





הנדסת תוכנה 6. בקרת תצורה git / github

Pragmatic Programmer Tip:

Always Use Source Code Control

Source code control is a time machine for your work—you can go back.

השבוע

- Version Control דקרת תצורה \ קוד
 - git / github כלים:
 - משימה אישית 3
 - ZFR סקר •
- MVP 1 פרויקט 6: תחילת סבבים• פרויקט 6: תחילת סבבים
 - תרגיל סקר ZFR (+ השלמת back-end)



בקרת גרסאות קוד - מקורות

- Sink, Version Control by Example
- Google Tech Talk: <u>Linus Torvalds on git</u>
- Spolsky, Hg Init: a Mercurial tutorial http://hginit.com
- Sink, Source Control HOWTO
 http://www.ericsink.com/scm/source_control.html
- Intro to Distributed Version Control <u>http://betterexplained.com/articles/intro-to-distributed-version-control-illustrated/</u>

VCS Links

- Eric Raymond on vcs
- The 10 commandments of good source control management, blog 2011
- FogBugz and Kiln <u>video</u>, 2011
 - DVCS University Slides (*)
- Spolsky, Hg Init: a Mercurial tutorial
- Azad, <u>A Visual Guide to Version Control</u>
 - Intro to Distributed Version Control (Illustrated)
- Eric Sink, Source Control HOWTO

Git Links

- http://git-scm.com/ (getting started)
- Set Up Git (Win), <u>ssh issues</u>
 http://help.github.com/win-set-up-git/
- Windows client list, O'Reilly Webcast:
 Git in One Hour

se16b-ya<mark>g</mark>el

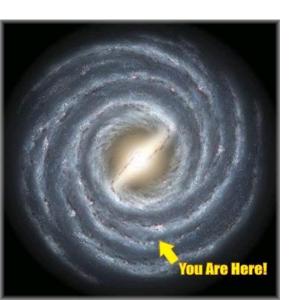
More Git / Github Links

- Git branching with git-flow, Heb. Video, 2013
- Videos: Git For Ages 4 And Up, Git Going(oredev'12), Think like a Git
- <u>learn.github.com/</u>, <u>try.github.com/</u>
 <u>Gitflow</u>
- saastv: <u>Using Branches with Git</u>
- no branches at flickr
- Insider Guide to GitHub (video)
- Articles: <u>article</u> (including .gitignore), <u>post</u>, <u>difficulties</u>,



איפה אנחנו בפרויקט (בקורס)?

- למה? בעיה (פלט: הצעת פרויקט\חזון\SOW)
- מי? צוות (Inception, אתחול\תכנון פרויקט)
 - מה? דרישות (SRS)
 - איך? תיכון (ארכיטקטורה) (SDS)
 - מתי? תכנון וניהול – (ZFR)
 - בניה (סבבי פיתוח)



בקרת תצורה (SCM)

- לפרויקט תוכנה תוצרים שונים:
- מסמכי דרישות ותיכון, קוד, executables, מדריכי שימוש, בדיקות,...
 - פרויקט תוכנה משתמש בכלים שונים:
 - ..., עורכים, צד ג', שת"פ, (מ"ה), מהדרים, עורכים, צד ג', שת

בקרת תצורה

- מבחינת תהליך זה אלו נקראים CI – Configuration Items
- לכל אחד יכולים להיות גרסאות ועותקים שונים
- אנו צריכים יכולת לזהות, לעקוב ולאחסן אותם
 - נתמקד בנושא של גרסאות

Version Control – בקרת גרסאות

- איך (האם?) אתם שומרים את תוצרי העבודהשלכם?
 - ? האם אפשר לשפר
- האם יש הבדל בין מפתח בודד לחברה גדולה?
 - שמות שונים:
 - בקרת תצורה
 - Revision Control -
 - Software Configuration Management –
 - Source-Code/Version Control System –

Joel Test (~2000 / stackoverflow)

1. Do you use source control?

וגם היום...

"You've just spent twenty minutes doing a presentation for your teammates on adopting source control. Yeah, they don't do source control at all. Yep, not at all—it's as if the last 20 years of computer science never happened. But better late than never, and frankly any source control is better than none, because disaster is one errant delete away."

- "Driving Technical Change: Why People on Your Team Don't Act on Good Ideas, and How to Convince Them They Should", chap. The Cynic

בקרת גרסאות – בשביל מה? יעדים



- איסוף כל הגרסאות ומעקב אחרי שינויים
 - חזרה לגרסה מסוימת, השוואה
 - מאפשר מחיקת קוד
 - ניהול מספר גרסאות במקביל
 - גיבוי והצלה
- שיתוף מספר מפתחים (מרוחקים) בו זמנית
 - טיפול בסתירות
 - מאגר מעודכן של תוצרי הפרויקט
 - ר∖daily build במיוחד עם –

בפרויקט תדרשו להדגים את בקרת התצורה שלכם

פעולות נדרשות

- בקרת שינויים
- זיהוי ותיעוד (למשל מי משנה, הסיבה, זמן וכדו')
 - ניתוח והערכה (של שינוי)
 - אישור ∖ דחיה
 - אימות, מימש ושחרור
 - בקרת גרסאות
 - מאגר
 - הכנסה והוצאה
 - ענפים ומיזוגים
 - תיוג –



<u>כלים: היסטוריה</u> (<u>השוואה</u>)

Generation	Networking	Operations	Concurrency	Examples
1	None	One file at a time	Locks	RCS, SCCS
2	Centralized	Multi-file	Merge before commit	CVS, SourceSafe, Subversion, Team Foundation Server, IBM Rational ClearCase
3	Distributed	Changesets	Commit before merge	Bazaar, <mark>Git</mark> , Mercurial

40 Years of Version Control







Image © TheSun.au

Git

Successful open source project

https://git.wiki.kernel.org/index.php/GitProjects
https://github.com/google, microsoft, facebook, twitter...
http://stackoverflow.com/research/developer-survey-2015#tech-sourcecontrol

- Problems / Issues:
 - Usability!
 - Mainly a scripted / toolset (by now IDE integration and GUIs)
 - Binary/big file
 - Enterprise (e.g. locking)

משל גיט

- Tom Preston-Werner

 http://tom.preston-werner.com/2009/05/19/the-git-parable.html
- Herland,
 http://www.infoq.com/presentations/git-details, slides:
 https://github.com/jherland/git_parable

