

ClearCase, the journey to Git

Migrating your skills and VOBs to Git

Luca Milanesio

Luca@GerritForge.com

June 2013

About the Speakers



Luca Milanesio

Luca Milanesio is Director and cofounder of GerritForge LLP, the leader Git and Gerrit competence center for the Enterprise and key technology partner of CollabNet Inc. Luca contributed to the Gerrit community and allowed the introduction of Enterprise code-review workflow in large Enterprises worldwide. Thanks to GerritForge LLP and CollabNet Inc. TeamForge is now the most advanced ALM platform with integrated Git support and Code-Review for the Enterprise.



Fredrik Luthander

Fredrik Luthander is an independent Git and Software Configuration Management consultant at Ericsson across four large R&D sites in Europe, Asia and North America. Fredrik is one of the historical Gerrit contributors and has more than 10 years' experience in ClearCase Multisite Application Management and Training and, before Ericsson, has administered very large installations at Siemens Medical and SONY Mobile and driven the company migration to Git.

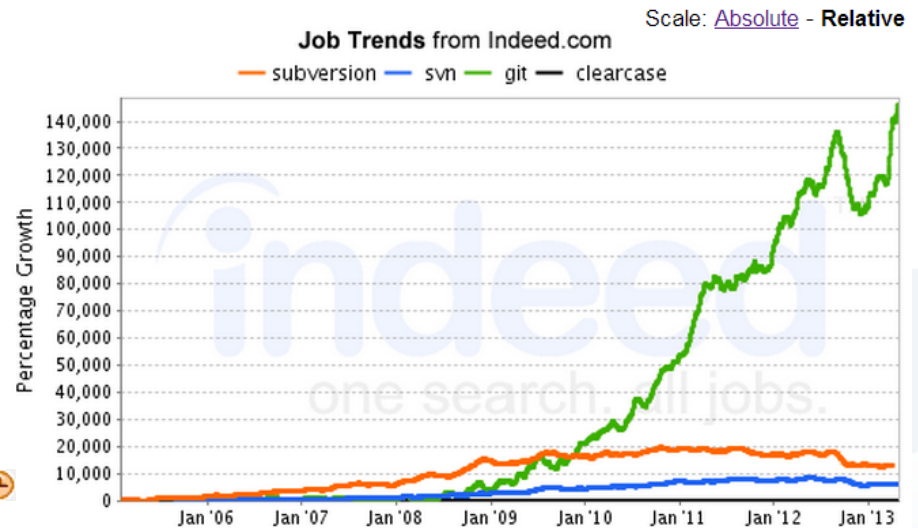
About CollabNet

- Industry leader in Enterprise Cloud Development (ECD)
- Agile ALM, DevOps, SCM, Git & SVN – Product & Services
- Download TeamForge for FREE @ www.collab.net/git

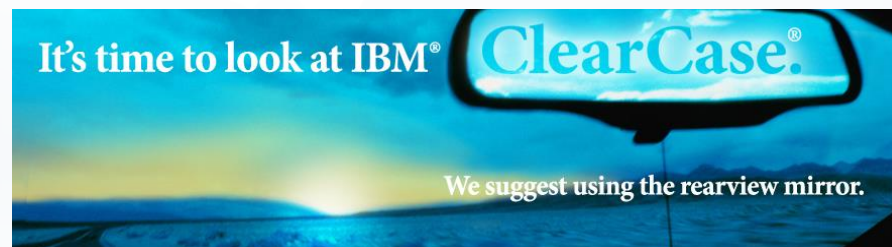
Agenda

- Bird's-eye view
 - Changes and revisions
- Version control lifecycle
 - Branches strategy and workarea
- Migration strategy and problems
 - Security
 - Replication

Enterprise Git - 100% Annual Growth



Source: Indeed.com



Learn more: www.collab.net/pullahead

Bird's-eye view

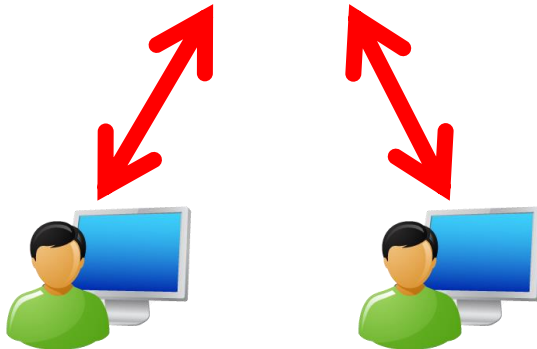
ClearCase and Git concepts and differences

ClearCase (CC) vs. Git – at a glance

<i>Aspect</i>	ClearCase	Git
Repository model	Client-server	Distributed
Revision IDs	Branch + number	Global alphanumeric ID
Scope of Change	File	Directory tree snapshot
Concurrency model	Merge	Merge
Storage Method	Deltas	Full content
Client	CLI, Eclipse, CC Client	CLI, Eclipse, GitEye
Server	UNIX, Windows <i>legacy systems</i>	UNIX, Windows, OS X, Cloud-based (CloudForge)
License	Proprietary	GPL

