## stage and commit

```
cv$ add new-file
```

```
cv$ commit
```

**revisions**

`cv$ log`

**just the headers**

`cv$ log -h`

## Create CVS repo

### create the remote repository

```
cvsc -d ~/cvsroot init
export CVSROOT=~/cvsroot
cvsc import -m "" project project initial
```

- ▶ first argument: remote folder name for the repository
- ▶ second argument: vendor tag, define your organization
- ▶ third argument: release tag, `initial`, `current` or `v0.0`

### create the cvs tracking within the existing folder

```
cvsc checkout -P project
```

# GIT operations

```
git clone <source> <dest>
```

```
https:// -- need to login for read-write access...
```

```
git://
```

```
ssh:// -- much nicer with authorized keys
```

## stage and commit

```
git pull
```

```
git add new-file
```

```
git commit
```

```
git push
```

## staging diffs

```
git add ...
```

```
git commit
```

```
git push
```

staged diff

see what has changed

```
git status
```

see what's been staged for commit already

```
git diff --cached --stat
```

```
git diff --cached
```

```
git status -v
```

# Review commits

see the changed files

```
git log
```

```
git show COMMIT --stat
```

see all the diffs

```
#git log -p
```

```
git show COMMIT
```



# Create GIT repo

*local vs. hosting*

remote clients cannot push

```
cd project/  
git init  
git add file1 file2  
git commit -m initial
```

remote clients can push

```
cd project.git/  
git init --bare
```

can convert bare to normal and *vice versa*

# GIT in production

update the list of branches and switch to the new revision

```
git fetch -a
git checkout release-1.1.0
git pull
```

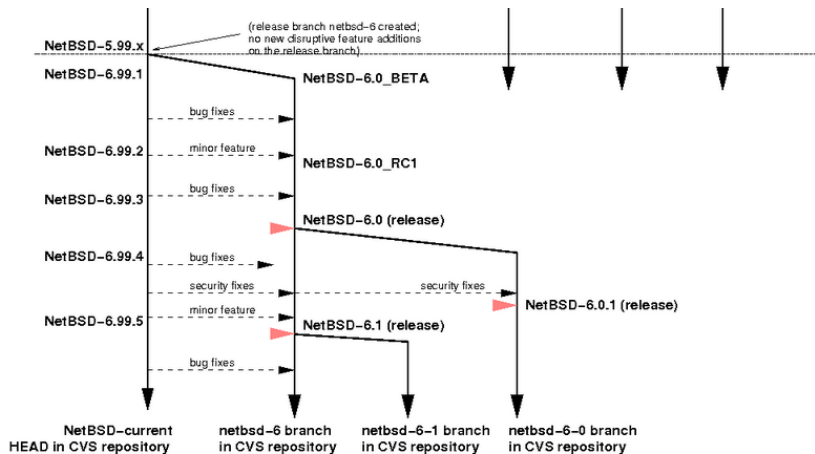
and restart the application

**WARNING DO NOT SHARE THE `.git/` FOLDER ON THE WEB!**

(\  
 \ ' \  
 \ ' \  
 / ' | ( ) \_\_\_\_\_  
 \ ' / \ ~~~~~ \  
 \ \ ~~~~~ \  
 ==) . \ \_\_\_\_\_ \  
 ( \_ ) ( ) \_\_\_\_\_

*// Questions on VCS operations?*

# CVS branches



main (HEAD) branch

stable (release) branch

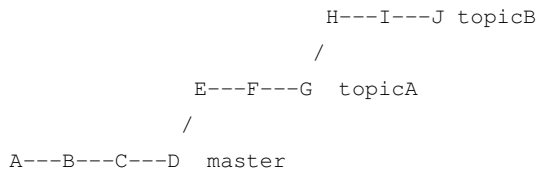
security/critical branches

Legend:

--▶ Pullups

▶ Release tag

# GIT branches



**new branch** myfeature

```
git checkout -b myfeature develop
```

**merge back to develop non-fast-forward**

```
git checkout develop
```

```
git merge --no-ff myfeature
```

```
git branch -d myfeature
```

```
git push origin develop
```

The other guy says linear commits are better: helps resolve code dependency issues

