

FreieSoftwareOG –

GIT -

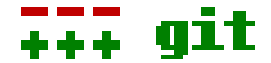
open source, fast,
distributed version control system

Vorstellung



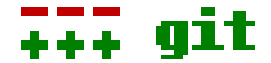
- Wer ist wer ?
- Erfahrungen
 - CVS ?
 - Subversion ?
 - andere DVCS ?
 - andere closed-source VCS ?

Agenda



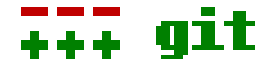
- Herkunft von GIT
- neue Paradigmen
- Wie läuft es ?
- Fazit
- Perspektive
- Referenzen

Herkunft von GIT



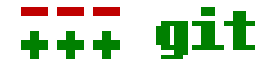
- Dinosaurier : RCVS, CVS, SVN
- Linux auf Bitkeeper [2002 - 2005]
- Linus ärgert sich und ... agiert
Kernentwicklung in 2 Wochen
- Vorreiter in der DVCS Welt

innovative Paradigmen

The Git logo, consisting of three red dashes above three green plus signs, followed by the word "git" in a green, monospace font.

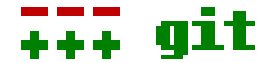
- WTF is DVCS ?
 - distributed ? ohne zentrale Referenz ?
 - off-line
 - client == server
- Branches sind die Norm
- Ordnerstruktur auch Versioniert
- Fortschritte in merging
 - Rahmen für die Lösung von Konflikten
 - Vorbereitung von Patches

neue Paradigmen



- Kompakte Speicherung
durch Hashing (SHA-1)
- gutes Handling von Metadaten
rebase, Bearbeitung der Historie...
- offenes Versionierungsmodell
- **Geschwindigkeit !**
ermöglicht viele kleine commits

Wie läuft es ?



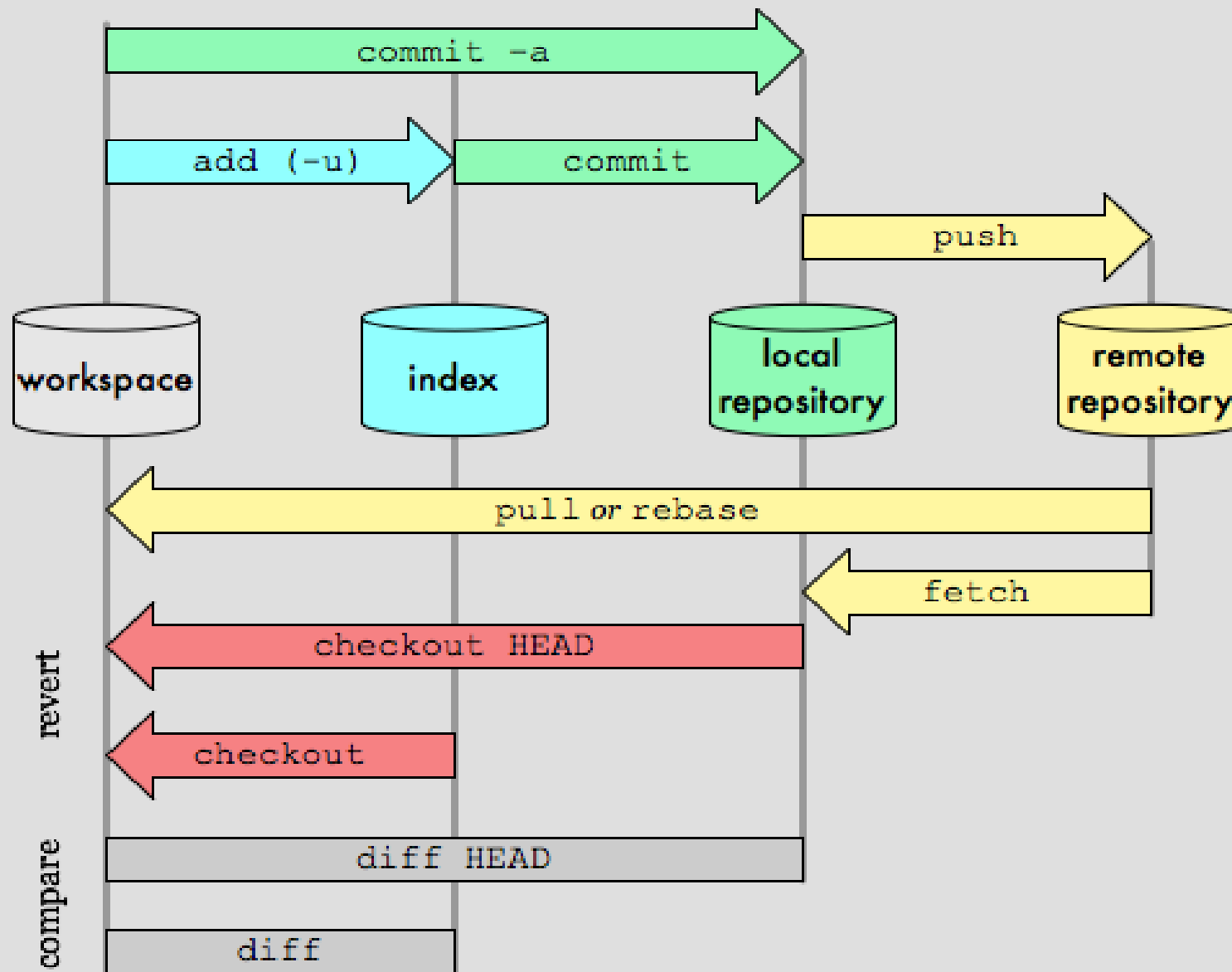
- laufender Ordner = aktuelle Version
`edit`
- `.git` Ordner = Änderungsdatenbank
`git commit`
- zentrales Repository
= Basis zum Austausch
`git clone`