Johan Herland

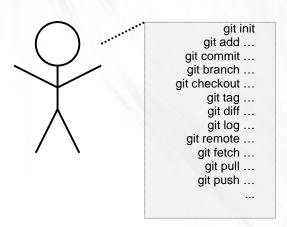
johan@herland.net

- Shamelessly stolen from Tom Preston-Werner
 http://tom.preston-werner.com/2009/05/19/the-git-parable.html
- I'm lazy...
- Also: Best introduction to Git I've found so far

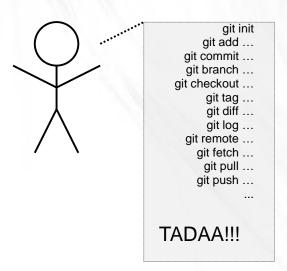
• Git - simple & powerful

20

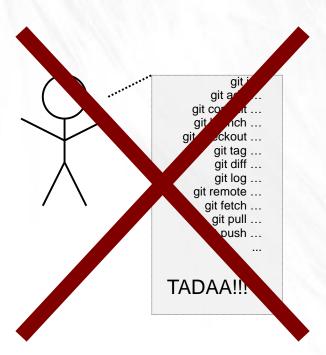
• Git - simple & powerful



• Git - simple & powerful

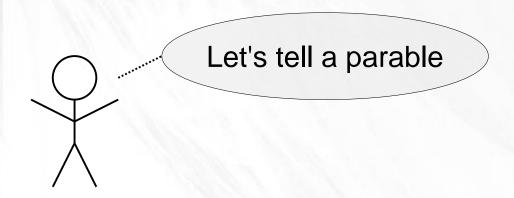


• Git - simple & powerful



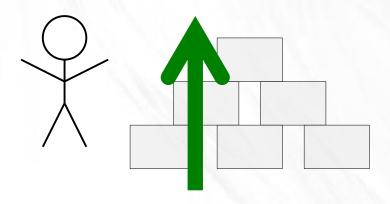
se16b-yagel

• Git - simple & powerful

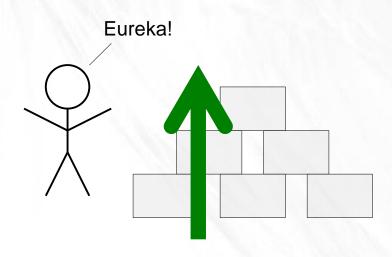


24

• Git - simple & powerful



• Git - simple & powerful



The Parable

- A simple computer
 - A text editor
 - A few filesystem commands



The Parable

Write a large software program

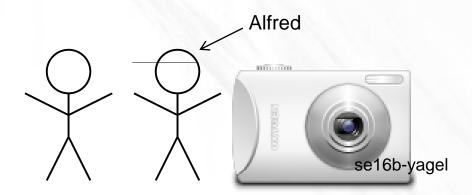


The Parable

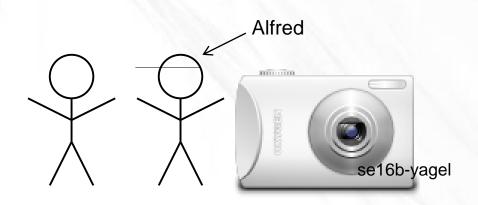
- Write a large software program
- Invent some method to keep track of versions
 - retrieve code that you changed/deleted



Alfred, the photographer



Alfred, the photographer

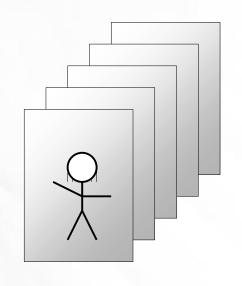




- Alfred, the photographer
- Hazel and her daughter

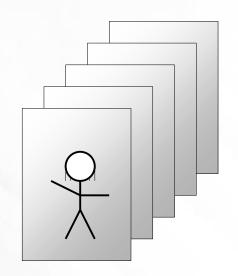


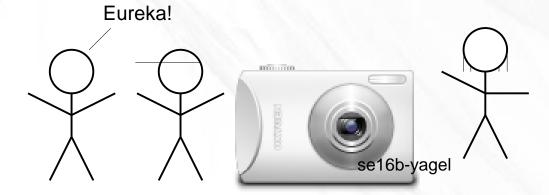
- Alfred, the photographer
- Hazel and her daughter
 - Remember what the daughter was like at each different stage



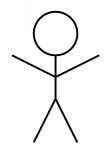


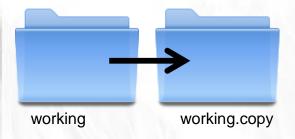
- Alfred, the photographer
- Hazel and her daughter
 - Remember what the daughter was like at each different stage

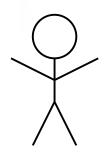




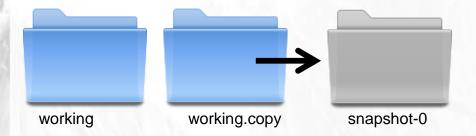


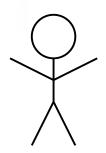




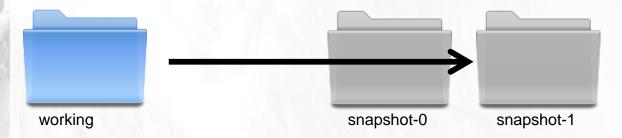


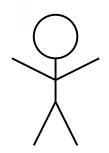
Snapshots





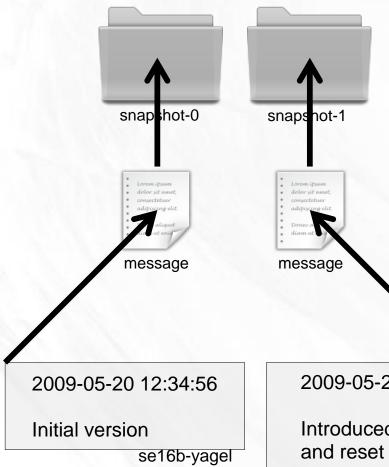
Snapshots

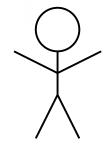




Snapshots







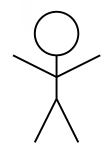
2009-05-21 23:45:01

Introduced a new foo, and reset the bar to xyzzy.