Anyway he advertises

```
git rebase --interactive
```

## GIT branching models

*A successful Git branching model*, Vincent Driessen, 2010

*A succesful Git branching model considered harmful*, Jussi Judin, 2016

-> Trunk based development (<https://trunkbaseddevelopment.com/>)



## Gitflow workflow

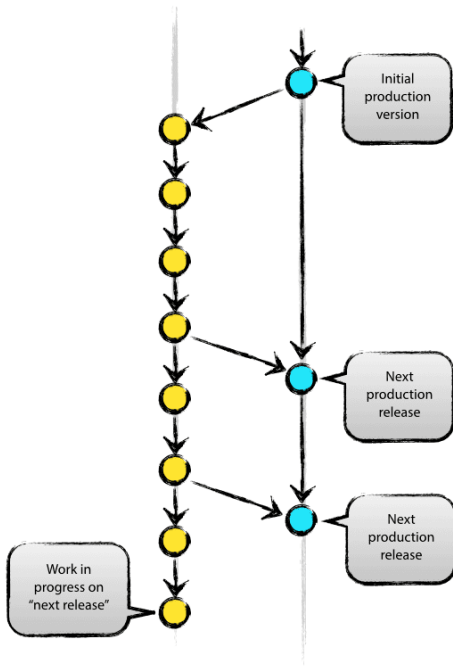
- ▶ usual production is based on release tarballs
- ▶ but GIT repo can be used as well
- ▶ no need to select a branch for production

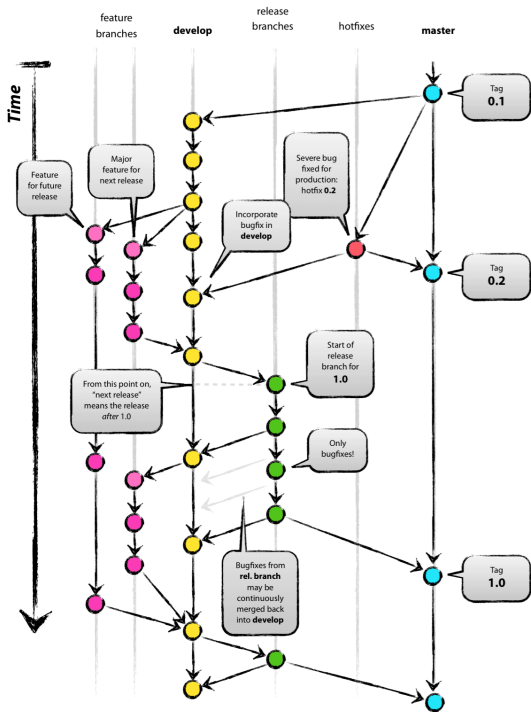
## Trunk-based workflow

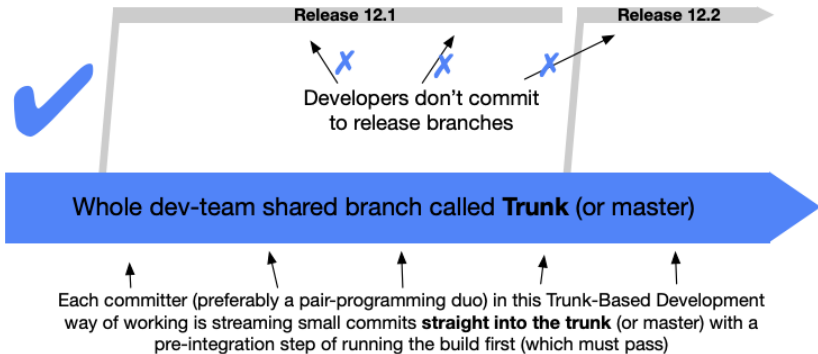
- ▶ makes more sense for developers who want to contribute
- ▶ same for sysadmins who need bleeding-edge features
- ▶ the thing they fetch is development/current