Wie läuft es ?

+++ git

Git Data Transport Commands

<http://osteele.com>



Wie läuft es ?



Git commands (137)

add	fast-export	merge-one-file	revert
am	fast-import	merge-resolve	rm
annotate	fetch	merge-subtree	send-email
apply	fetch-pack	merge-tree	send-pack
archimport	filter-branch	mergetool	sh-setup
archive	fmt-merge-msg	mktag	shell
bisect	for-each-ref	mktree	shortlog
blame	format-patch	mv	show
branch	fsck	name-rev	show-branch
bundle	fsck-objects	pack-objects	show-index
cat-file	gc	pack-redundant	show-ref
check-attr	get-tar-commit-id	pack-refs	stash
check-ref-format	grep	parse-remote	status
checkout	gui	patch-id	strip-space
checkout-index	hash-object	peek-remote	submodule
cherry	http-fetch	prune	svn
cherry-pick	http-push	prune-packed	symbolic-ref
citool	imap-send	pull	tag
clean	index-pack	push	tar-tree
clone	init	quiltimport	unpack-file
commit	init-db	read-tree	unpack-objects
commit-tree	instaweb	rebase	update-index
config	log	receive-pack	update-ref
count-objects	lost-found	reflog	update-server-info
cvsexportcommit	ls-files	relink	upload-archive
cvsimport	ls-remote	remote	upload-pack
cvsserver	ls-tree	repack	var
daemon	mailinfo	repo-config	verify-pack
describe	mailsplit	request-pull	verify-tag
diff	merge	rerere	what-changed
diff-files	merge-base	reset	write-tree
diff-index	merge-file	rev-list	
diff-tree	merge-index	rev-parse	gitk

Wie läuft es ?

+++ git

Some GUI tools...

add

fetch

rm

mergetool

shell

branch

mv

show

gc

stash

checkout

grep
gui

status

citool

pull
push

tag

clone
commit

init

rebase

config

log

remote

diff

merge

reset

gitk

Wie läuft es ?

+++ git

Every day use...

```
add
branch
checkout
clone
commit
config
diff

fetch
gc
grep
init
log
merge

mv
pull
push
rebase
remote
reset

rm
show
stash
status
tag
```

Wie läuft es ?