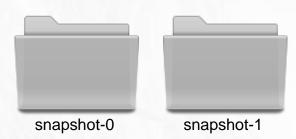




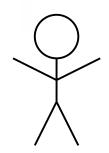


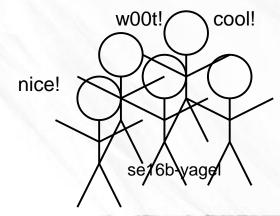




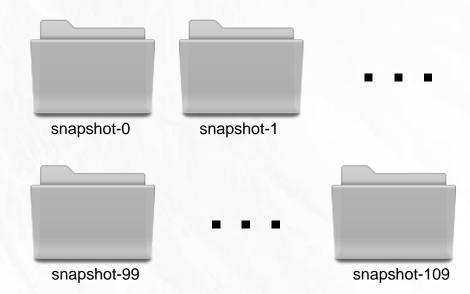


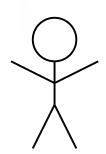






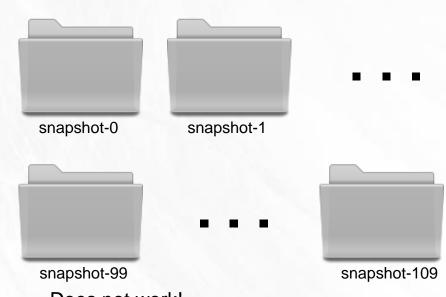


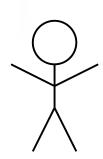


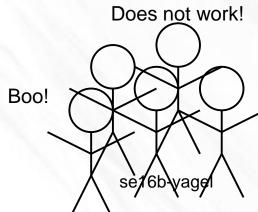


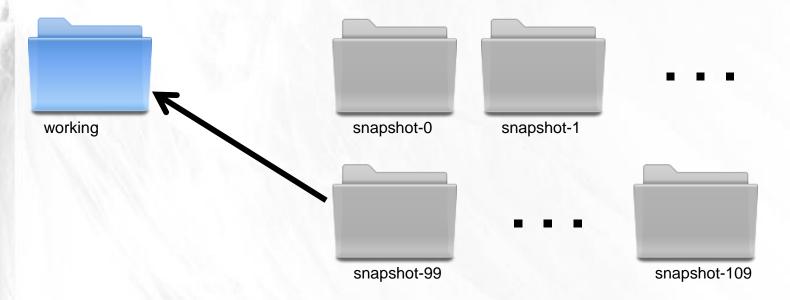


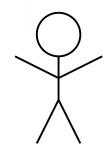
