Centralized vs. Distributed Version Control

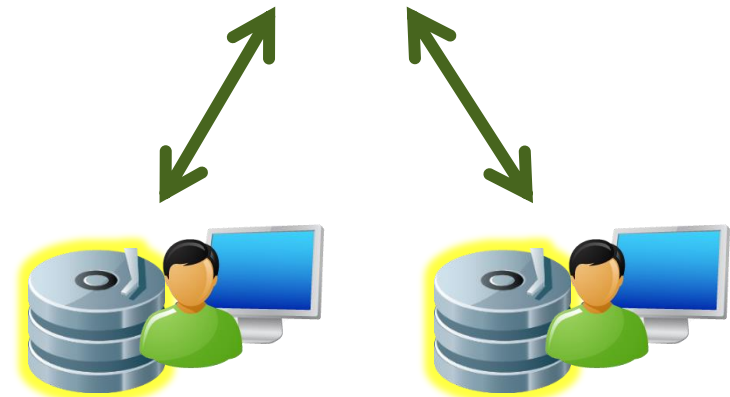
Centralized



Server = Single Point of Failure

High availability / bandwidth needed

Distributed

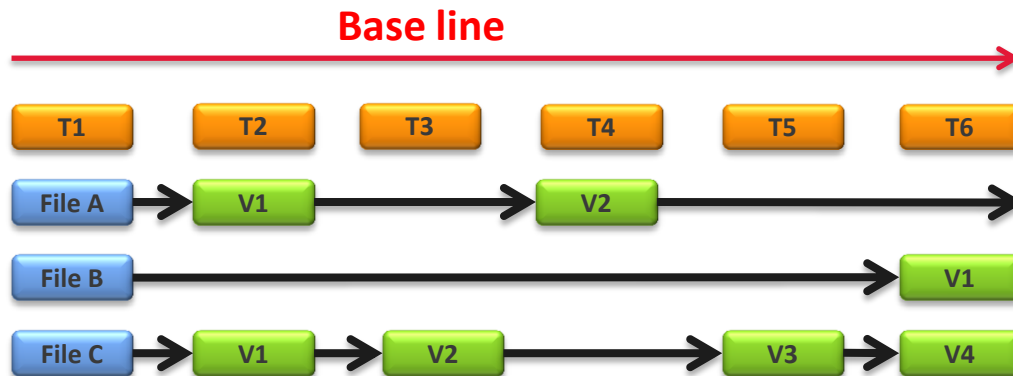


Peer-to-Peer availability

High performance with low-bandwidth

CC file deltas vs. Git snapshots

- CC = individual files version deltas



Legend

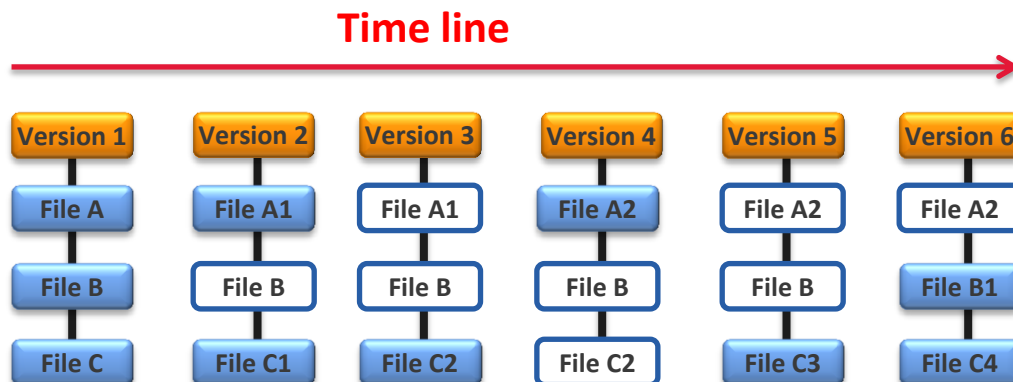
Time / Version
snapshot

File Delta

File content

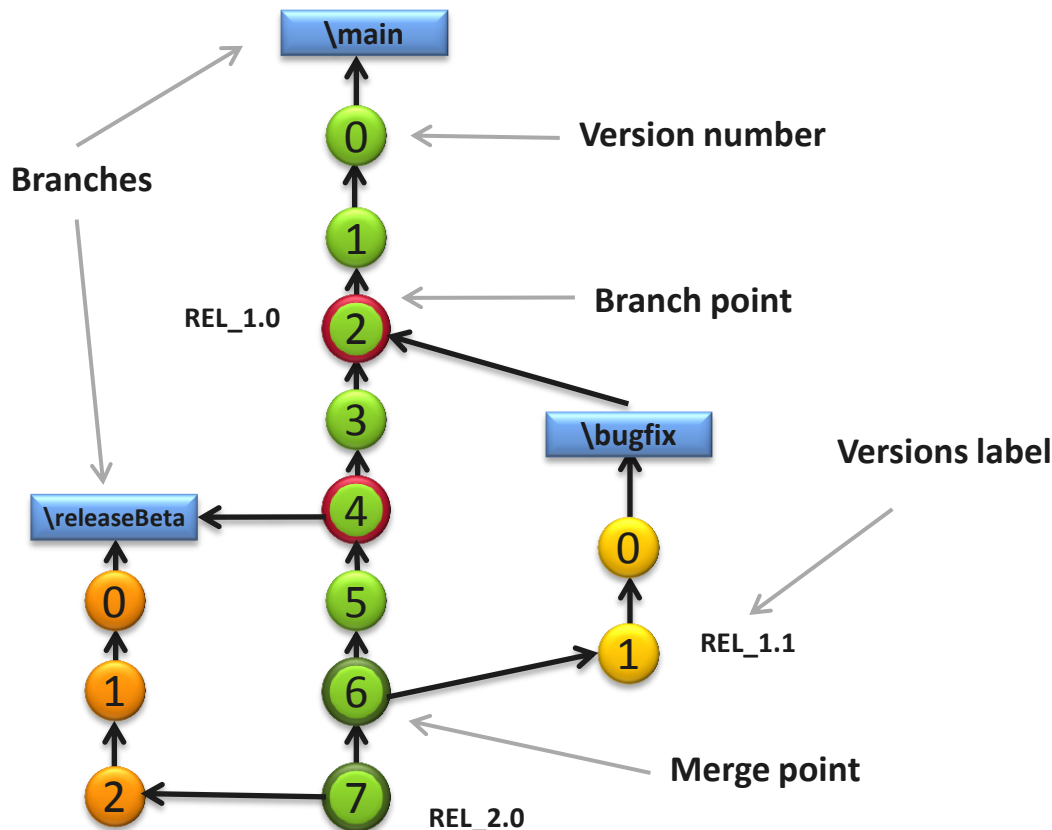
File link

- Git = full content snapshot



CC: linear version history

- Every file has an incremental version history
 - One version has one predecessor
 - Branch and merge points between baselines