Occasional use...

```
add
am
annotate      fetch
apply

archive
bisect
blame
branch
bundle

checkout      grep
              gui

cherry
cherry-pick
citool
clean
clone
commit

config        instaweb
              log

describe
diff          merge

revert
rm
send-email

mergetool

mv

shell
shortlog
show
show-branch

stash
status

submodule
svn

pull
push
quiltimport

rebase

reflog

remote

reset

what changed

gitk
```

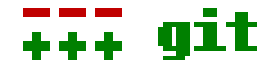
Wie läuft es ?



And the plumbing...

```
add
am
annotate
apply
archimport
archive
bisect
blame
branch
bundle
cat-file
check-attr
check-ref-format
checkout
checkout-index
cherry
cherry-pick
citool
clean
clone
commit
commit-tree
config
count-objects
cvsexportcommit
cvsimport
cvsserver
daemon
describe
diff
diff-files
diff-index
diff-tree
fast-export
fast-import
fetch
fetch-pack
filter-branch
fmt-merge-msg
for-each-ref
format-patch
fsck
fsck-objects
gc
get-tar-commit-id
grep
gui
hash-object
http-fetch
http-push
imap-send
index-pack
init
init-db
instaweb
log
lost-found
ls-files
ls-remote
ls-tree
mailinfo
mailsplit
merge
merge-base
merge-file
merge-index
merge-one-file
merge-resolve
merge-subtree
merge-tree
mergetool
mktag
mktree
mv
name-rev
pack-objects
pack-redundant
pack-refs
parse-remote
patch-id
peek-remote
prune
prune-packed
pull
push
quiltimport
read-tree
rebase
receive-pack
reflog
relink
remote
repack
repo-config
request-pull
rerere
reset
rev-list
rev-parse
revert
rm
send-email
send-pack
sh-setup
shell
shortlog
show
show-branch
show-index
show-ref
stash
status
strip-space
submodule
svn
symbolic-ref
tag
tar-tree
unpack-file
unpack-objects
update-index
update-ref
update-server-info
upload-archive
upload-pack
var
verify-pack
verify-tag
what-changed
write-tree
gitk
```

Wie läuft es ?

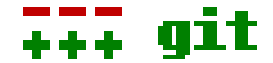


Demo !

