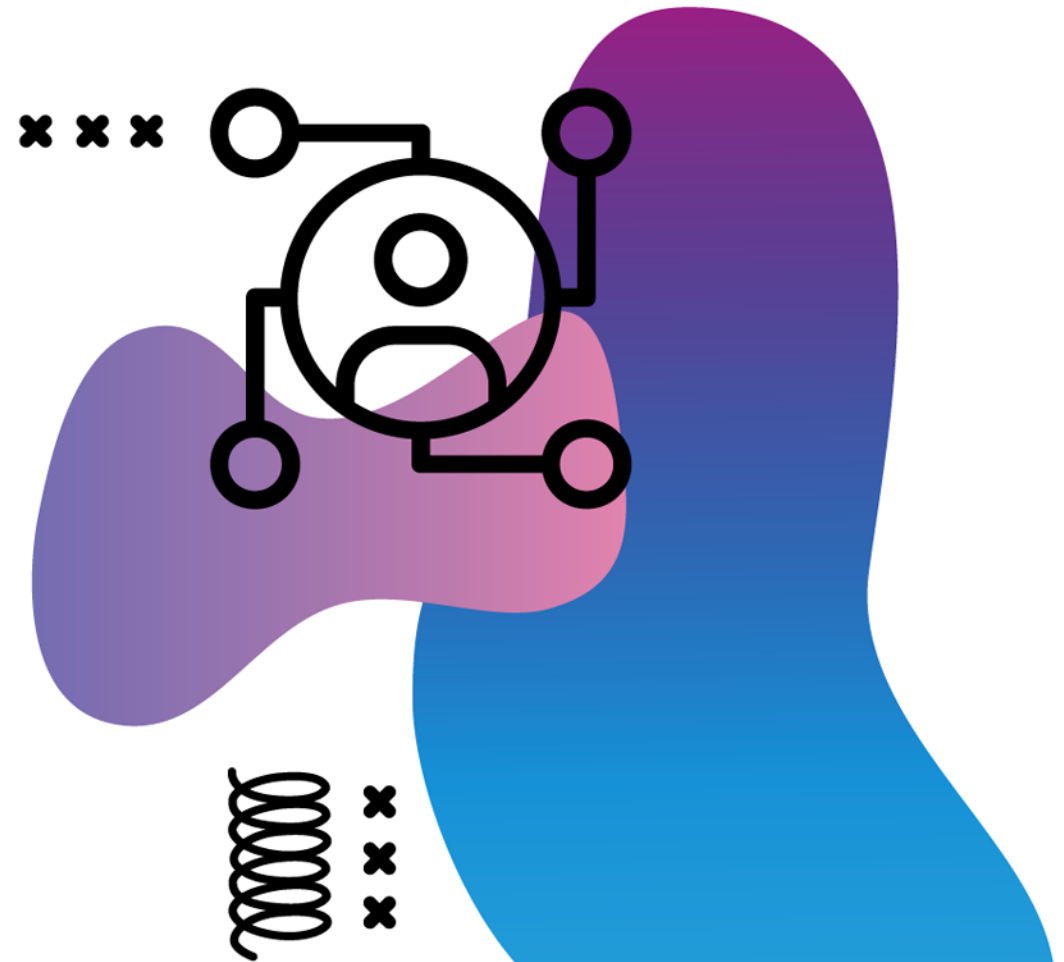


Indonesia

Making the best out of Git

Jon Reid, Kamila Duniec

PhUSE Single Day Event – BASEL – 28 June 2018



More potential –
Expanding
opportunities

In Scope

- Idorsia reporting environment
- Git key features and differentiators
- Git key disadvantages
- Git workflows at Idorsia
- Automation
- Imagine the future

Out of Scope

- Why Version Control
- Git history and basics

Disclaimer

- The content is based on early experience not years of exposure and deep expertise
- The opinions presented are of the authors and not of Idorsia

Idorsia reporting environment

How did we get it up and running within 7 months?