develop

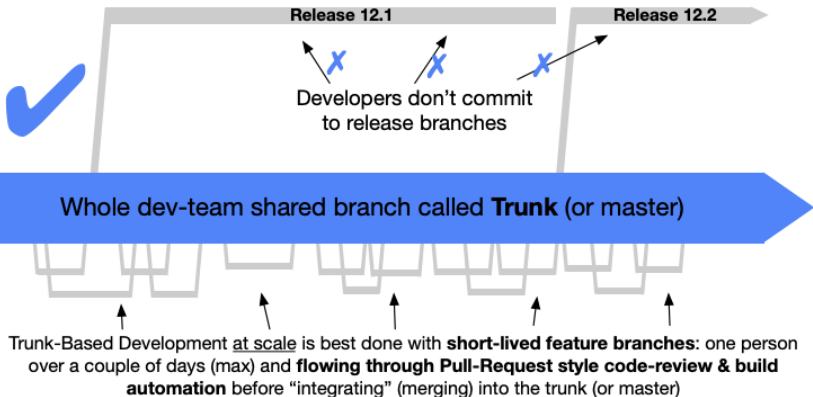
master







`master` for development // [trunkbaseddevelopment.com](http://trunkbaseddevelopment.com)



// trunkbaseddevelopment.com

## Merging method recommended practice

- ▶ do not `git-merge` (join two or more development histories together)
- ▶ but `git-rebase` (re-apply commits on top of another base tip)

Seems to be nice with Gerrit...

## Best practices by Aleksandr

1. Use master/develop branches, with feature-based (or sometimes even task-based) branching for develop because it gives you better tracking of your tasks. (And also short branches make merging less painful)

2. Use same format for commit messages. If you violate this rule or dont have rule at all, your colleagues will not understand what you have just change if there is a lot of code in your repo. I try use like

```
git commit -m "ADD: some feature; UPD: optimized  
algorithm; FIX: some bug; DEL: file I dont need"
```



3. If your version control tool allows to do that, configure CI instructions for pull requests to both: master and develop branch. Because your system will probably be deployed on server for prod and servers for test

4. If you make a change in architecture, make a separate branch with name which differs from your feature-based branches

5. Less branches = good

# Hosting VCS UI

*online hosting vs. on-premises*

## VCS “In the cloud”

- ▶ easy public & open-source repositories (read-only for everyone)
- ▶ *server-less* hosting
- ▶ Public/ro access (`https://` & `git://`)
- ▶ Commit/diff visualization UI
- ▶ Bug/issue tracker
- ▶ Markdown/HTML renderer
- ▶ Wiki
- ▶ Release tarballs
- ▶ (Social network)
- ▶ No advertisements

# VCS hosting market

- ▶ Github
- ▶ Gitlab.com – got source
- ▶ Atlassian Bitbucket (Mercurial, GIT)
- ▶ Atlassian FishEye (CVS, SVN, Mercurial, GIT, Perforce)
- ▶ Savannah (CVS, GIT, GNU Bazaar)
  - ▶ GNU
  - ▶ Non-GNU
- ▶ gnumonks.org (GIT)
  - ▶ GPL-only

*Any problem with this?...*

- ▶ If it's free, you're the product
  - ▶ Google is watching your every move
  - ▶ Github?...
- ▶ It does not get much better even if you pay for it
- ▶ Cannot be used for critical data, even on Github's private repo offer
- ▶ Esp. not if there is your
  - ▶ internal sysadmin documentation
  - ▶ SCM infrastructure setup configurations
  - ▶ possibly containing IP addresses

*So what if you want to host a repository yourself?...*

## VCS on-premises

- ▶ makes sense for private repos
- ▶ easy rw/authenticated access (`ssh://`)
- ▶ but you get only the repo,
  - ▶ no UI
  - ▶ no bug-tracking
  - ▶ no nothing!



# VCS self-hosting suites

## CVS

- ▶ ViewVC (CVS, SVN)
- ▶ CVSWeb
- ▶ OpenGrok (CVS, SVN, Mercurial)

## GIT

- ▶ Gitlab – on-premises
- ▶ Gogs / **Gitea**
- ▶ GitWeb
- ▶ Cgit
- ▶ Gerrit (GIT) – heavy client

## Hybrid

- ▶ Redmine (CVS, SVN, Mercurial, GIT, Bazaar)
- ▶ Trac (SVN, GIT)



Search

SIMtester

[Overview](#)
[Activity](#)
[Wiki](#)
[Repository](#)

simtester @ master



Statistics

Branch:

master

Revision:

Name	Size	Revision	Age	Author	Comment
SIMLibrary		cf0a4d3b	4 months	Luca	Added source for v1.9
SIMTester		cf0a4d3b	4 months	Luca	Added source for v1.9
binaries		a62d6fa6	4 months	Luca	Added binaries for v1.9
.gitignore	51 Bytes	f6e9d9b4	about 6 years	Lukas Kuzmiak	Public release of: - SIMTester v1.4.5, 2013-12...