X **Each file** has a **different history**

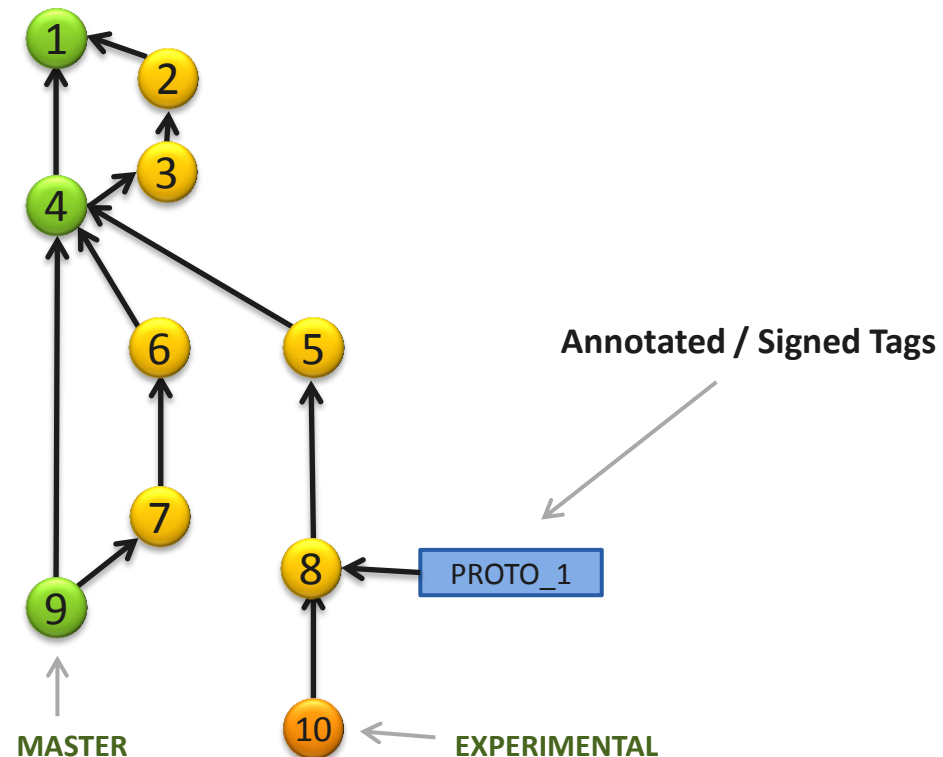
X **Lots of tags** needed for builds demarcation

X **Difficult to reconstruct the full graph of changes**

Git: graph of commits

- ✓ **Global view** of the repository
- ✓ **Commit ID** identify a full consistent snapshot
- ✗ **Commits** may belong to more than one branch
- ✗ **Branches** complexity may increase

- Git a generic directed acyclic graph of commits
 - Commits branches are paths in the graph
 - Tags are first class objects for actual releases



POLL: What is your organizations' Git experience ?

- None - New to Git
- Familiar with Git, but not using
- Started ramping up on Git skills
- Git used actively, alongside other tools
- Git is our leading SCM tool



Version Control lifecycle

ClearCase versus Git lifecycle

CC Join UCM project vs. Git clone

- CC: to join a UCM project

1. Select the VOB (versioned object base) + join the project
2. Create a local UCM (Unified Change Management)
3. Create a local UCM integration view to provide changes.

- GIT: different ways to start a project

- Clone the entire repository
- Create a local repository
- Connect to a remote repository

