

Dr. R. Guderlei | eXXcellent solutions GmbH

Versionsverwaltung - Zentral oder verteilt?

Grundsätzliches

- Kein Git/Mercurial/Bazaar/whatever-Manual-Vortrag
- Kein „X ist besser als Y“-Vortrag
- Fragen sind immer willkommen!

Warum die Frage?

- Latente Unzufriedenheit mit SVN
- DVCS sind „cool“
- Rasante Entwicklung bei DVCS
in den letzten 2 Jahren

Agenda

- Was ist verteiltes Versionsmanagement?
- Workflows im Team
- Arbeiten mit DVCS:
 Branching/Tagging/Merging
- Tools
- Umstieg auf ein verteiltes
 Versionsmanagement

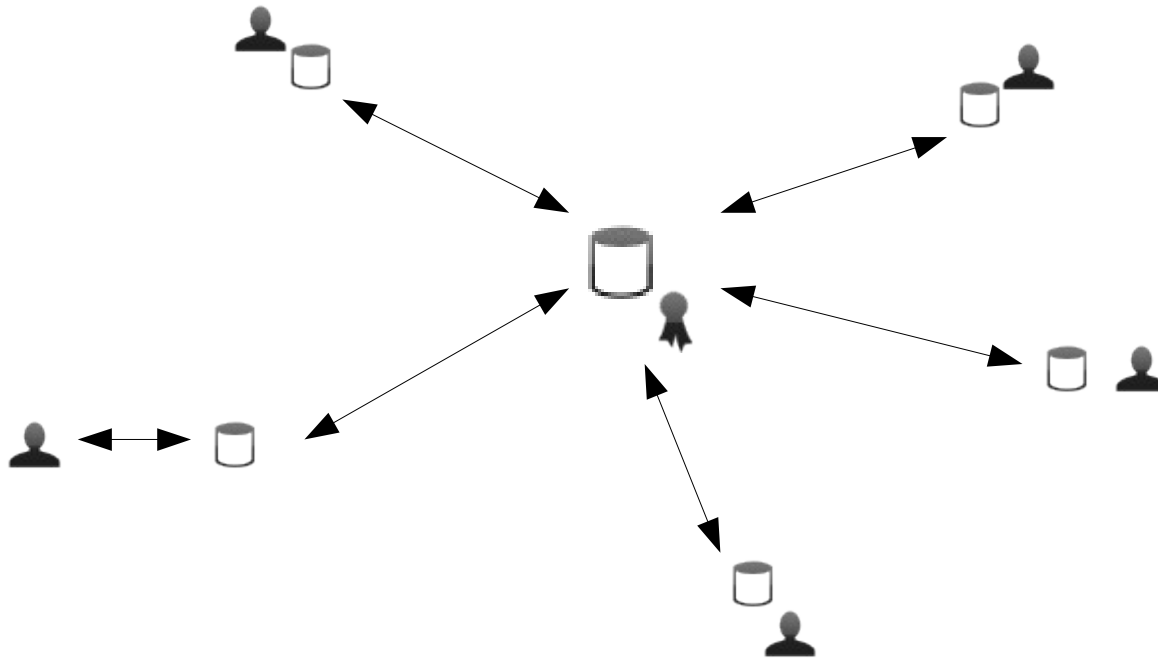
Verteiltes Versionsmanagement

- Peer-to-peer-Architektur, (kein client-server)
- Kein zentrales Repository
- Jeder Entwickler besitzt ein vollständiges lokales Repository
 - offline arbeiten ist möglich
 - hohe Geschwindigkeit vieler Operationen, da kein Netzzugriff notwendig ist