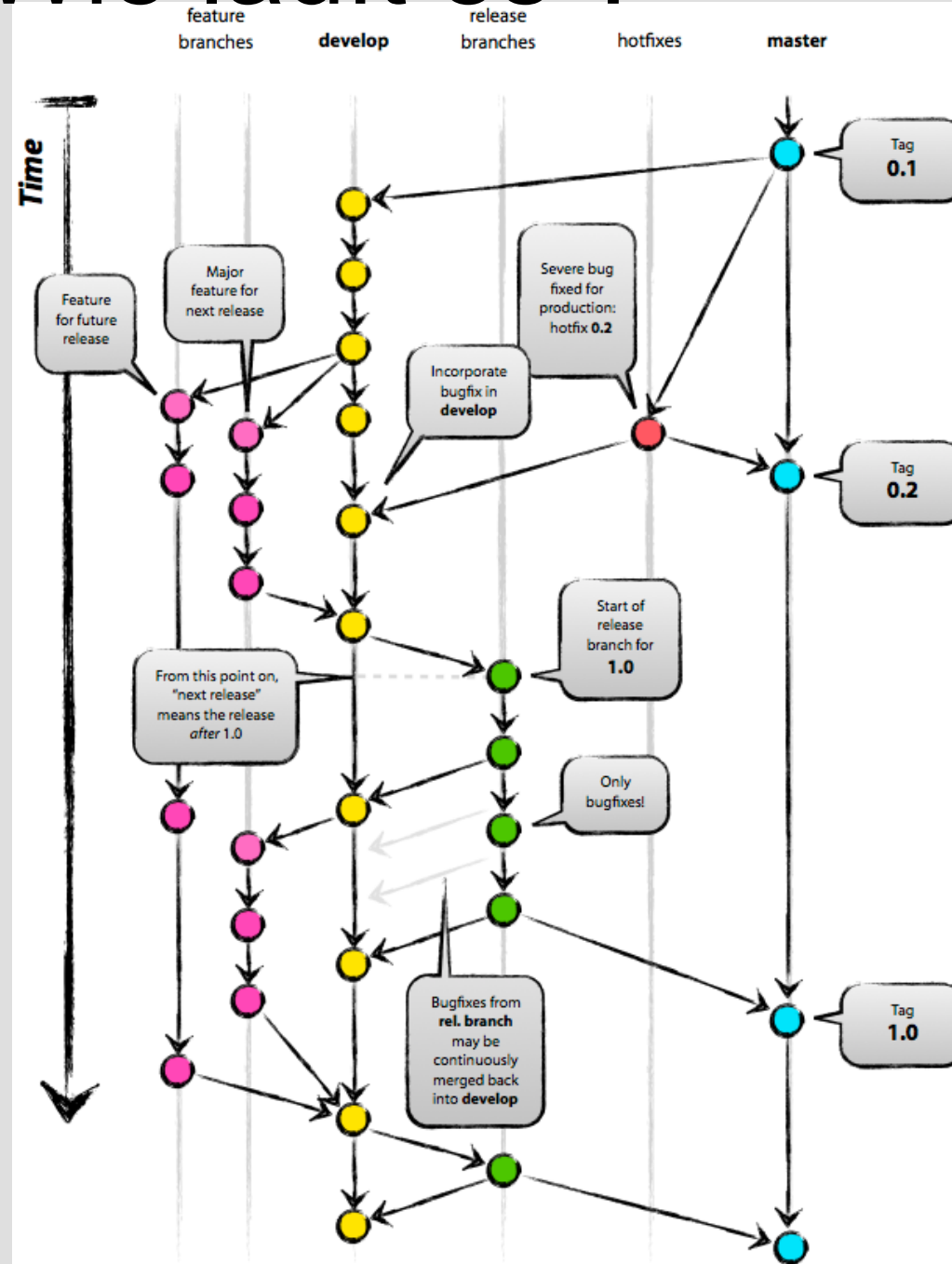
```
manou@snb1: ~/DATA/GIT-TALK/GIT-DEMO
Datei Bearbeiten Ansicht Terminal Hilfe
aaa.txt: locally modified
manou@snb1:~/DATA/GIT-TALK/GIT-DEMO$ git diff
diff --git a/aaa.txt b/aaa.txt
index 3cb4eab..b84b9aa 100644
--- a/aaa.txt
+++ b/aaa.txt
@@ -1,2 +1,3 @@
 This is some text I tried to type quickly.
+This is another line

manou@snb1:~/DATA/GIT-TALK/GIT-DEMO$ git commit -m "2. commit"
# On branch master
# Changed but not updated:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   aaa.txt
#
no changes added to commit (use "git add" and/or "git commit -a")
manou@snb1:~/DATA/GIT-TALK/GIT-DEMO$ git add .
```

Wie läuft es ?