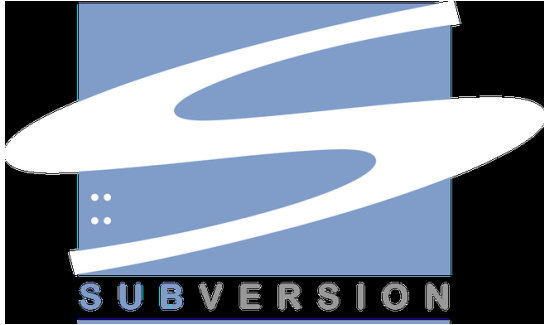
Game-changing factors:

- Assess in-house capacity - get external experts (IT PM, OCS)
- Prioritize roll-out over additional components and features
- Scope relevant to Idorsia's case
- Use cases based on system scope, no extensive documentation (URS, bp etc.)
- Team empowerment to take decisions
- Team up with IT and QA



**THINK WIDE – ACT
NARROW**

Business plans without IT input



Conclusion:

For our use subversion is better suited:

- More intuitive
- Centralized history in SVN compared with Git
- Branching and co-development is not needed

Git key features and differentiators



Key Features:

- Open source (free and globally used)
- Compare versions of code, revert if needed
- Audit trail (who made what changes and when)
- Scalability
- Remote work
- IT - experienced with Git, source of good advice
- history of a project/file
- Ignore certain files (*.sas7bdat, *.log, *.lst)
- Tags
- Command line, GUIs, application integration
- Managing conflicts (2 people update the same code)

More:

- Hashing algorithm - difficult to corrupt the

Git key disadvantages

- Open-source - no owner in case of bugs/issues
- To customize one needs to develop their own GUI to pre-set and limit the functionalities. Need to control by SOPs and Work Instructions
- SAS Programmers in pharma do not have much experience with Git

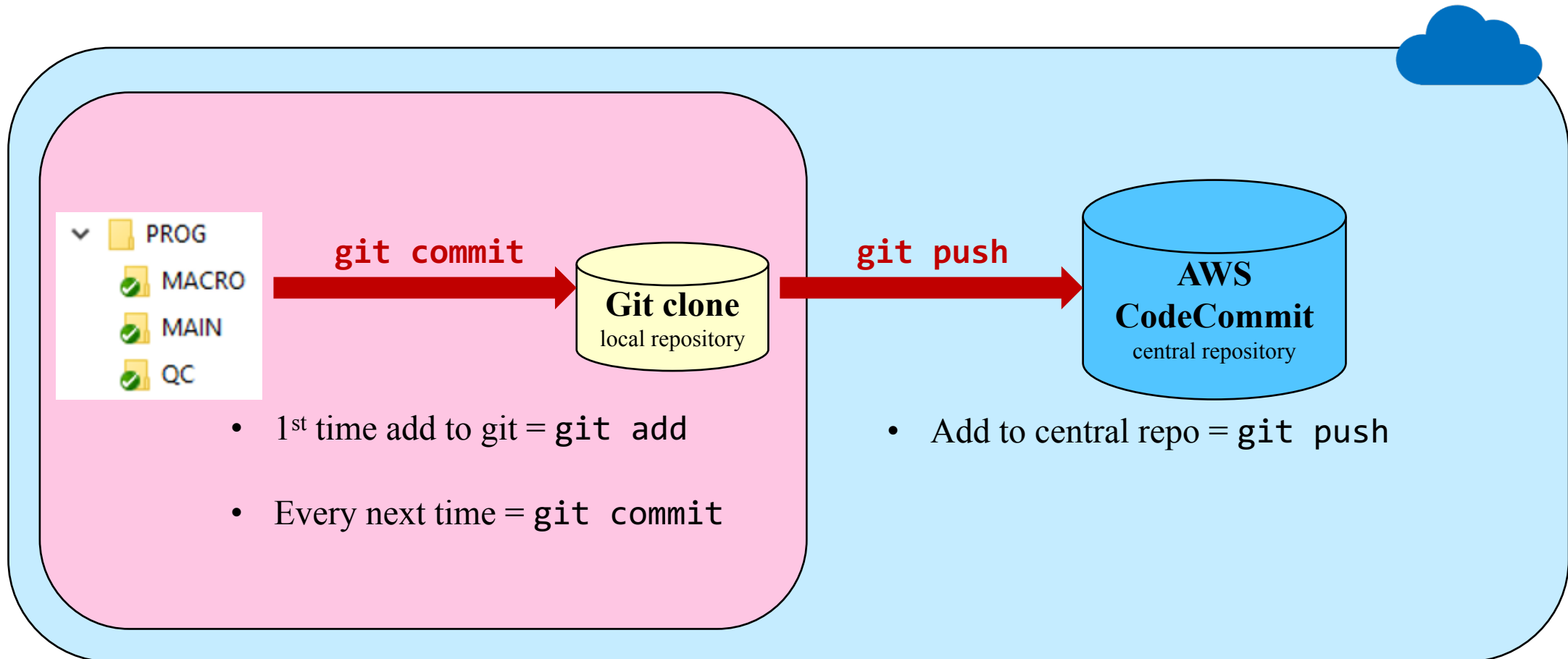
Git workflows at Idorsia

- How to design the workflow?
 - Habits from past systems and processes
 - Team size (concurrent programmers)
 - Complexity of dependencies
 - Frequency of use (expected)