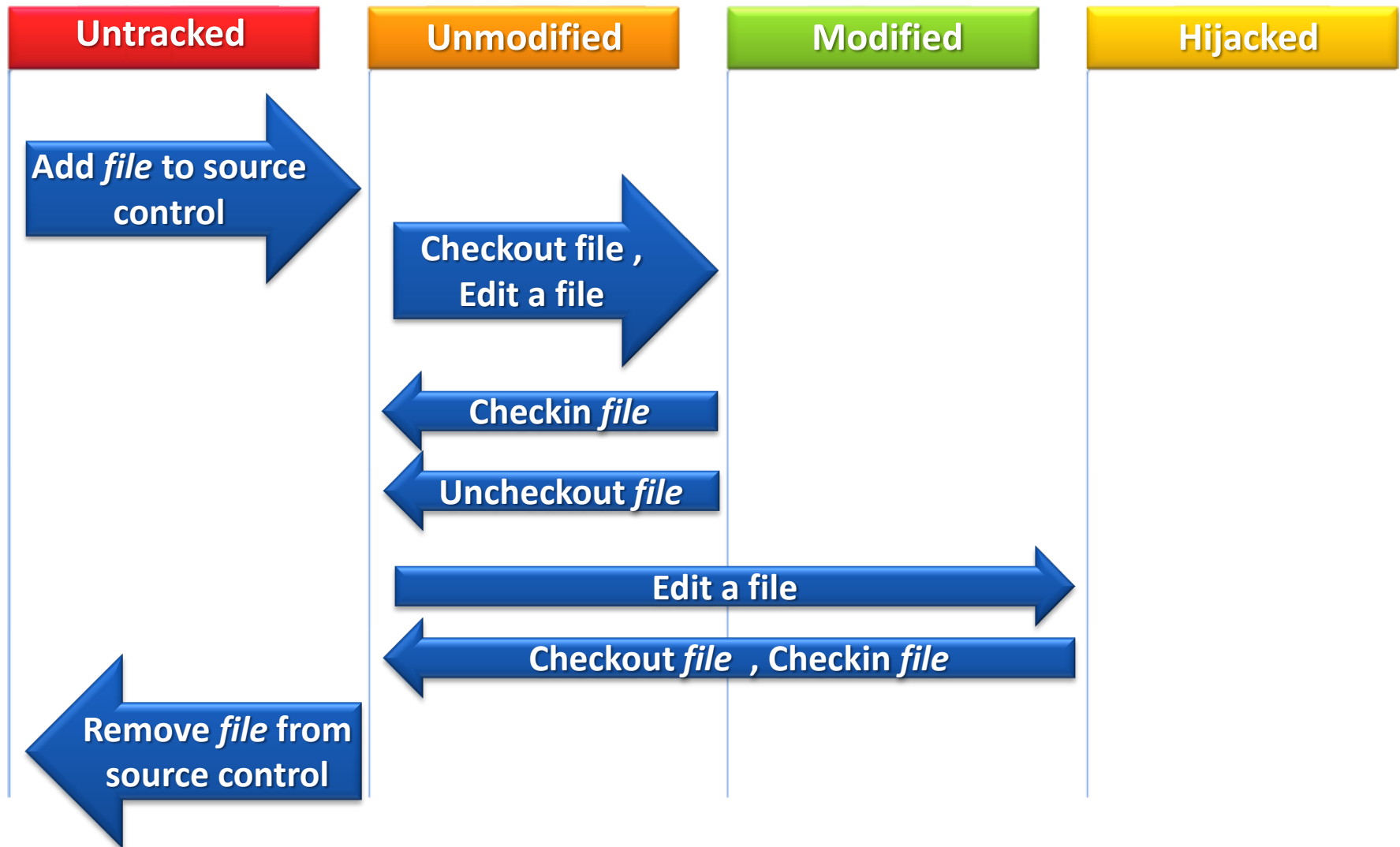


Define your company
workflow and policy

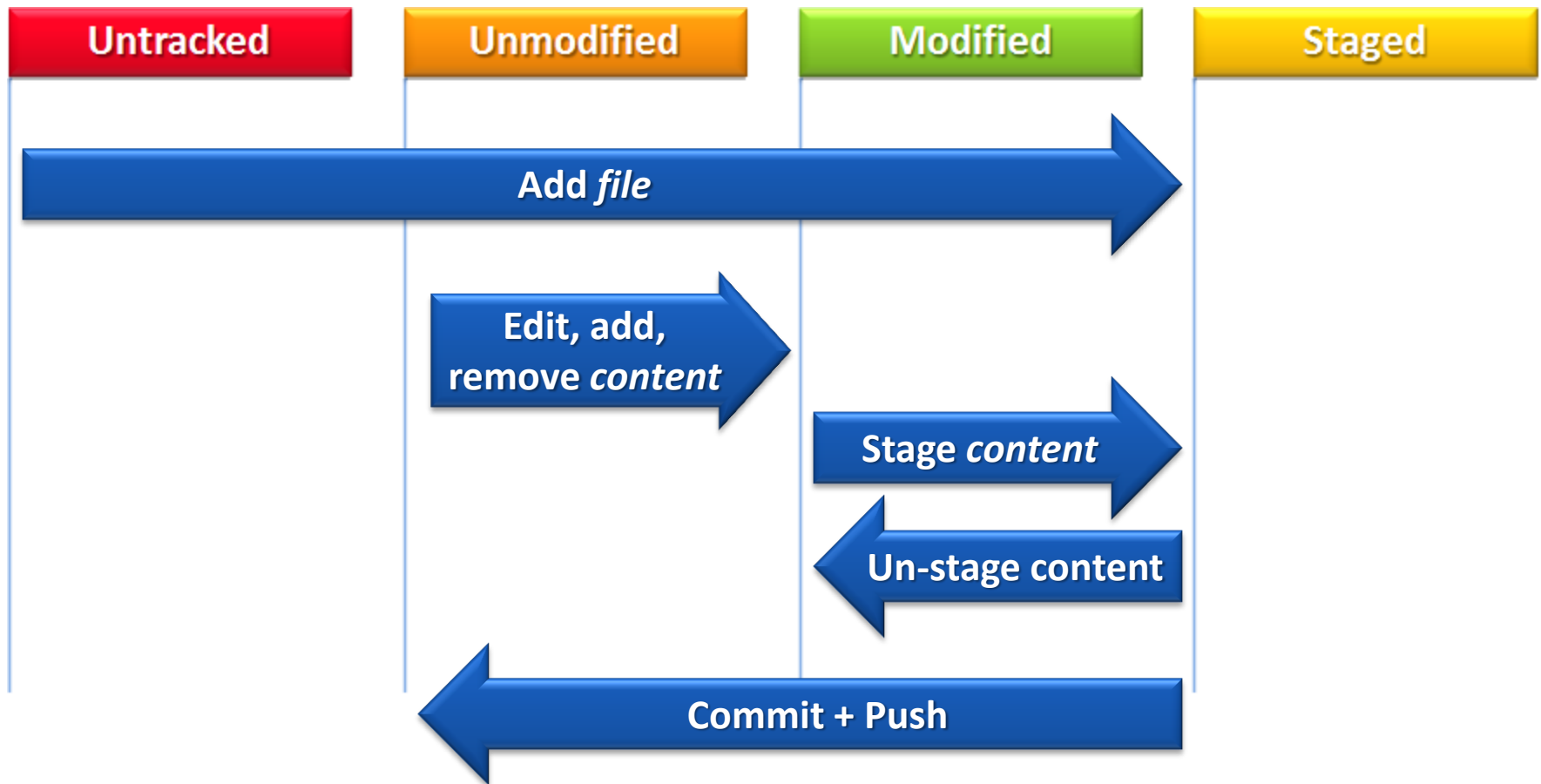
+

Publish your
Git cheat-sheet

CC: file Lifecycle



Git: content Lifecycle



CC checkout CC vs. Git

- Checkout CC
 - Define the intention to change a file / **LOCKING**
 - **Reserved**: exclusive lock – nobody can checkout the same file
 - **Unreserved**: shared lock – others can perform an unreserved checkout
- Checkout Git
 - Locking **DOES NOT EXIST** (Git is peer-to-peer)
 - Operations are **made LOCALLY** (1 user = locking not needed)

... what then is Git Checkout for ?