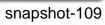


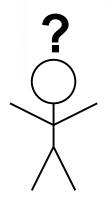




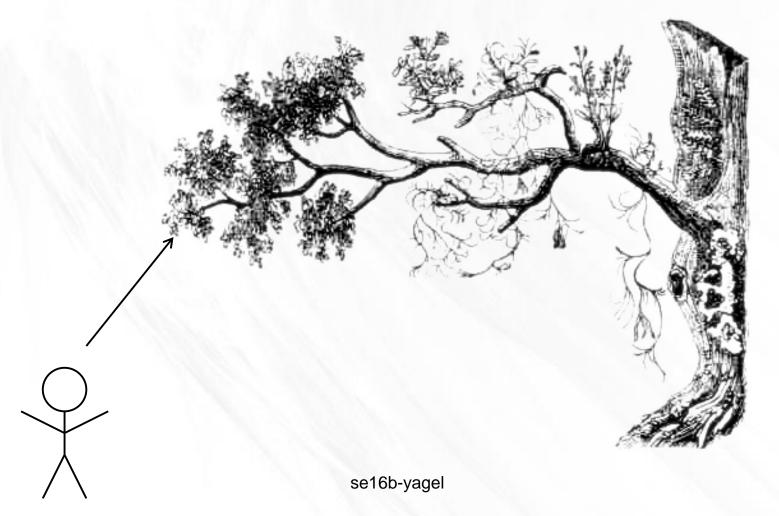


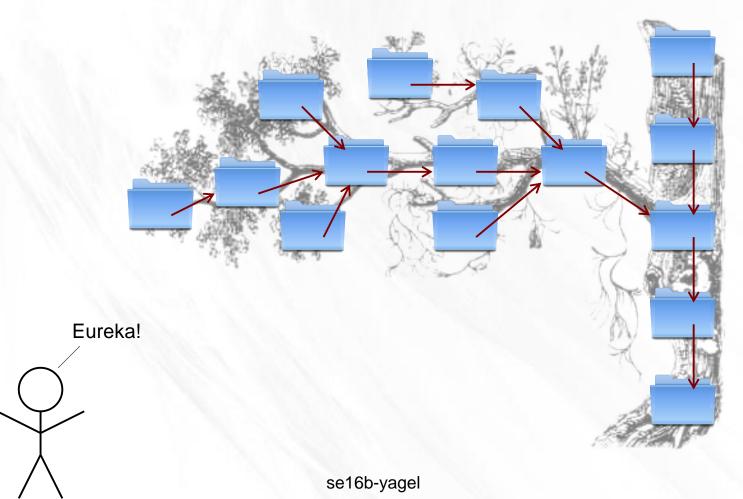


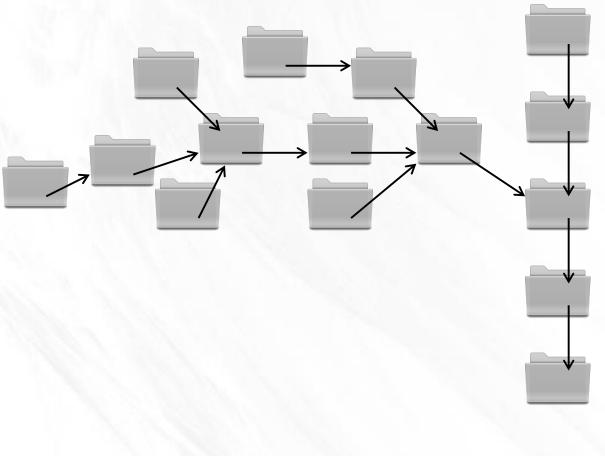


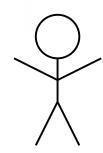


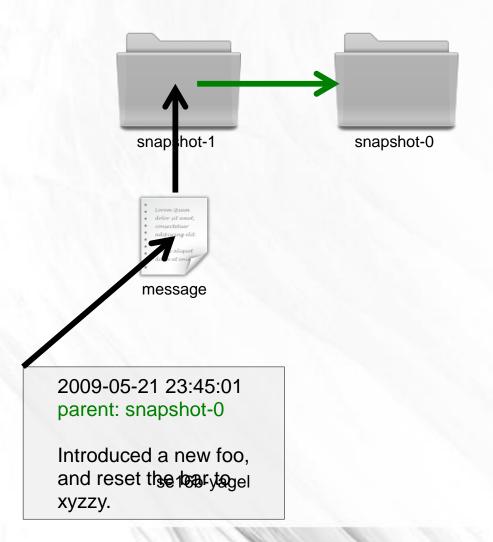
se16b-yagel

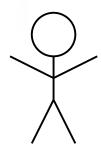








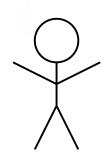


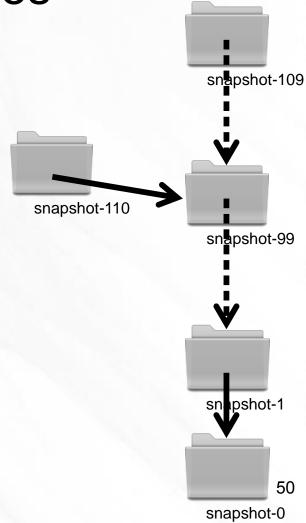


Branch Names