Group	Git workflow
BIostatistician S	zero solution: no version control
DM Programmers	simple solution: one clone for the project SDTM programs
Stat. Programmers	team solution: project clone and user clones

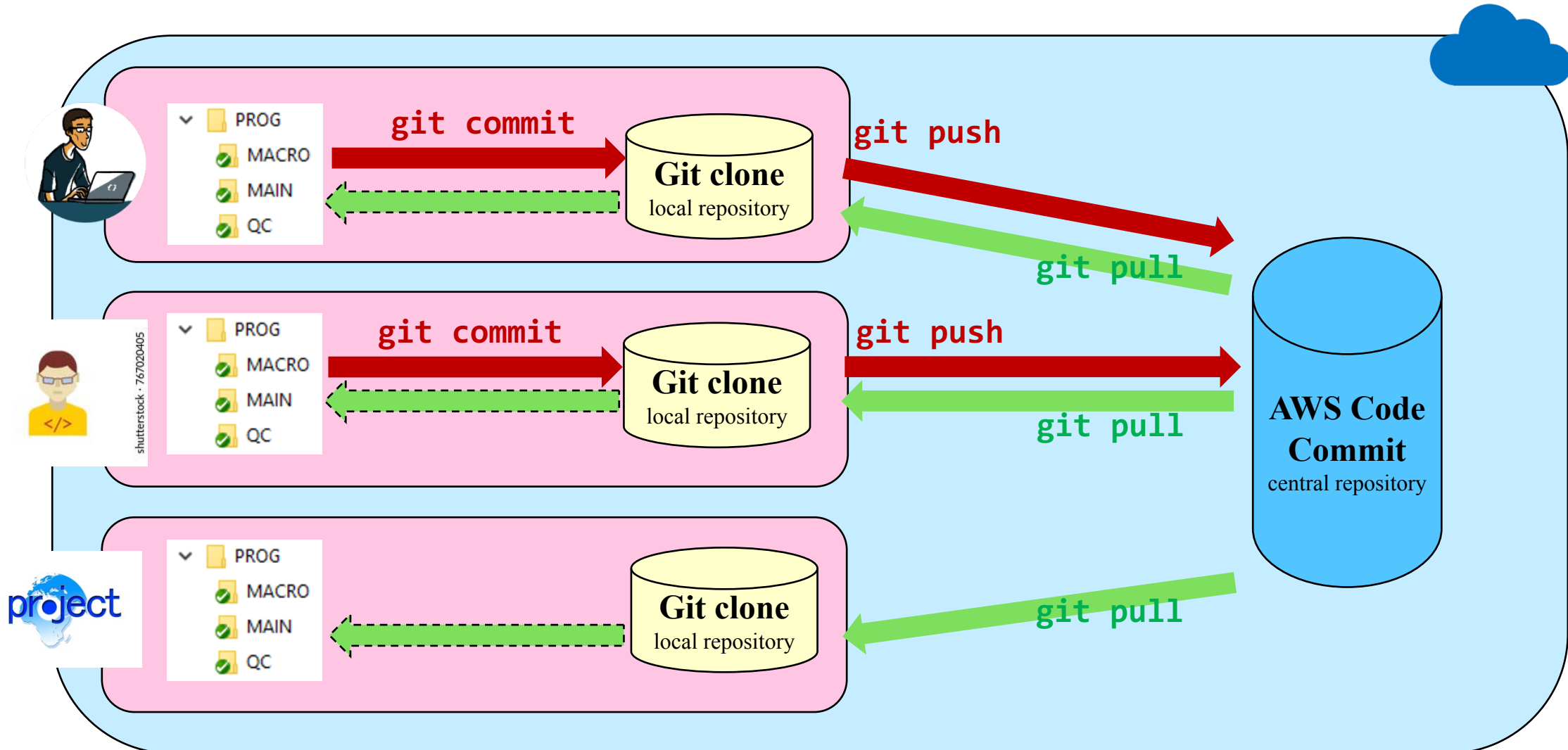
Versioning with Git

Simple Solution

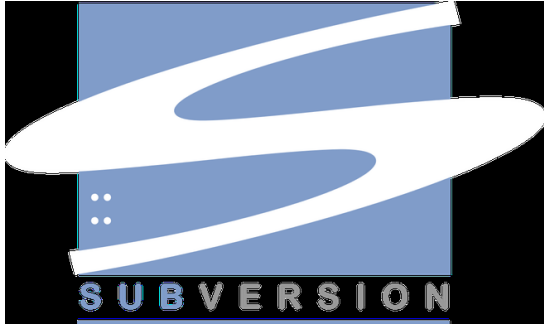


Versioning with Git

Team Solution



If we were deciding now?



Conclusion:

Git, in spite of the learning curve, proved to be a good solution for us:

- Git allows you to be flexible with how you set up your workflow and change whenever best for the project
 - For example different outsourcing models
- Git allows programmers to work in a separate environment, which they can control, but which can be synchronized with the project area

Automating the Git workflow

PhUSE 2017

Paper TT03

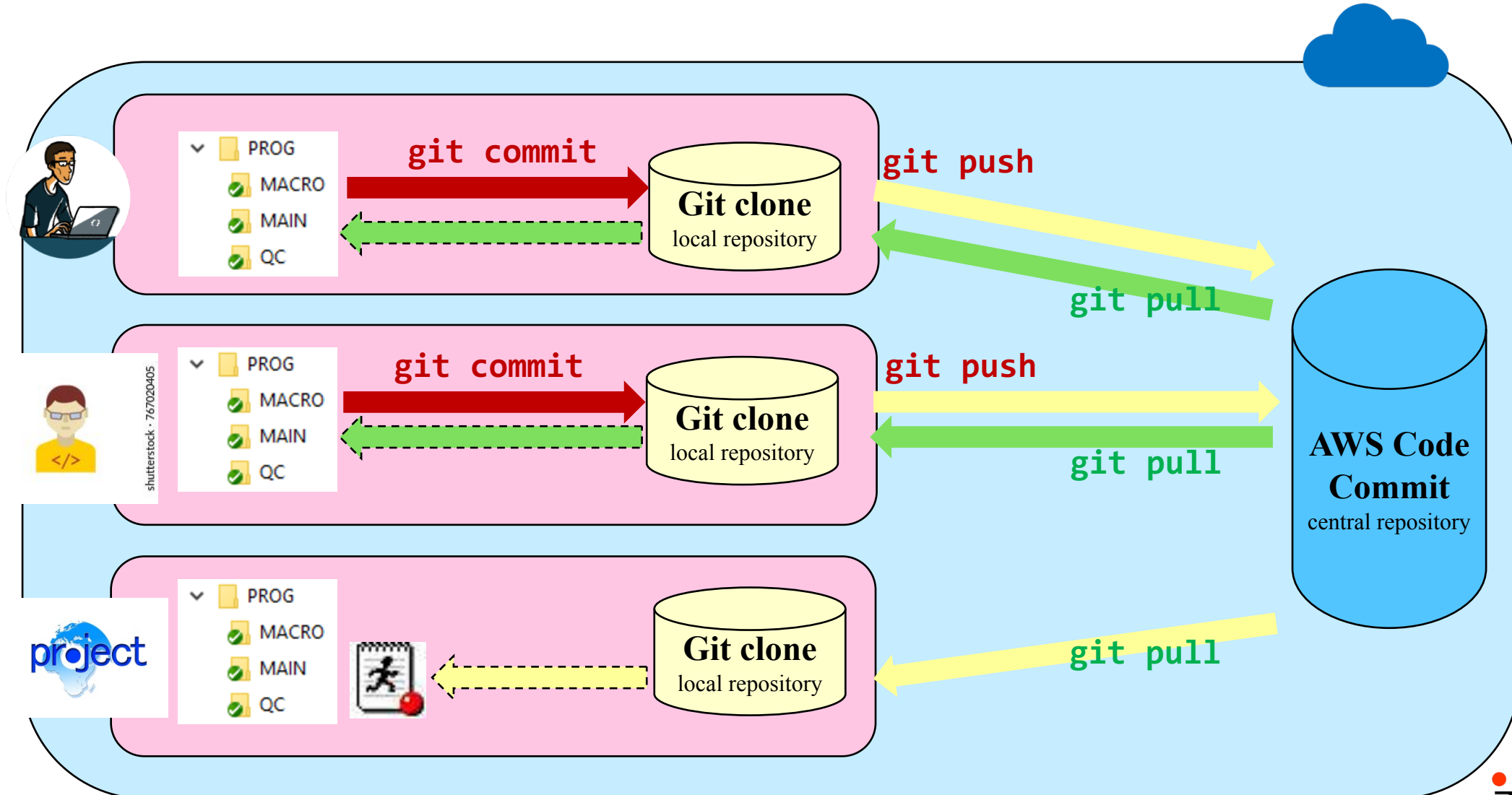
Catching Up: Continuous Integration Pipelines For Clinical Analysis