- Arbeitsmodell

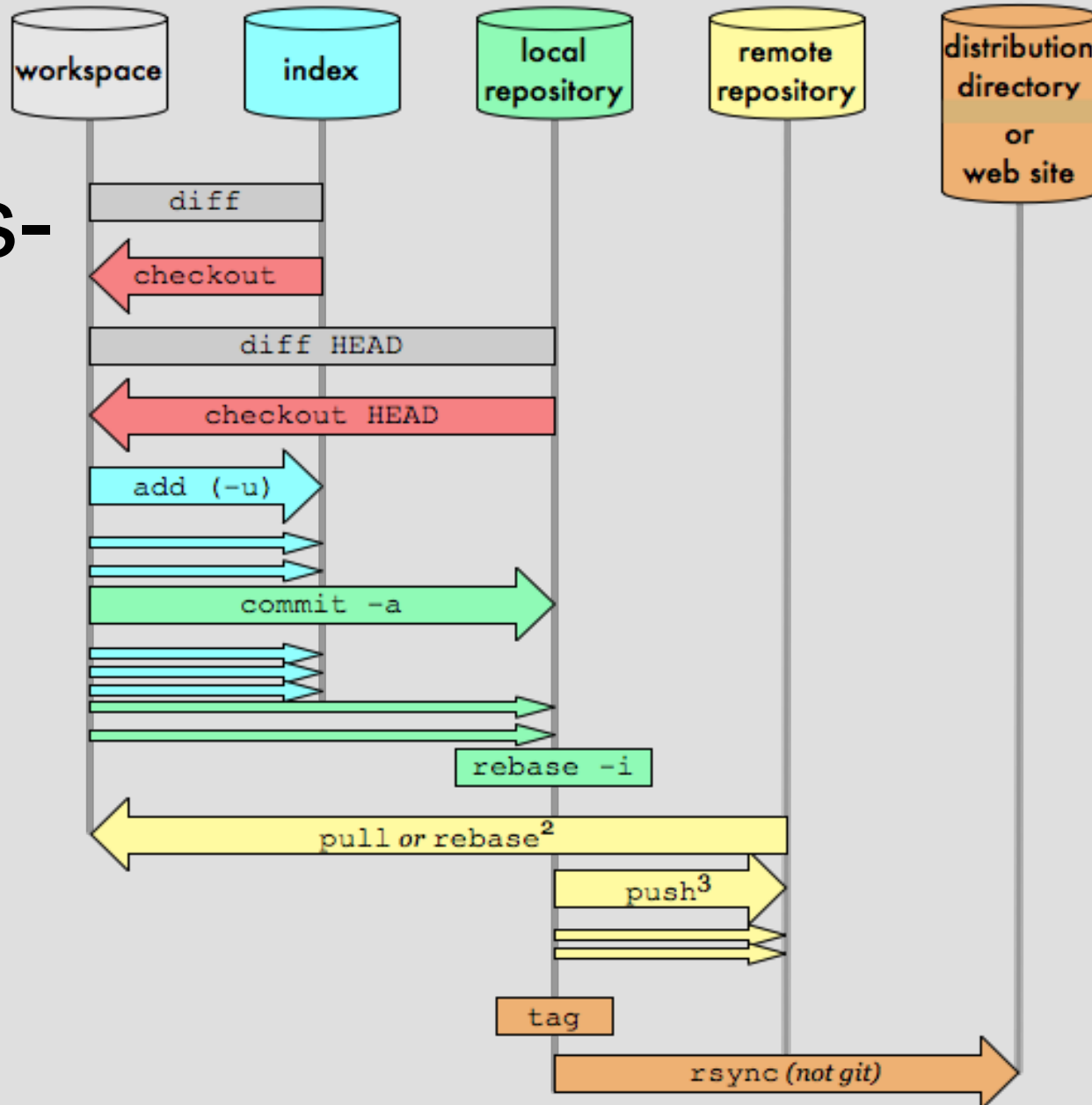


Wie läuft es ?

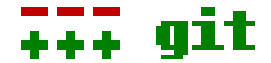
+++ git

A¹ Git Workflow
<http://osteele.com>

- Arbeitsmodell



Fazit



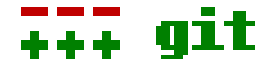
- modernes Ecosystem
 - GITHub, ...
- Vorteile
 - viele Anwender
 - unter den Free Software Projekten
 - integriert in standard IDEs (Eclipse, ...)
 - Offenheit vom Branching-modell
- Nachteile
 - steile Lernkurve
 - Offenheit vom Branching-modell

Perspektive



- GIT ist reif
- modernes Ecosystem
GITHUB, ...
- DVCS Welt ist in schneller Bewegung

Referenzen



- Linus bei Google Tech Talks (video)
- www.git.org
- Eric Sink
- excess.org/article/2008/07/ogre-git-tutorial/
- Philip Hofstetter (videos)
- a successful git branching model
- GIT: branching and Fast-Forward Merging
- Oliver Steele
- uvm...

Danke !



Fragen ?

Weitere Informationen bekommen Sie hier:

<http://www.FreieSoftwareOG.org>

und

Kontakt@FreieSoftwareOG.org

oder kommen Sie doch einfach zu unserem
regelmäßigen Treffen,

jeden 1. Mittwoch im Monat ab 20:00 Uhr.

(Treffpunkt laut Webseite)