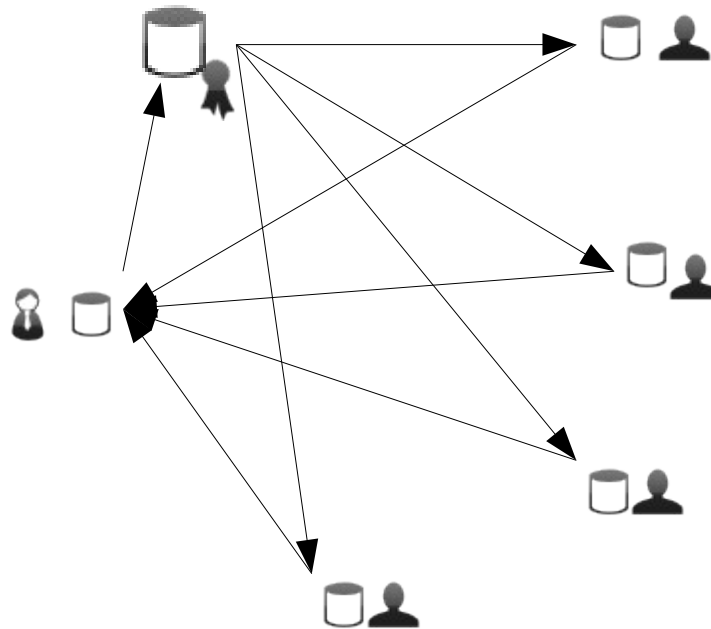
Die Kandidaten



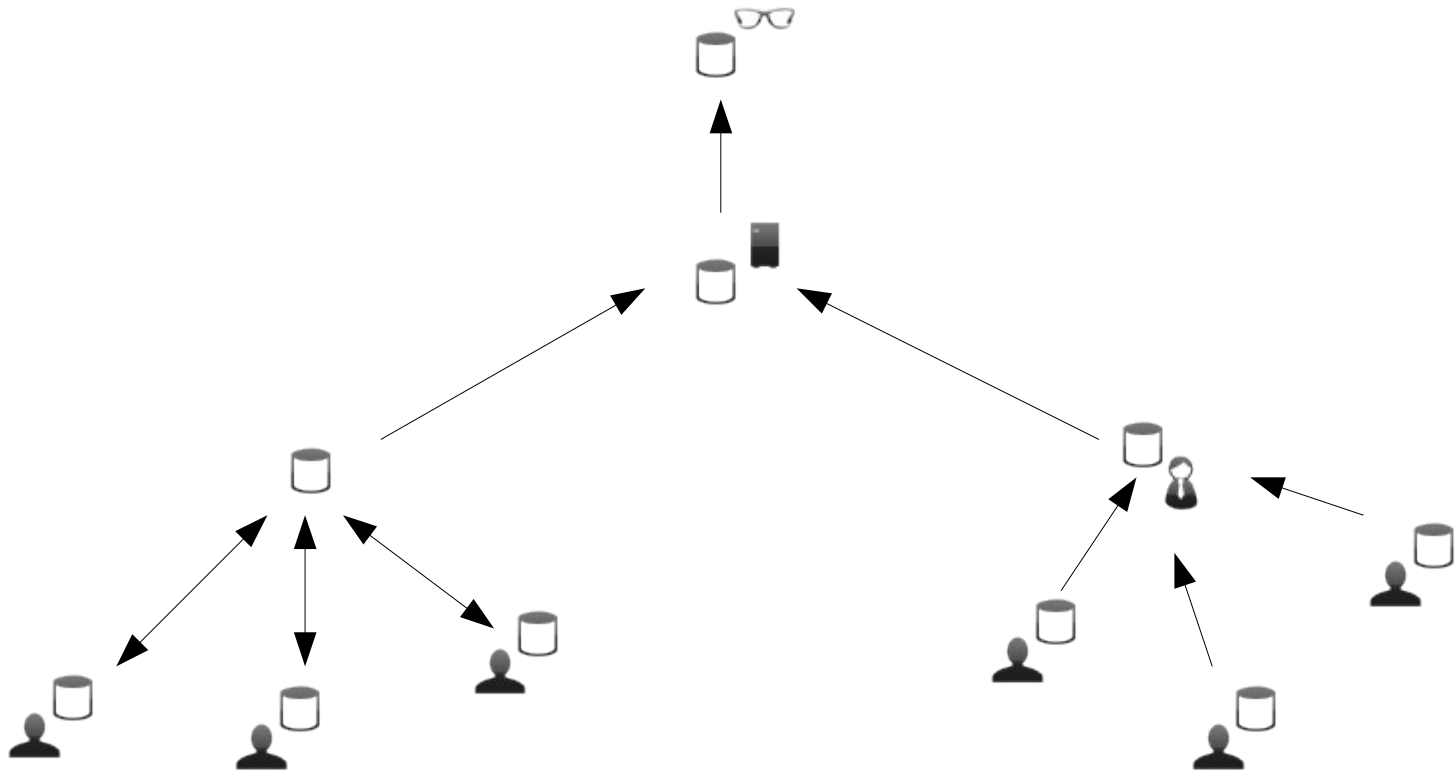
Workflow - „zentrales“ Repo