Andrew Karpow, Bayer AG, Berlin, Germany



Jenkins

Automating the Git workflow



Imagine the future



- ? Automate some more...
- ? Use Git for distributed development to develop SAS code remotely by contractors, FTEs, vendors
- ? Branching and merging
 - For co-development of standard macros
 - For similar deliverable (publications)
- ? Master programmer peer review and sign-off

- Re-assess after a year a of use

Acknowledgements

- Idorsia IT team
- Idorsia Biometry colleagues

AOB?

Library of Links

- [What is version control and what is Git](#)
- [Git Best Practices](#)

- [2017 - Git with It!](#)
- [2017 - A User Guide for the PhUSE GitHub Repository](#)
- [2016 - Git and Github](#)
- [2014 – Check it out! Versioning in SAS® Enterprise Guide®](#)
- [2013 - Hitchhiker's Guide to GitHub](#)
- [2012 - An Introduction to Git Version Control for SAS Programmers](#)

- http://www.sascommunity.org/wiki/Version_control

GIT CHEAT SHEET

```
git init Initialize a Git repository.
git status Shows all the changes that have occurred in your repository.
git add <file> Adds a file to the index/staging area.
git commit Creates a commit.
-a adds changes to the index
-m "message" specifies a message
git pull --rebase
git log Shows the history of your repository.
git diff Compare the changes in your working tree against the staging area.
git diff HEAD Compare your working tree including staged changes to repository.
git diff -cached view the changes in the index/staging area to the repository.
.gitignore Text file that specifies files for Git to ignore.
git rm -cached <file> remove a file previously added to the repository.
git tag Lists the tags.
<tagname> Create a tag in the repository.
git checkout tags/<tag name> Checks out the tag into your working tree.
git branch List the branches.
<branch name> Create the branch.
-d <branch name> Delete the branch.
git checkout <branch> Copy the code in that branch to your working tree to work on.
-b Create a new branch from the current branch and checks it out.
git merge <other branch> Merge <other branch> into the CURRENT branch.
git clone <repository URL> Create a local copy of a remote repository.
```

Tortoise Git

<https://tortoisegit.org/>

The screenshot shows the TortoiseGit Log Messages window. At the top, it displays the current branch as 'master' and filters for messages from '2008-11-05' to '2013-10-08' by 'Authors'.

The main area shows a commit graph on the left and a list of commit messages on the right. The selected commit message is: "Make tgit protocol handler include a command".

Below the messages, the commit details are shown, including the SHA-1 hash: c44d398b2cfb311202b3eb5fbb0ab07adf198880. The commit message is repeated: "* Make tgit protocol handler include a command". Below this, it says "Now tgit-links work as follows:" and provides the URL: "tgit://clone/https://code.google.com/p/tortoisegit/". The author is listed as "Signed-off-by: Sven Strickroth <email@cs-ware.de>".

At the bottom, a table shows the files changed in this commit:

Path	Extension	Status	Lines added	Lines removed
src/Changelog.txt	.txt	Modified	1	1
src/TortoiseGitSetup/StructureFragment.wxi	.wxi	Modified	1	1
src/TortoiseProc/TortoiseProc.cpp	.cpp	Modified	5	1

At the bottom of the window, it shows "Showing 6969 revision(s), from revision 60636b3 to revision 61f6d1c - 1 revision(s) selected, 0 file(s) selected". There are checkboxes for "Show Whole Project" and "All Branches", a search box for "Filter paths", and buttons for "Refresh", "Statistics", "Walk Behaviour", "View", "Help", and "OK".