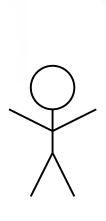


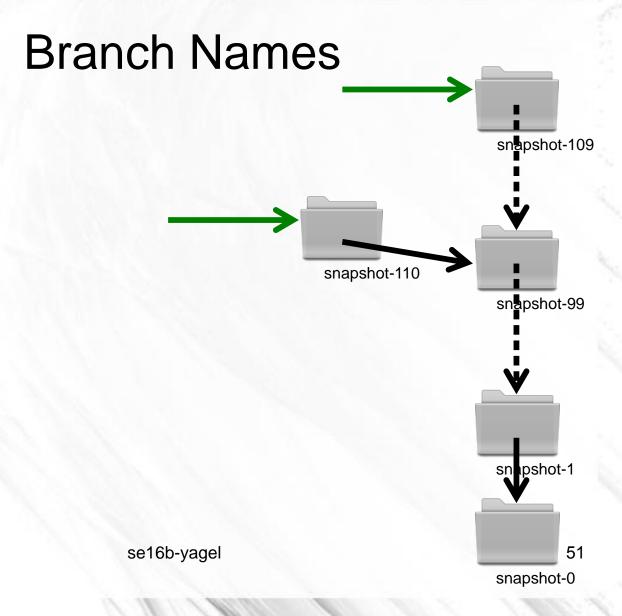




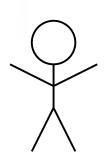


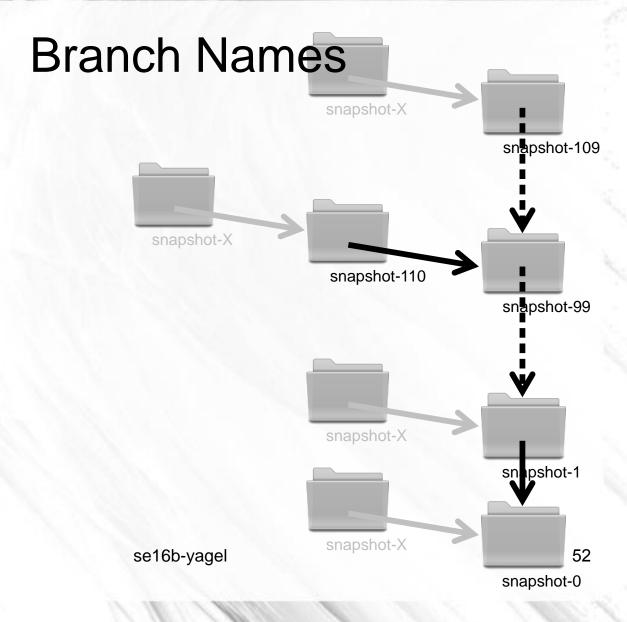


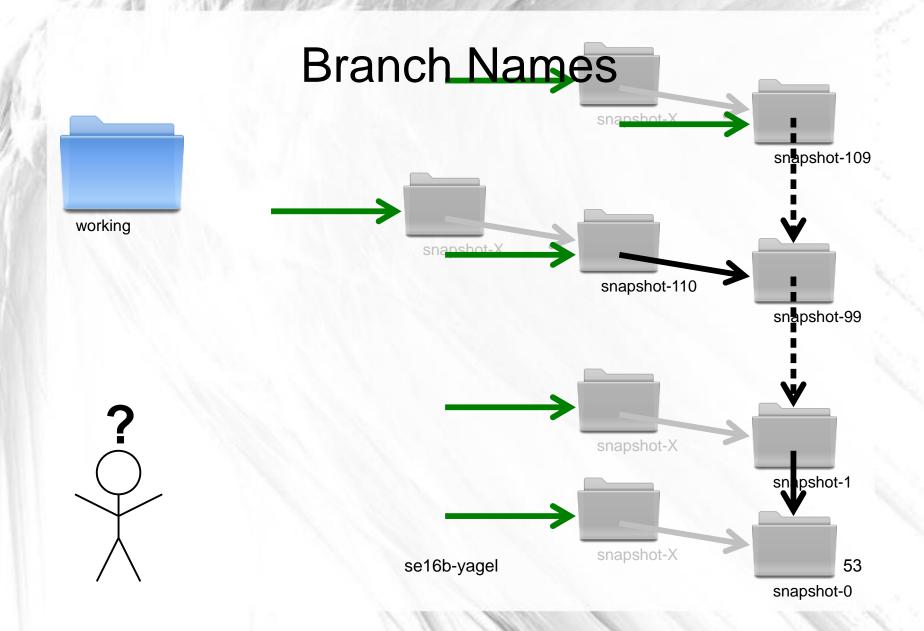


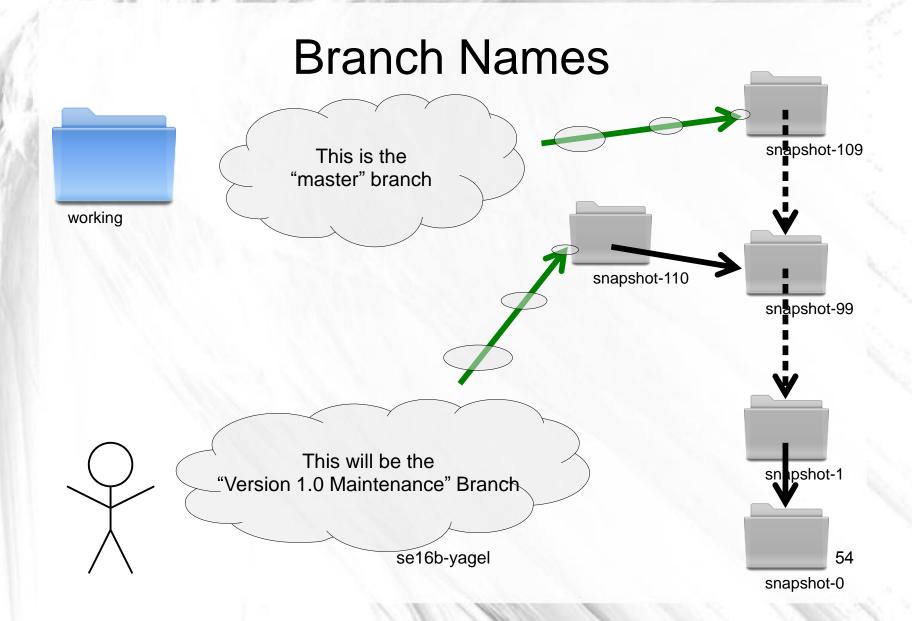


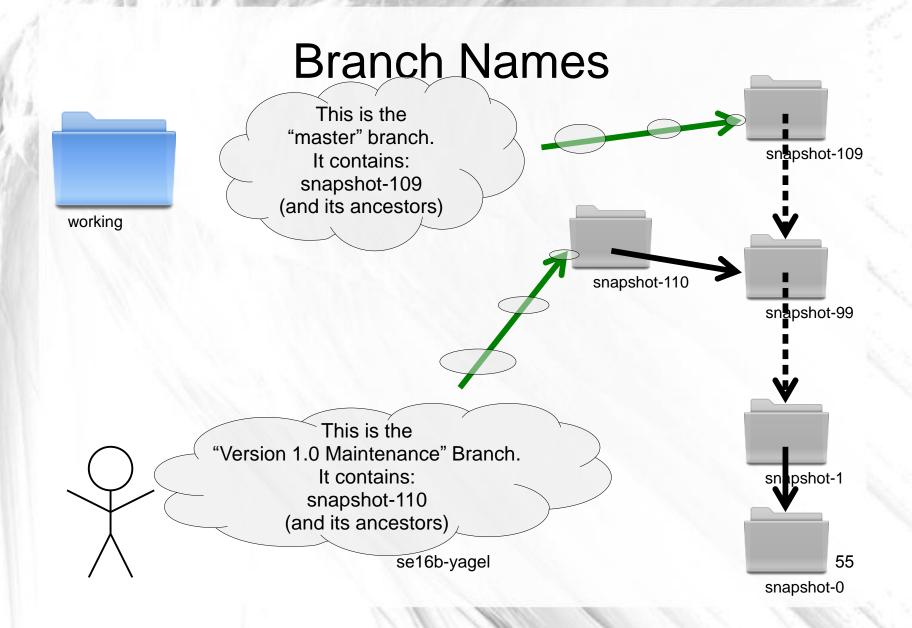


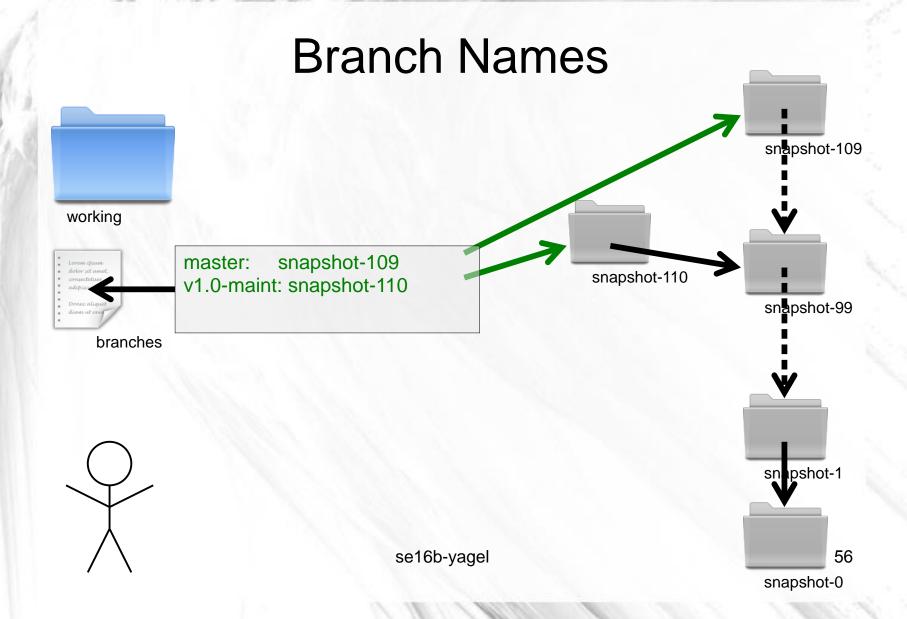


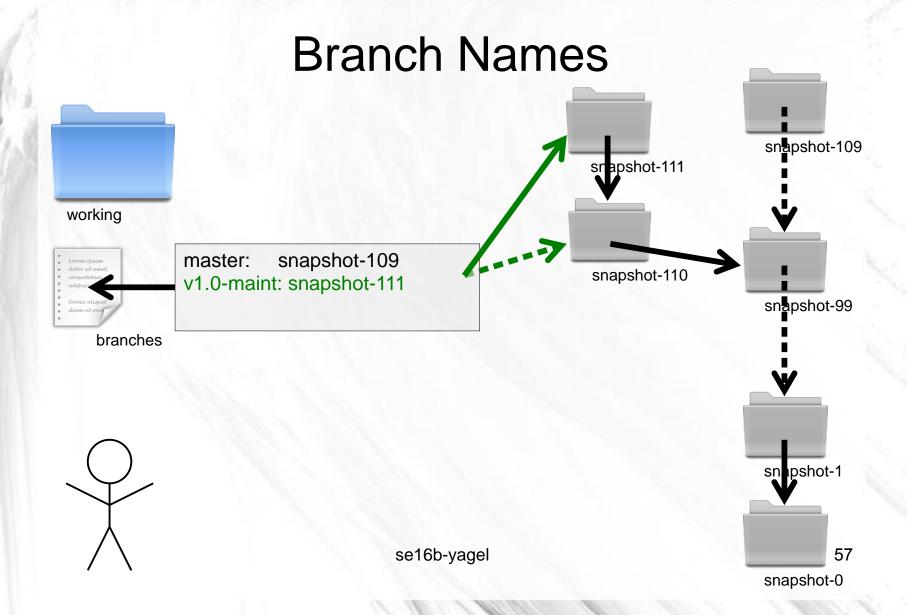


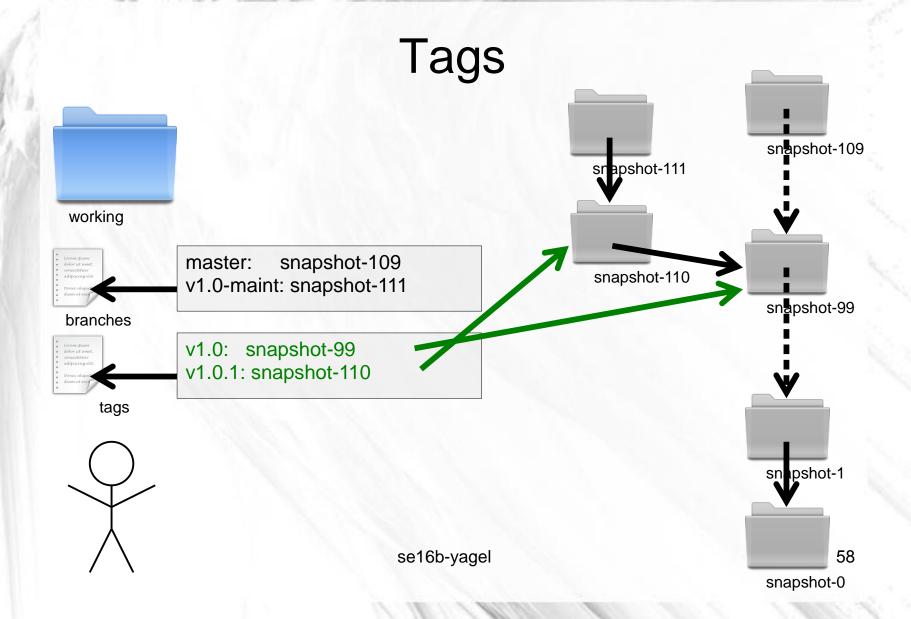


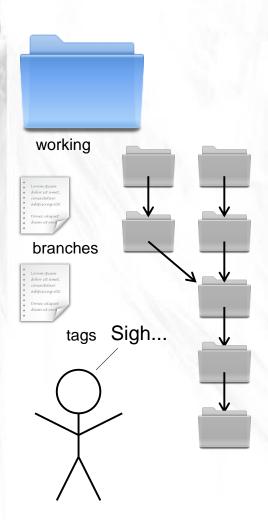