Restore files in the working tree to the archived version

Move to a different version / branch

CC: files updates

Not applicable when directories / files are checked out

- Update modes
 - **OVERWRITE**
Replace hijacked contents with the latest version
 - **NOVERWRITE**
Leaves unchanged hijacked files
 - **RENAME**
Renames hijacked files (*fileexample.keep*)



Git: content stash, pull and rebase

1. Git STASH

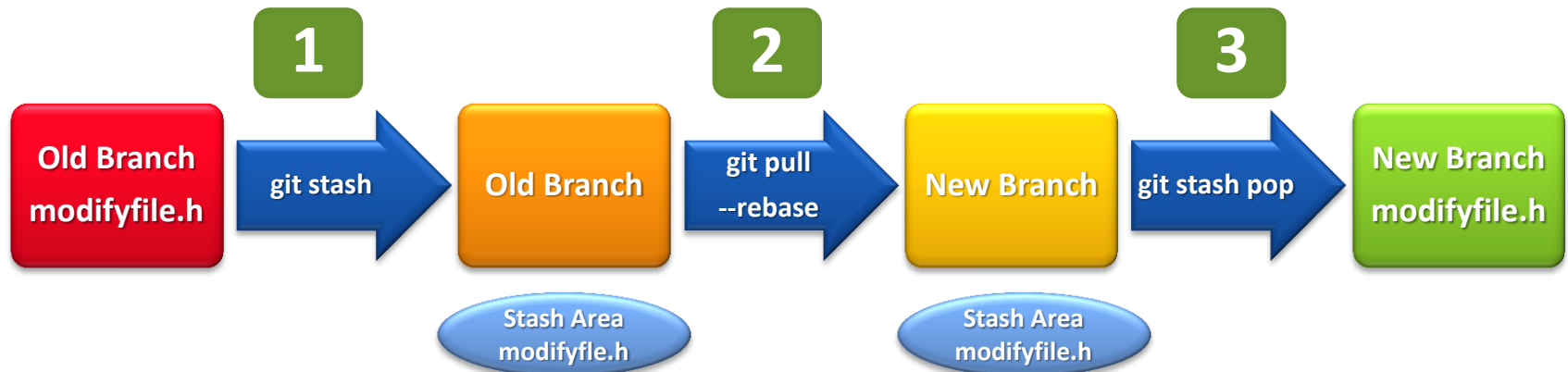
- Local changes are put into a temporary area

2. Git PULL (with rebase / with merge)

- Fetches the contents from the remote repository and rebase / merge the local branch

3. Git STASH POP

- Restore and merge local changes (*conflicts to be resolved manually*)



CC: files checkin

- Checkin creates a new version of a file
- Checkin of reserved / unreserved checkouts
 - **SUCCEEDS** when
 - You are the owner of reserved checkout.
 - All checkouts unreserved, and no one has checked in.
 - **FAILS** when
 - Another user have reserved checkout.
 - Another user checked in a successor version before you.

Git: content add, commit and push

1. Git ADD

- Adds your contents to the "staging area" for commit

2. Git COMMIT

- Check-in ALL the staged content to the local repository

3. Git PUSH

- Synchronize all the commits to the remote repository.



POLL: What is your ClearCase vs. Git adoption ?

- High level assessment of alternatives
- Training on Git or other Version Controls
- Using Git and ClearCase side-by-side
- Started ClearCase to Git migration
- Completing ClearCase to Git migration



Migrating to Git

Strategies, tools and issues in the ClearCase migration to Git

Migration overview and approach

① Manage costs & benefits vs. expectation

- ✓ Evaluate options
- ✓ assess costs and benefits
- ✓ Talk to Teams and manage expectations

② Plan and automate

- ✓ **VOB-2-Git converter does not exist**
- ✓ Transition period to be planned and managed
- ✓ Script the migration and automate in small chunks

③ Track and assess risks and issues

- X **Big VOBs** could be problematic
- X Git **file size limited to 2GB** (32bit systems)
- X Different **concurrency model**

There is no magic tool !