Tortoise Git

Icon overlays



normal



assume-valid



added



normal.cpp



assume-valid.cpp



added.cpp



modified



deleted



ignored



modified.cpp



deleted.cpp



ignored.cpp



conflicted



skip-worktree



non-versioned



conflicted.cpp



skip-worktree.cpp



non-versioned.cpp

Tortoise Git – resolving conflicts

The screenshot displays the TortoiseGitMerge application window for a file named MERFE0B.tmp. The interface is divided into three main sections for comparing and merging code:

- Theirs - MERFE0D.tmp:** Shows the code from the other branch. Line 519 is highlighted in red, indicating a conflict where the value is '1'.
- Mine - MERFE0B.tmp:** Shows the code from the local branch. Line 519 is highlighted in red, indicating a conflict where the value is '3'.
- * Merged - MERFEE9.tmp:** Shows the result of the merge. Line 519 is highlighted in green, indicating it has been successfully merged with the value '1'.

The top toolbar includes various editing and navigation tools such as Save, Undo, Redo, Copy, Paste, Find, Goto Line, Mark as resolved, Previous difference, Next conflict, Previous inline difference, Next inline difference, Use left block, Use left text block, Show Whitespaces, Compare whitespaces, Ignore whitespace changes, Ignore all whitespace changes, Inline diff, Regex Filter, Ignore Comments, View Bars, Wrap long lines, and View.

At the bottom, the status bar indicates the current view settings: Left View: ASCII ▾ LF ▾ Tab 4 ▾ - 1 / + 1, Right View: ASCII ▾ LF ▾ Tab 4 ▾ - 1 / + 1, Bottom View: ASCII ▾ LF ▾ Tab 4 ▾ - 1.

Sourcetree

<https://www.sourcetreeapp.com/>

Graph	Commit	Author	Description	Date
	b7358c7	Rahul Chha...	master origin/master origin/HEAD Removing ol...	Mar 3, 2016, 11:...
	bdb8bef	Rahul Chhab...	Merged in update-google-verification (pull request #14)	Feb 18, 2016, 1:3...
	dfe975d	Tyler Tadej...	origin/update-google-verification Update google verificati...	Feb 11, 2016, 2:2...
	3bc3290	Tyler Tadej...	Replace outdated Atlassian logo in footer with base-64 en...	Feb 11, 2016, 2:1...
	dba47f9	Tyler Tadej...	Add gitignore	Feb 11, 2016, 1:3...
	ff67b45	Mike Minns...	Updated Mac min-spec to 10.10	Feb 15, 2016, 11:...
	72d32a8	Michael Min...	Merged in hero_images (pull request #13)	Feb 15, 2016, 10:...
	246c4ff	Joel Unger...	origin/hero_images hero_images Used Tinypng to c...	Feb 11, 2016, 3:3...
	9d9438c	Joel Unger...	Replacing hero images with new version of SourceTree	Feb 9, 2016, 2:59...
	ce75b63	Michael Min...	Merged in bug/date-https (pull request #12)	Feb 15, 2016, 10:...
	85367bb	Patrick Tho...	origin/bug/date-https fixed date and https errors	Jan 7, 2016, 12:2...
	4f9b557	Joel Unger...	New Favicon	Feb 8, 2016, 3:55...
	384e6d5	Rahul Chhab...	origin/search-console-access search console google ver...	Feb 3, 2016, 2:09...
	6fa47a9	Mike Minns...	updated to move supported version to OSX 10.9+	Dec 15, 2015, 2:0...
	8dd87bb	Mike Minns...	remove extra , when a line is skipped due to empty server	Nov 23, 2015, 2:2...
	faa195e	Mike Minns...	Skip records with empty server/user id as gas rejects them	Nov 23, 2015, 2:1...
	0cdf96	Mike Minns...	corrected paths after merge	Nov 23, 2015, 2:0...
	051ab1b	Mike Minns...	corrected column counting	Nov 23, 2015, 1:5...
	a723bc2	Mike Minns...	Merge branch 'au2gex'	Nov 23, 2015, 1:5...
	65fd580	Mike Minns...	deal with invalid instanceids	Nov 23, 2015, 1:5...
	500a892	Michael Min...	Merged in au2gex (pull request #11)	Nov 23, 2015, 1:0...