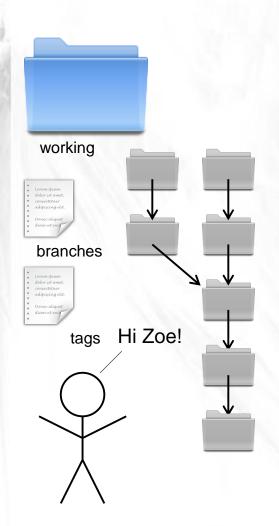


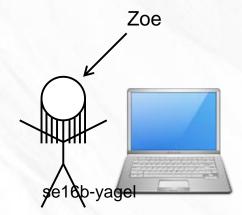


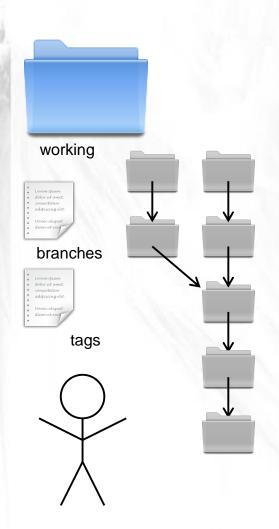


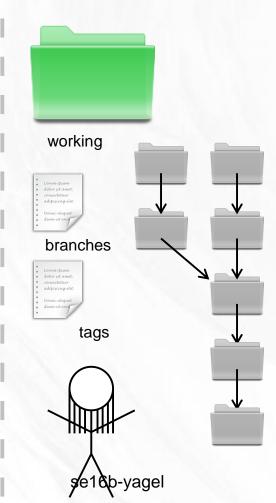


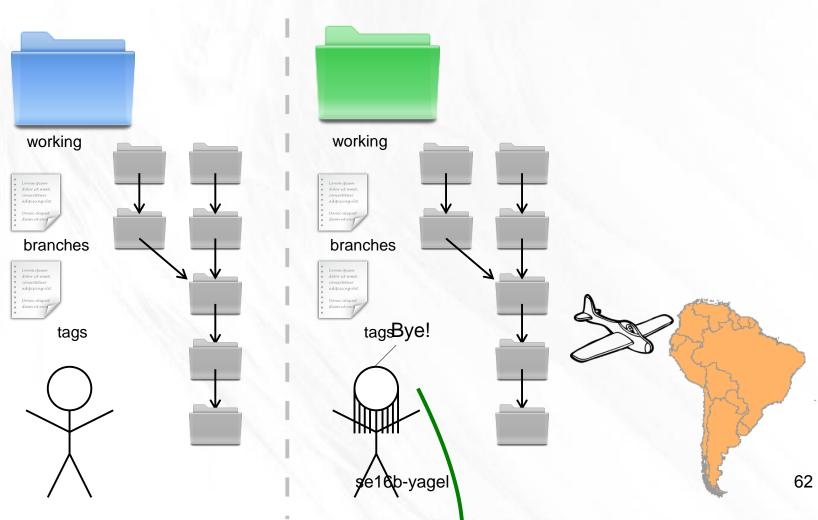


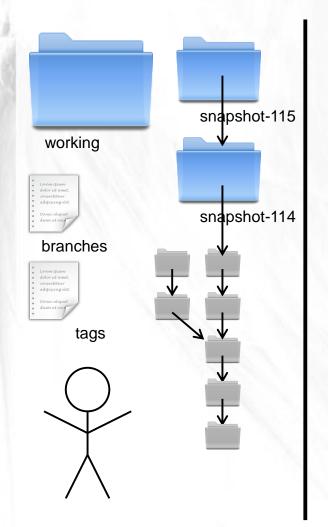


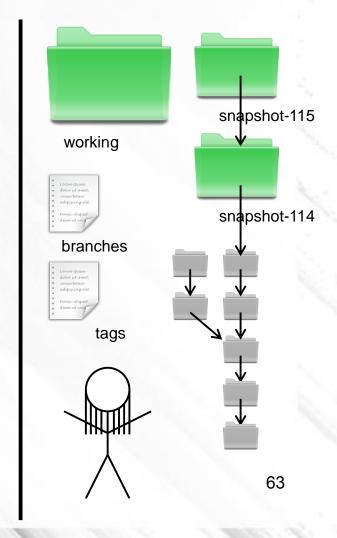


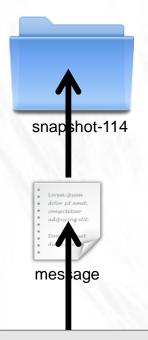








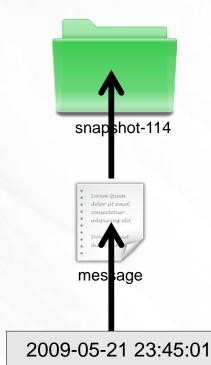




2009-05-22 12:12:12 parent: snapshot-113 author: Me <me@me.me>