ClearCase cannot be mapped 1:1 to Git

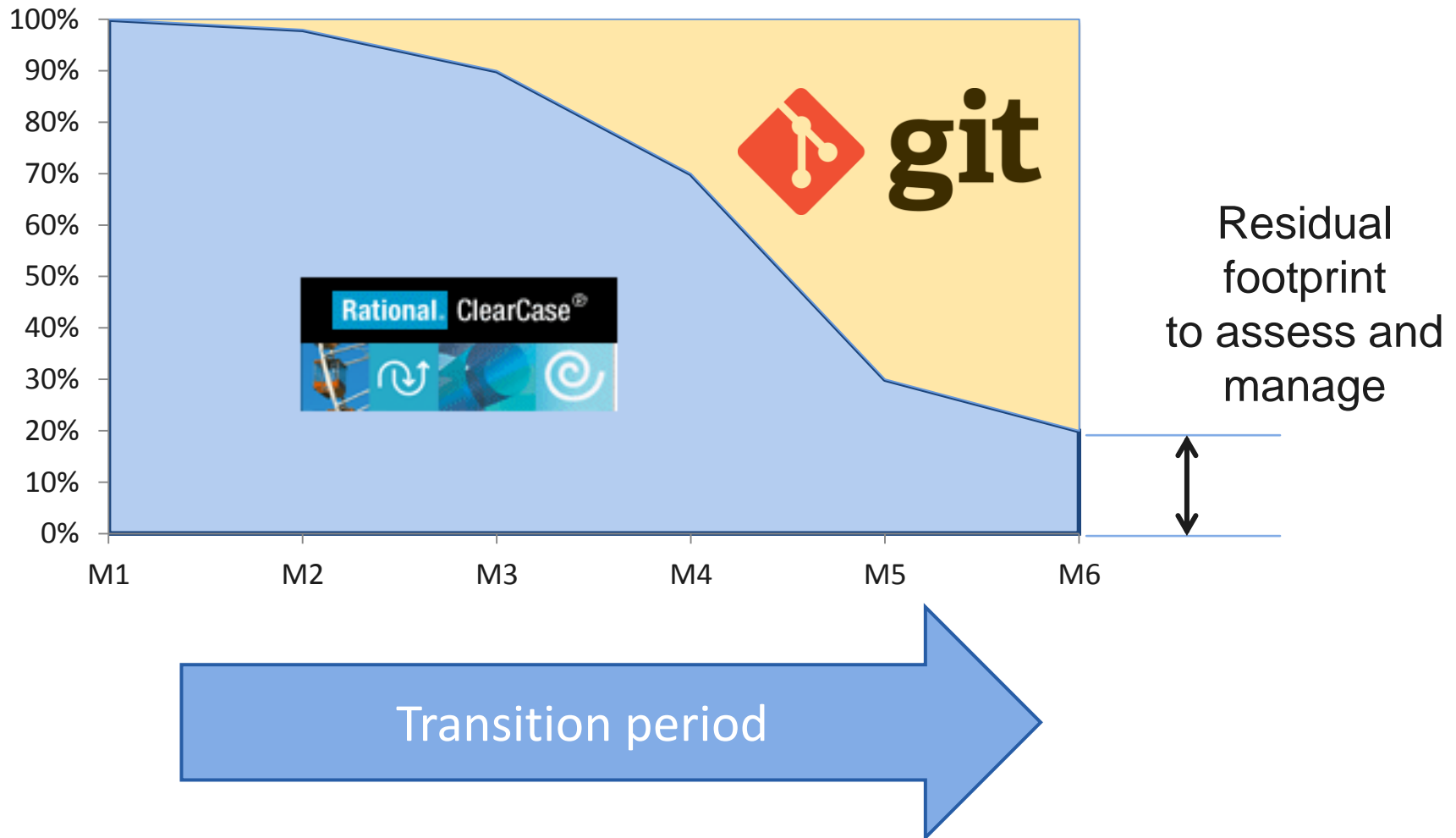
Option 1 – Bridge through Subversion

- CC to SVN (SVN Importer)
- SVN to Git (\$ git svn clone <http://svn/repo/here/trunk>)
- Keep SVN for full history backup

Option 2 – Git from ClearCase baseline

- Check out baseline from ClearCase → Continue on Git
- Leave ClearCase confined to **maintenance / archive**

Planning



Problems and workarounds

- File Size
 - Files are **compressed in memory** (2 GB on 32-bit systems)
 - Binary files **increase the repo size**
 - **Solutions and workarounds:**
 - Store big / binary files in **Subversion** or **binary repositories**
 - **git-annex / git-bigfile** - binary files outside Git
 - **Bup** (<https://github.com/bup/bup>) - client-based script on top of Git repo
- Repository size vs. bandwidth
 - Provide **Git snapshot bundles over HTTPS**
 - Use **Gerrit Code Review** replication
 - Split into **Git sub-modules**

Authentication / Protocol Security

- ClearCase protocol
 - Proprietary RPC protocol
 - User always authenticated via OS (and thus via LDAP)
 - **NOT Internet-friendly (requires VPNs)**
- Git protocol
 - **SSH, HTTPS** or TCP/Git (discouraged)
 - User authenticated via OS (discouraged) or OpenID / LDAP / External (i.e. Gerrit with CollabNet TeamForge)
 - **Works on the Internet** / works in **Disconnected** mode
 - **Fully audited, X.509 and Strong Encryption / Authentication**