### Latest revisions

#		Date	Author	Comment
cf0a4d3b		09/27/2019 07:36 PM	<a href="#">Luca</a>	Added source for v1.9
a62d6fa6		09/27/2019 07:31 PM	<a href="#">Luca</a>	Added binaries for v1.9
6f3ceef		02/04/2016 11:22 AM	<a href="#">Philipp</a>	SIMTester v1.8.1, 2016-02-04 released
ea2e3692		01/04/2014 05:24 PM	<a href="#">Lukas Kuzmiak</a>	SIMTester v1.5, 2014-01-03 released - gsmmap.org upload functionality - AppDeSelect is not retried if it fails so the scan can continue (reported by Ondrej Mikle) - TARs to be fuzzed are now properly displayed (minor)
0e5523ea		12/25/2013 02:57 PM	<a href="#">Lukas Kuzmiak</a>	version 1.4.6 released: - transmitOnDefaultChannel() wrapper implemented, tries to reset the card after error during transmit() (on default channel)
f6e9d9b4		12/23/2013 07:32 PM	<a href="#">Lukas Kuzmiak</a>	Public release of: - SIMTester v1.4.5, 2013-12-21 - SIMLibrary v1.4.5, 2013-12-21
886e8416		12/23/2013 07:19 PM	<a href="#">Peter</a>	Init

[View differences](#)
[View all revisions](#) | [View revisions](#)

## Repository (Subversion)



## Browse

Name	Size	Revision	Date	Author	Comment
branches		1010	2007-12-18 19:12	jplang	Changes for 0.6.3 release.
plugins		838	2007-10-13 16:12	jplang	Simple CI plugin initial import.
tags		1012	2007-12-18 20:02	jplang	tagged version 0.6.3
trunk		1023	2007-12-22 13:47	jplang	Fixed 'export to' links positioning on wiki page.
app		1023	2007-12-22 13:47	jplang	Fixed 'export to' links positioning on wiki page.
config		1011	2007-12-18 19:50	jplang	Moved ProjectsController#list_documents and add...
environments		975	2007-12-10 18:58	jplang	Merged Rails 2.0 compatibility changes. Compati...
boot.rb	548 Bytes	772	2007-09-28 00:30	jplang	Native eol property set on config/*
database.yml.example	946 Bytes	772	2007-09-28 00:30	jplang	Native eol property set on config/*
environment.rb	3.3 KB	975	2007-12-10 18:58	jplang	Merged Rails 2.0 compatibility changes. Compati...
routes.rb	1.9 KB	1011	2007-12-18 19:50	jplang	Moved ProjectsController#list_documents and add...
settings.yml	2.5 KB	944	2007-12-02 14:52	jplang	Email notifications are now sent as Blind carbo...
db		994	2007-12-14 19:54	jplang	Search engine: issue custom fields can now be s...
doc		1004	2007-12-16 15:03	jplang	Changes for 0.6.2 release.
extra		1002	2007-12-15 13:23	nbc	bug when using apache authentication method
sample_plugin		774	2007-09-29 12:37	jplang	Redmine acts_as_* plugins moved to vendor/plugins.
svn		1002	2007-12-15 13:23	nbc	bug when using apache authentication method
Redmine.pm	5.2 KB	930	2007-11-24 17:55	jplang	Added user status criteria in Redmine.pm
create_views.sql	631 Bytes	396	2007-04-01 21:43	jplang	Initial commit for svn repository management an...
reposman.pl	4.1 KB	916	2007-11-18 19:51	nbc	* add Redmine.pm to authenticate with mod_perl ...
reposman.rb	6.3 KB	1002	2007-12-15 13:23	nbc	bug when using apache authentication method
svnservice.wrapper	932 Bytes	402	2007-04-02 21:01	jplang	added svn-executable property on perl scripts
files		67	2006-12-05 21:45	jplang	trunk moved from /trunk/redmine to /trunk
lang		1003	2007-12-16 14:33	jplang	Updated Japanese translation (Satoru Kurashiki).
lib		1021	2007-12-20 20:44	jplang	Trunk version changed to 0.6.devel