Workflow - blessed Repo



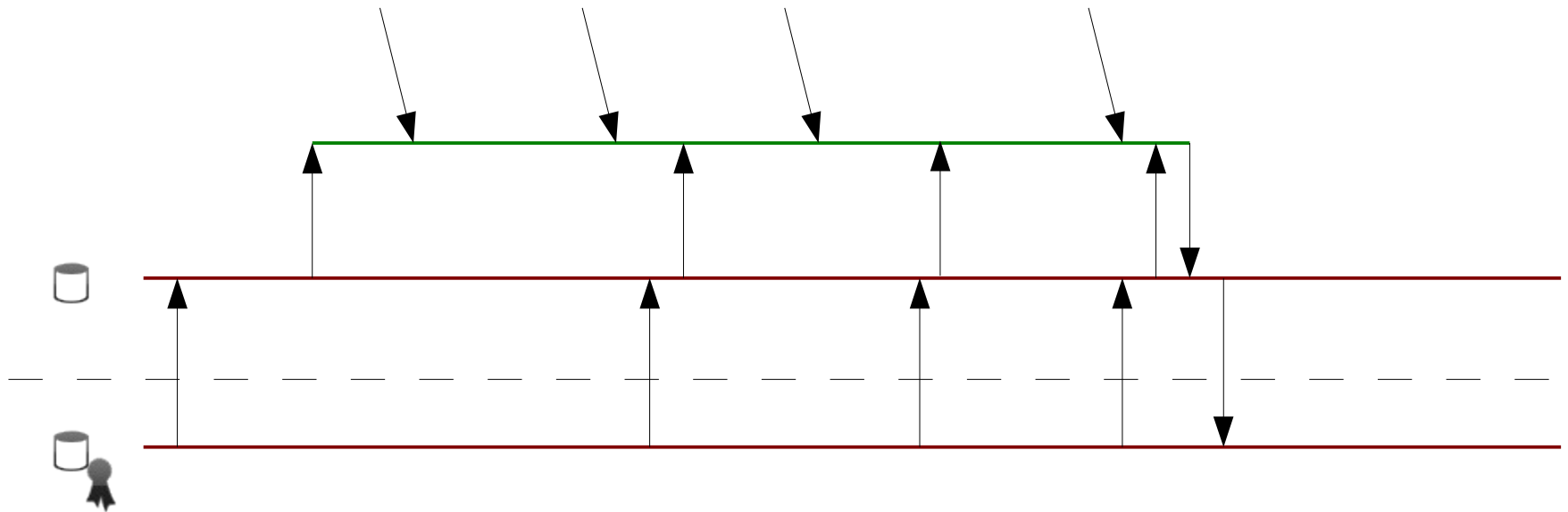
Workflow - individuelle Struktur



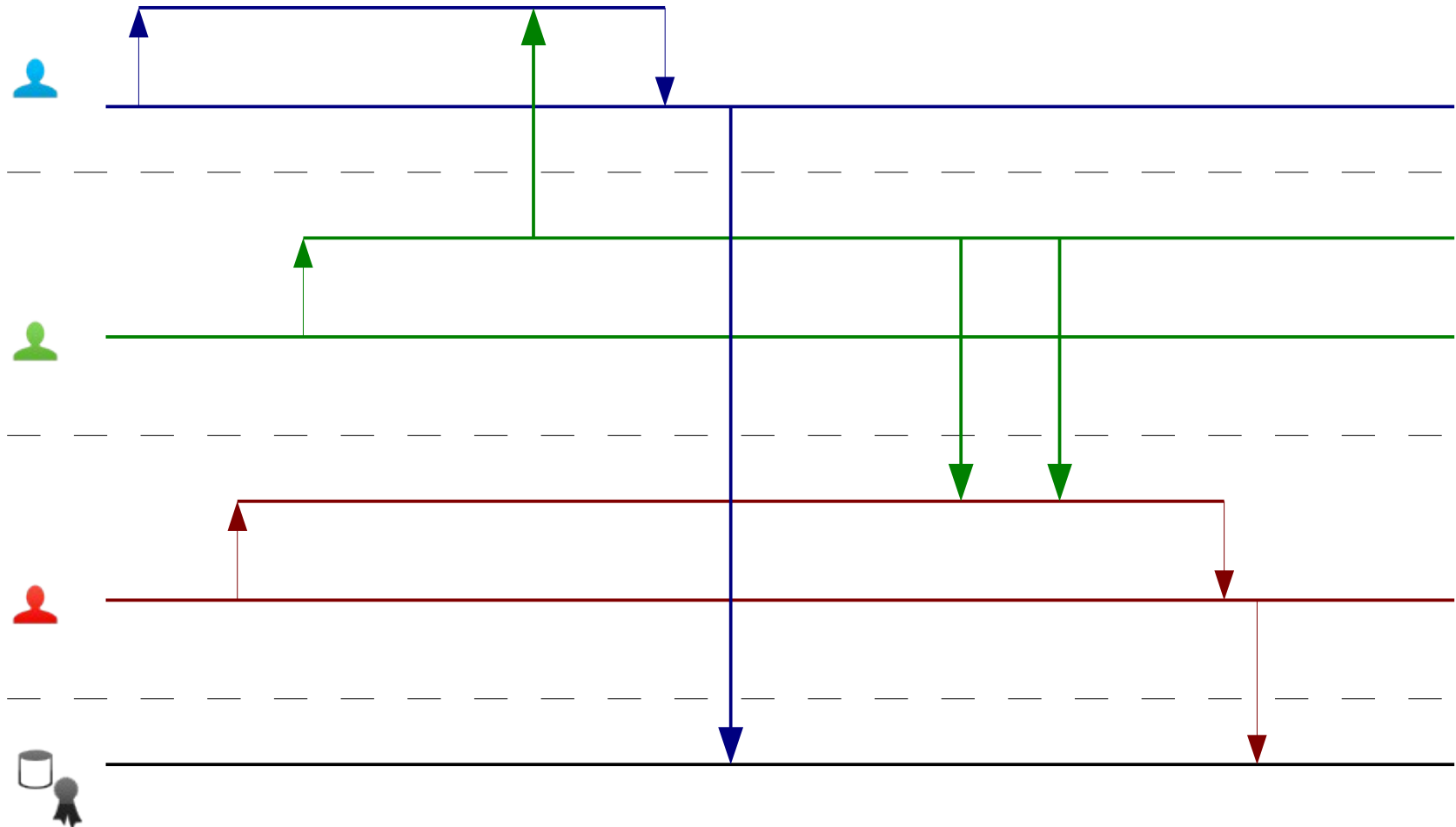
Arbeiten - Generelles

- Commit little, commit often
- Integrate often
 - continuous integration
 - feature branches