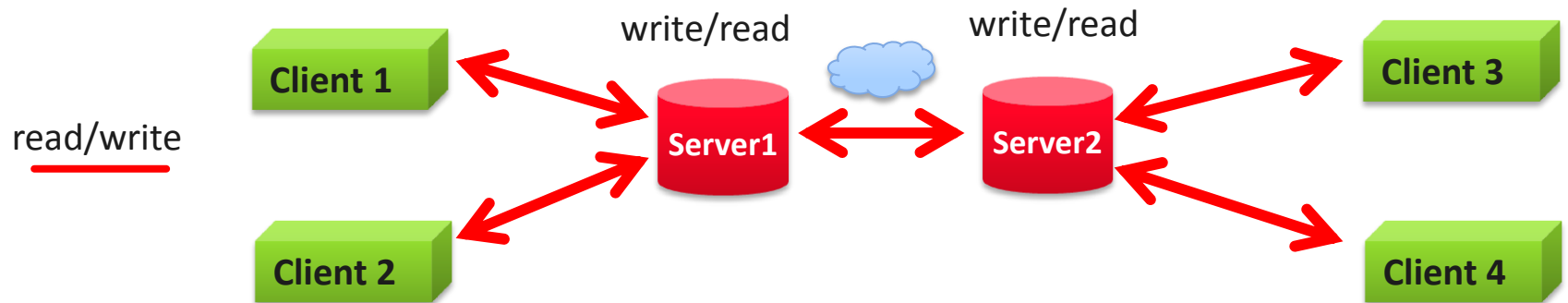
Security

Access Control

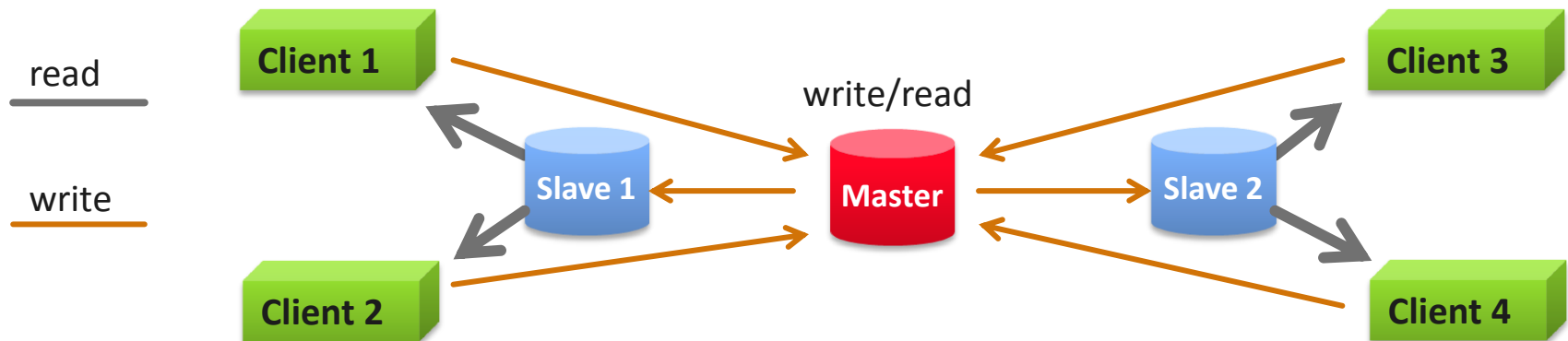
- ClearCase is OS-level permission scheme
 - File-system VOBs based security
 - RWX scheme on a per Owner / Group / Others
 - Assign ACLs to files
 - ClearCase Meta-Data are objects for permissions (Projects, Activies)
- Git/Gerrit is RBAC (as CollabNet TeamForge)
 - Project / Branch / Name-space based security
 - Fine-grained Git scheme on a per role-based
 - **NO file-based Access Control**
 - **Meta-data Access Control driven by CollabNet TeamForge**

Replication strategy

ClearCase - multi-site



Git - master-slave



Summary

- ClearCase is very different from Git
- File vs content snapshot for version tracking
- Branches strategy to be defined and agreed
- Migration strategy and problems
- Security guidelines
- Replication strategy



CollabNet for ClearCase > Git Migration

032453

**users switched from
ClearCase to TeamForge**

Leading telco provider – 11,000 users

Large retailer – 1,950 users

Top logistics company – 2,000 users

Mobile technology provider – 138 users

Semiconductor manufacturer – 3,100 users

Global hi-tech leader – 9,000 users

Bio-technology innovator – 750 users

European hi-tech firm – 1,200 users

Healthcare leader - 500 users

Banking provider - 200 users

What Developers Love about Git



- 65% of 'Git using' organizations like 'powerful branching & merging' (and 57% like this the most)
- 57% of 'Git using' organizations like 'network performance' (and 22% like this the most)
- 32% of 'Git using' organizations like 'Git's popularity' (and 10% like this the most)

People love Git for its powerful branching and merging capabilities and solid network performance.

Source: InformationWeek Survey 2013, State of Git in Enterprise, <http://visit.collab.net/git-survey.html>

Enterprises continue to have reservations about Git

performance
handling binaries
administration
submodules
commit and syntax

security*



*51% polled believe security needs improving

Anybody (Git using orgs):

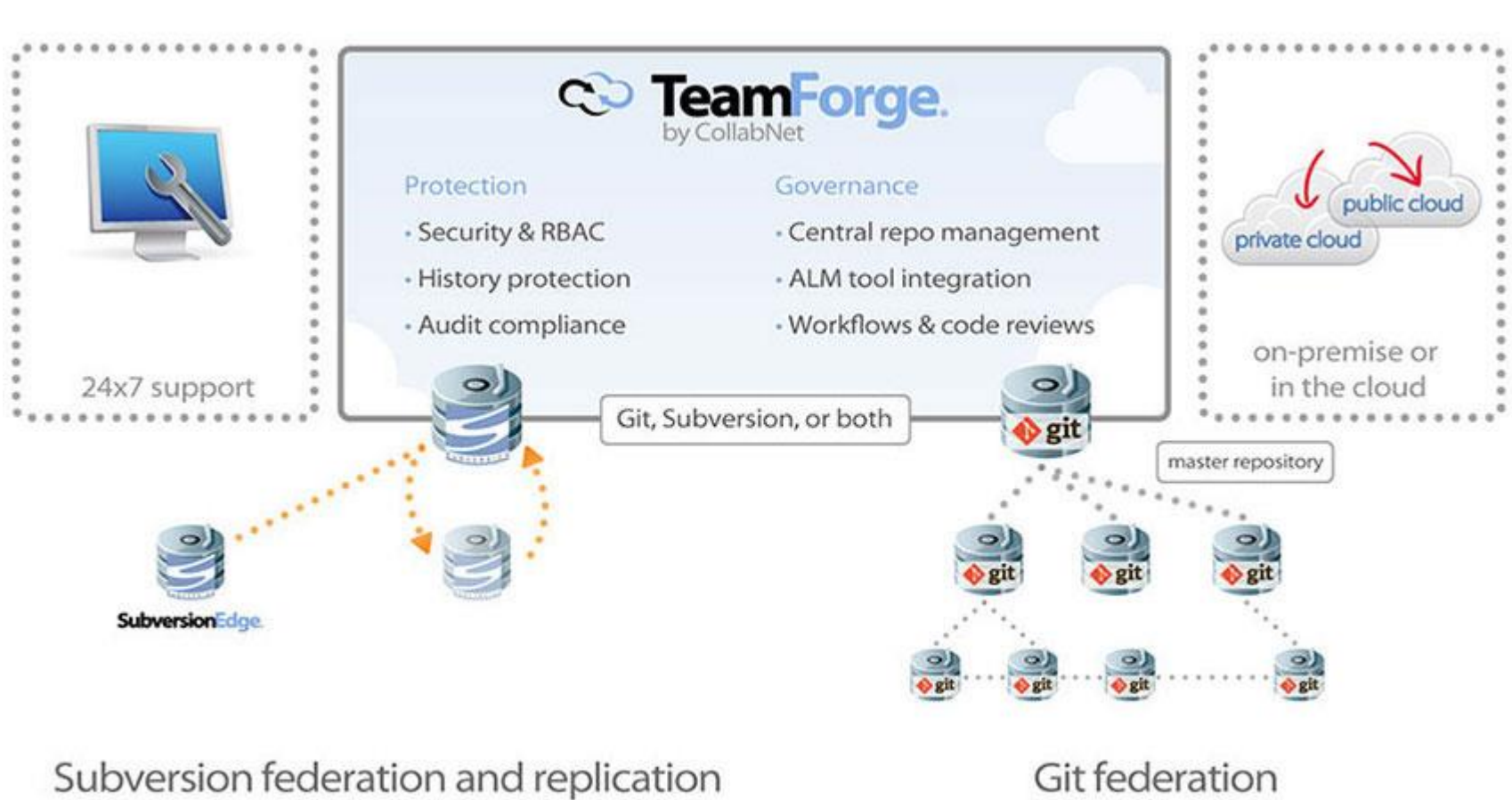
- 37% Tool integration (trackers, CI/CD, ...)
- 36% Co-existence with other SCM tools
- 33% Security, traceability, process governance
- 16% 24/7 Support and SLA's