Blarfle, a cool new feature; extends the existing blorg.

se16b-yagel



parent: snapshot-113
author: Zoe <zoe@z.oe>

Introduced a new foo, and reset the bar to 64 xyzzy.

8ba3441b6b89cad23387ee875f2ae55069291f4b SHA1

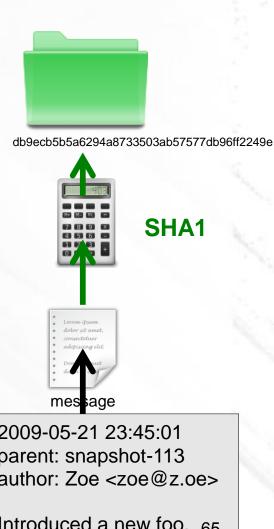
2009-05-22 12:12:12 parent: snapshot-113 author: Me <me@me.me>

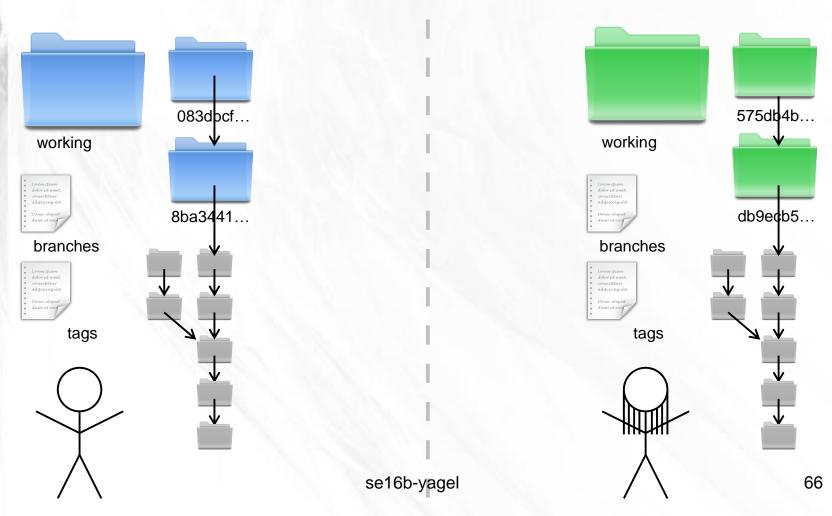
message

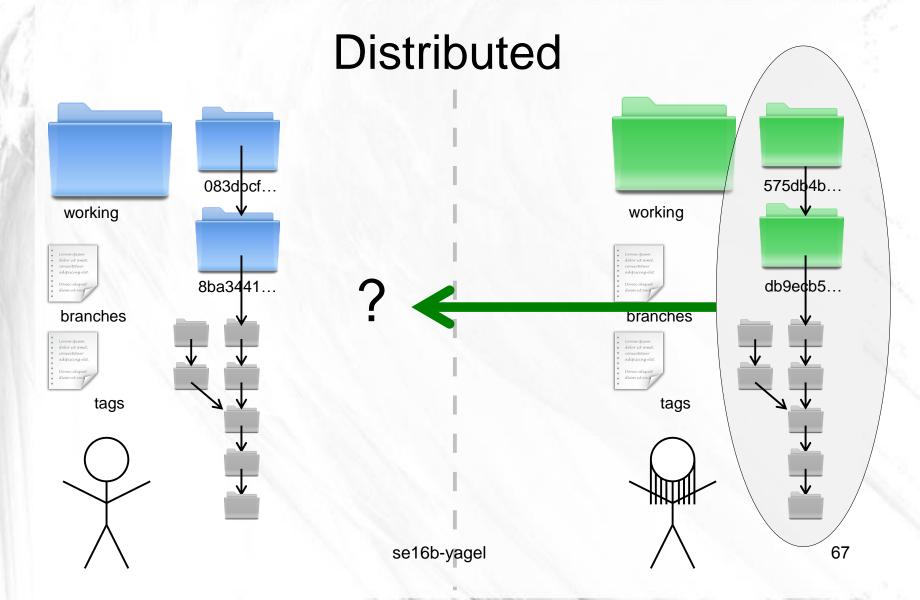
Blarfle, a cool new feature; extends the existing blorg.