Redmine/SVN

## Revision 1022:

View differences

inline ▾

## trunk/app/views/layouts/base.rhtml (revision 1022)

@1022 @1021	
	1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
1	<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2	<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
3	<head>
4	<title><%=h.html_title %></title>
73	<div id="ajax-indicator" style="display:none;"><span><%= l(:label_loading) %></span></div>
74	
75	<div id="footer">
	Powered by <%= link_to Redmine::Info.app_name, Redmine::Info.url %> <%= Redmine::VERSION %> ©copy 2006-2007
76	Jean-Philippe Lang
	Powered by <%= link_to Redmine::Info.app_name, Redmine::Info.url %> <%= Redmine::VERSION %> ©copy: 2006-2007
76	Jean-Philippe Lang
77	</div>
78	</body>

## trunk/app/views/layouts/\_project\_selector.rhtml (revision 1022)

@1022 @1021	
1	<%= user_projects_by_root = User.current.projects.find(:all, :include => :parent).group_by(&:root) %>
2	<select onchange="if (this.value != '') { window.location = this.value; }">
3	<option selected><%= l(:label_jump_to_a_project) %></option>
4	<option disabled>---</option>
3	<option selected="selected"><%= l(:label_jump_to_a_project) %></option>
4	<option disabled="disabled">---</option>
5	<%= user_projects_by_root.keys.sort.each do  root  %>

Redmine/GIT

# Software Configuration Management (SCM)

- ▶ Ansible, Puppet, Chef, Saltstack, ...
- ▶ includes Version Control Systems
- ▶ although it has a broader meaning
- ▶ as broad as Infrastructure as Code

*// Questions on hosting VCS UI?*

## Who likes Java?

==> almost nobody, but there are a few exceptions where Java r0cks

- ▶ SIMTester
- ▶ CAS for SSO
- ▶ Apps leveraging ElasticSearch
- ▶ Apps leveraging Hadoop

# Documentation Management

```
_____  
/      /,  
/      //  
/____//  
(____(/
```

Jussi Roine



# Normal Documents

- ▶ Old-school network shares
- ▶ LibreOffice supports edit locks?

BONUS QUESTION // check if libreoffice has the file-share locking notice as Word does

# Collaborative Documents

## WYSIWYG

- ▶ Google Docs
- ▶ NextCloud / Collabora
- ▶ Collabora Online Development Edition (CODE) Docker Image

## Markup

- ▶ v2.overleaf (LaTeX → PDF)
- ▶ hackmd.io (Markdown → HTML/PDF)

*Which one do you have to avoid, by law?...*

# FEDERAL LAW 152 on personal data

==> avoid Google Docs

*What is personal data?...*

==> any **combination** of

- ▶ name surname
- ▶ phone number
- ▶ email
- ▶ tax
- ▶ cards
- ▶ address
- ▶ birth date
- ▶ ...

Try `host` and `tracert` against Google's services...

==> NOT HOSTED IN RUSSIA (at least not the entry points)

Note that even if it were, a CDN can be misleading.

# Content Management System (CMS)

## ON-PREMISES

- ▶ Wordpress (possibly headless)
- ▶ Joomla
- ▶ Drupal
- ▶ ...

# Wiki is dead

- ▶ Crappy syntax
- ▶ Docs are unmaintained (but Wikipedia)
- ▶ Docs now often go within the code repository

# DokuWiki Syntax

**\*\*bold\*\***

*//italic//*

\_\_underline\_\_

' 'monospace' '

<sup>super</sup>

<sub>suber</sub>

<del>crossed</del>



# Markdown Syntax

# main title

## title2

*\_italic\_*

**\*\*bold\*\***

<TAB>code block

<4 spaces>code block

some text

some `inline code` and text

1. ordered item

1. ordered item

- unordered item

- unordered item

# Markdown to HTML

## Discount is the fastest

```
apt install discount
```

## inline conversion

```
markdown file.md > file.html
```

## full web page creation

```
mkd2html file.md
```

# Markdown to PDF

Markdown → LaTeX → PDF

```
pandoc file.md -o file.pdf
```

# Documentation Management System (DMS)

## IN-THE-CLOUD WYSIWYG

- ▶ Altasian Confluence

## ON-PREMISES Wiki syntax

- ▶ MediaWiki (PHP)
- ▶ DokuWiki (PHP), no database

## ON-PREMISES Markdown syntax

- ▶ Redmine (Wiki/Markdown plugin)
- ▶ GIT/Gollum
  - ▶ gets best edited locally
  - ▶ web interface is good only for reads
- ▶ **CODIMD / HackMD's engine**
  - ▶ nice web interface for collaborative interaction