Managers and Above (Git using orgs):

- 51% Security, traceability, process governance
- 49% Tool integration
- 37% Co-existence with other SCM tools
- 22% 24/7 Support and SLA's

Source: InformationWeek Survey 2013, State of Git in Enterprise, <http://visit.collab.net/git-survey.html>

Manage Git and Subversion with One Enterprise Platform



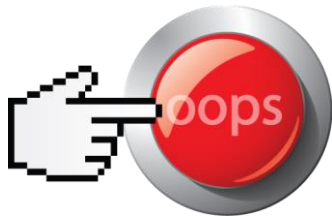
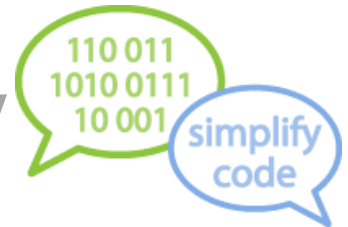
Why TeamForge for Enterprise Git – The Six Reasons

Git + Subversion – Integrate or Mitigate



Tool Integration – Application Lifecycle

Dev Productivity – Code Reviews & Discovery



History protection – “Un-do” Any Commit

Access Control – Protect Your IP



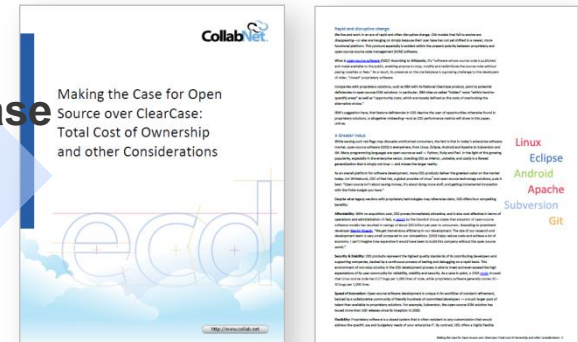
24/7 Support + Services – Enterprise-Ready Git

Learn More

WHITEPAPER: The Case for Open Source over ClearCase

[http://visit.collab.net/
MakingtheCaseforOpenSourceoverCCandTCO_WP.html](http://visit.collab.net/MakingtheCaseforOpenSourceoverCCandTCO_WP.html)

Free 5 page executive whitepaper



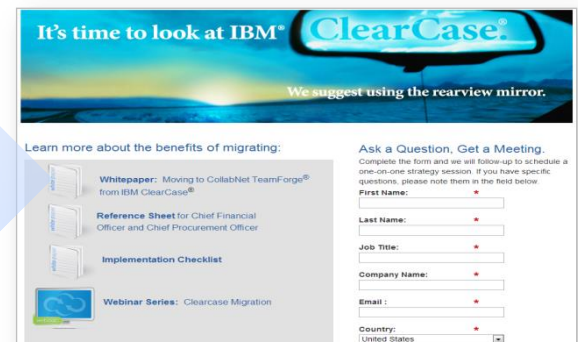
TeamForge for Enterprise Git

<http://www.collab.net/products/teamforge/git-for-the-enterprise>



WORKSHOP: Plan your Migration Strategy

<http://www.collab.net/pullahead>



Go Agile with Git, Part 1 of 3: Workflows, Branching & Merging

Tuesday, January 15, 10:00 AM – 11:00 AM PST »

Go Agile with Git, Part 2 of 3: Peer Programming & Code Reviews

Tuesday, January 29, 10:00 AM – 11:00 AM PST »

Go Agile with Git, Part 3 of 3: Hands-On Lab with Gerrit & Jenkins

Tuesday, February 12, 10:00 AM – 11:00 AM PST»

Questions?

Luca Milanesio

luca@gerritforge.com

www.collab.net

+1-650-228-2500

+1-888-778-9793

 blogs.collab.net

 twitter.com/collabnet

 www.facebook.com/collabnet

 www.linkedin.com/company/collabnet-inc

About CollabNet

CollabNet is a leading provider of Enterprise Cloud Development and Agile ALM products and services for software-driven organizations. With more than 10,000 global customers, the company provides a suite of platforms and services to address three major trends disrupting the software industry: Agile, DevOps and hybrid cloud development. Its CloudForge™ development-Platform-as-a-Service (dPaaS) enables cloud development through a flexible platform that is team friendly, enterprise ready and integrated to support leading third party tools. The CollabNet TeamForge® ALM, ScrumWorks® Pro project management and SubversionEdge source code management platforms can be deployed separately or together, in the cloud or on-premise. CollabNet complements its technical offerings with industry leading consulting and training services for Agile and cloud development transformations. Many CollabNet customers improve productivity by as much as 70 percent, while reducing costs by 80 percent.

For more information, please visit www.collab.net.



CollabNet, Inc.

8000 Marina Blvd., Suite 600
Brisbane, CA 94005

www.collab.net

+1-650-228-2500

+1-888-778-9793

 blogs.collab.net

 twitter.com/collabnet

 www.facebook.com/collabnet

 www.linkedin.com/company/collabnet-inc

© 2012 CollabNet, Inc., All rights reserved. CollabNet is a trademark or registered trademark of CollabNet Inc., in the US and other countries. All other trademarks, brand names, or product names belong to their respective holders.