Arbeiten - Feature Branches



Arbeiten - Promiscuous Integration

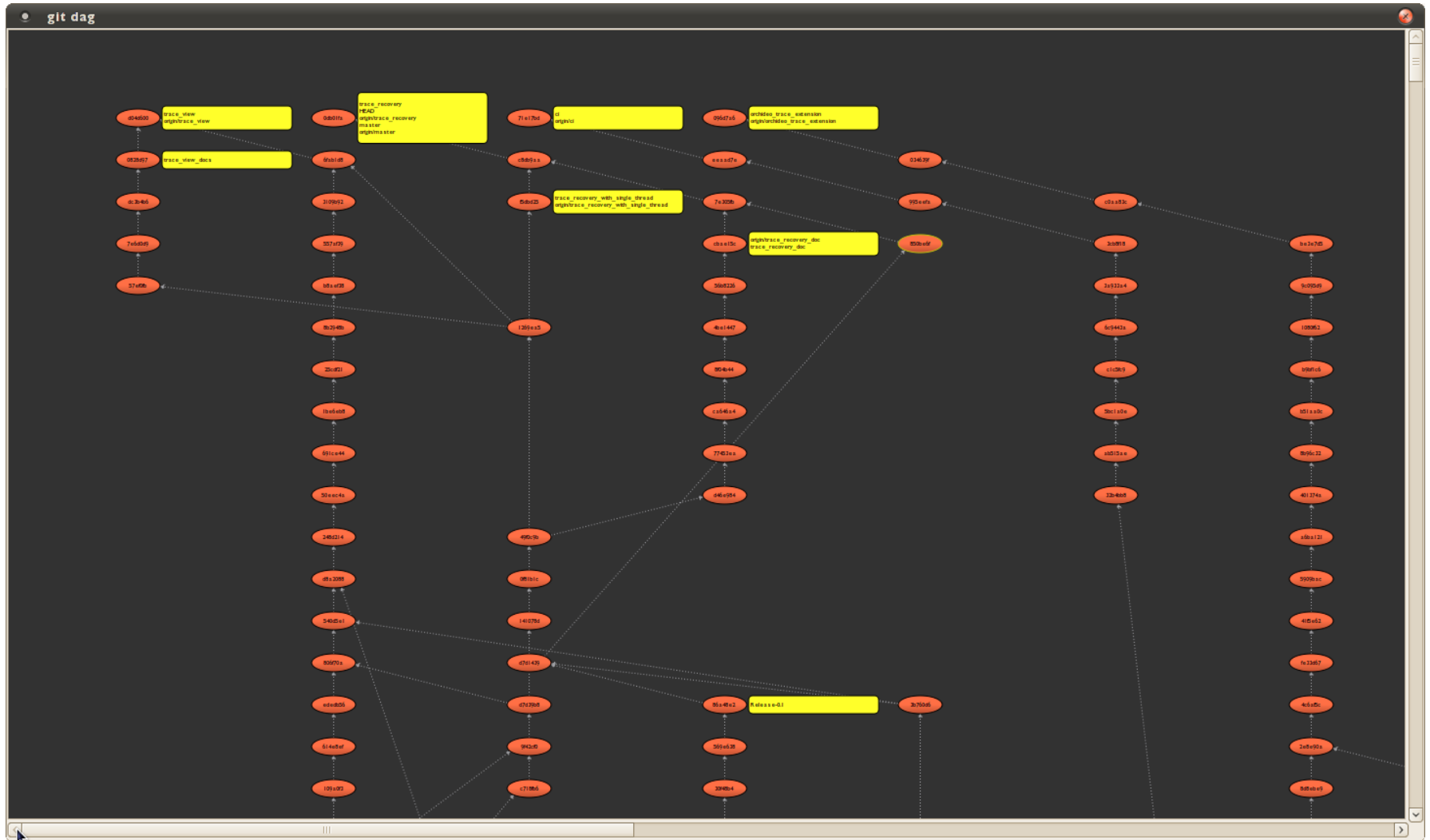


Arbeiten - Release Branches



Erweiterung: Ein Branch pro (Major-)Release

Never merge back!



Arbeiten - Tagging Guidelines

- DVCS verwenden hashes statt Versionsnummern und haben i.A. keine lineare Historie
→ tagging erleichtert die Orientierung
- Getagged werden sollten deshalb
 - Releases
 - Builds (im Repo des CI-Systems)
 - Abstimmungspunkte
 - Wichtige Aktionen: QA, Approvals, ...

Tools - ein Überblick

- Wichtigster Client: Command Line
- Standalone für Windows, Linux, OS X
- IDE-Integration (Eclipse, IntelliJ, Netbeans),
- Build-Tools (Maven, CI-Server)
- Integration in Windows-Explorer (TortoiseXyz)

Tools - Add-ons

- Reviews
 - Git: Gerrit
 - Mercurial: ReviewBoard, FileReview
- Online-Hosting
 - Github
 - Gitorious
 - bitbucket

Umstieg - DVCS als Client

- Alle DVCS-Clients bieten Spezial-Clients für SVN (git-svn, hgsubversion, bzs-svn)
- Vorsicht bei:
 - svn:external
 - Partial checkouts
 - Merges zwischen öffentlichen Branches, da evtl. die lineare Historie zerstört wird

Umstieg - Repository-Migration

- Im-/Export-Funktionalität für
 - Subversion
 - CVS
 - Viele andere
- <http://mercurial.selenic.com/wiki/RepositoryConversion>
- https://git.wiki.kernel.org/index.php/Interfaces,_frontends,_and_tools#Interaction_with_other_Revision_Control_Systems

Zentral oder Verteilt?

- It depends ...
 - Spielt die verteilte Natur ein Rolle?
 - Wird die Flexibilität benötigt?
 - Wie groß ist die Akzeptanz für ein neues Tool?

Und, welches DVCS jetzt?

- Git
 - Schnell
 - Local branching
 - Mercurial
 - schnell
 - Google verwendet es
 - Bazaar
 - Rename tracking
- selber ausprobieren

Fazit

- Branching/Merging besser als bei zentralen VCS
- Stabile Branches
- Weicher Übergang zu DVCS möglich
- Höhere Komplexität vs. Mehr Flexibilität
- Konkrete Wahl des DVCS ist Frage des persönlichen Geschmacks

**Vielen Dank für Ihre
Aufmerksamkeit!**

Fragen??