# Learning Management System (LMS)

- ▶ aka Virtual Learning Environment (VLE)
- ▶ aka Learning content management system (LCMS)

## ON-PREMISES

- ▶ Moodle
- ▶ any other one?

\_\_\_\_\_  
/        /,  
/        //  
/\_\_\_\_\_//  
(\_\_\_\_\_(/

Jussi Roine

*// Questions on doc mgmt?*

# Bug-Tracking & Helpdesk Systems

## Proprietary

- ▶ SpiceWorks
- ▶ *any other in mind?*

## FOSS

- ▶ GLPI – inventory and fleet maintenance
- ▶ Redmine – project mgmt & bug tracking
- ▶ Bugzilla – used at Redhat & FreeBSD
- ▶ GNATS – used at NetBSD
- ▶ ORTS – the heavy germans
- ▶ Trac – SCM & project mgmt

## Corporate workstations...

Users may not have root/administrator access

- ▶ less damage when infected (userland only)
- ▶ your chosen daemons will run...



## Workstation incident monitoring

*assuming a dedicated host group*

- ▶ workstation status on the incident dashboard
  - ▶ apps installed?
  - ▶ system up-to-date? – *no need to update an inventory*
- ▶ eventually more in the status **without alerting**
  - ▶ VPN enabled?

LAB // monitor and PoC a fleet of workstations with appropriate scripts

## Workstation performance monitoring

- ▶ eventually grab performance metrics
- ▶ **but not every 30 seconds**
- ▶ (& caching-capable)

LAB // which are the caching-capable agents?

## Rescuing workstations

*Assuming internal network or vpn*

*You got a ticket from a user asking for help on his desktop env*

*How to go and reach his desktop?...*

## Remote desktop protocols

- ▶ RDP (new session)
- ▶ VNC & alike (possibly same user session)
- ▶ SSH with X11 forwarding (apps only)

even easier when no graphical interface is required

- ▶ SSH (can do a lot including `nmcli...`)

→ simply deploy ops' pubkey(s) on all workstations (and bind sshd to internal IPs)

# Remote desktop products

## Windows

- ▶ (RDP server)
- ▶ ultravnc
- ▶ realvnc

## GNU/Linux

- ▶ x11vnc vs. tigervnc vs. tightvnc
- ▶ NoMachine
- ▶ X2GO
- ▶ XRDP (RDP on MacOS and Linux)
- ▶ SSH with X11 forwarding (screen not shared with the user)

### *clients-only*

- ▶ rdesktop – RDP
- ▶ Remmina – RDP
- ▶ ultravnc – VNC

# SSH with X11 forwarding

## On the server-side

```
apt install xauth x11-apps  
# xvfb
```

Note XVFB prevents from installing a full-blown X server

## On the client-side

*windows + xming or mobaxterm*

PuTTY + enable X11 forwarding

*gnu/linux*

```
ssh -X / -Y ...
```

**check when ready**

```
xclock
```

```
xeyes
```

*What to do when the user is behind a foreign NAT?...*



==> two solutions

- ▶ intermediate server
- ▶ reverse tunnel

## With an intermediate server

### GUI

- ▶ (TeamViewer)
- ▶ (AnyDesk)
- ▶ NoMachine? // LAB as intermediate?
- ▶ X2GO? // LAB as intermediate?

### CLI

- ▶ `tmate`

Note: similar to SSH + `screen -x`

## With an reverse tunnel

CLI or GUI, whatever you need

- ▶ setup a script on user's machine beforehand (hint: `ssh -R`)
- ▶ which calls a dedicated shell server of yours to create a reverse tunnel
- ▶ enable that script at init time

You can then connect to the appropriate port on your reverse-bounce server.

*// Questions on rescuing workstations?*