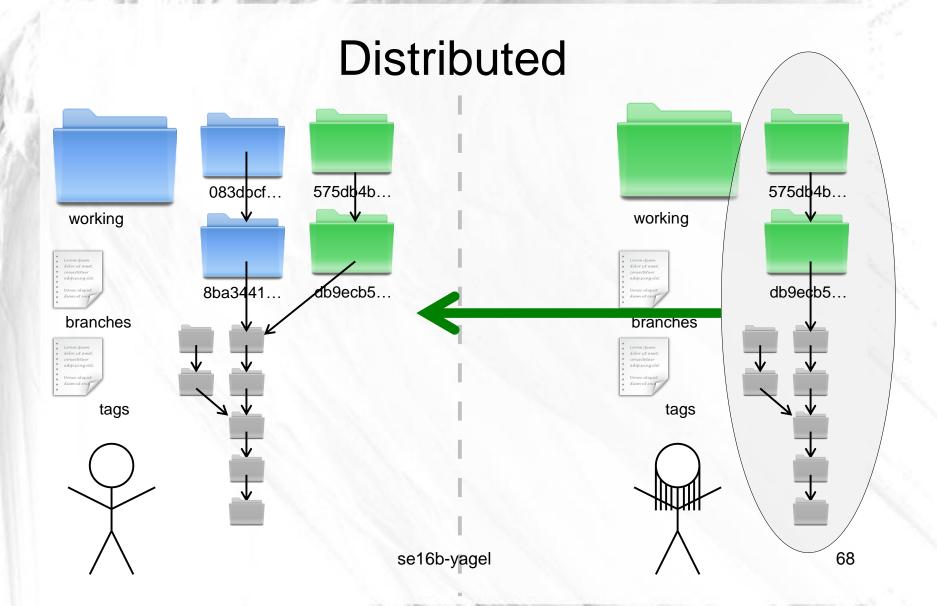
Distributed

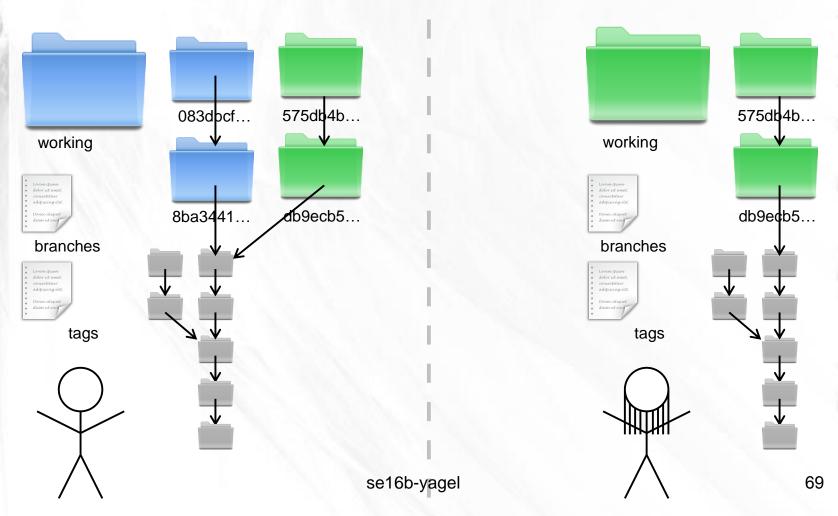
message 2009-05-21 23:45:01 parent: snapshot-113 author: Zoe <zoe@z.oe> Introduced a new foo, 65 and reset the bar to xyzzy.



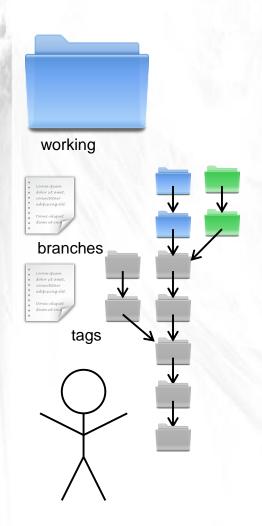


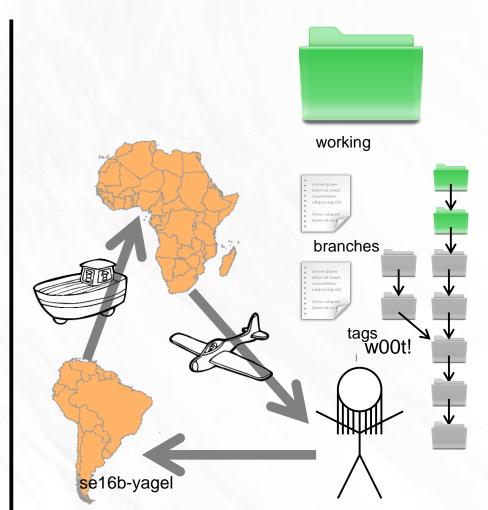




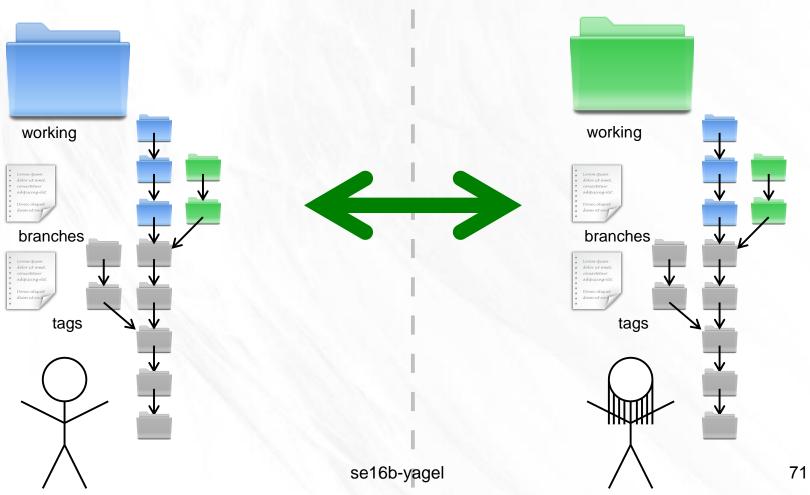


Offline

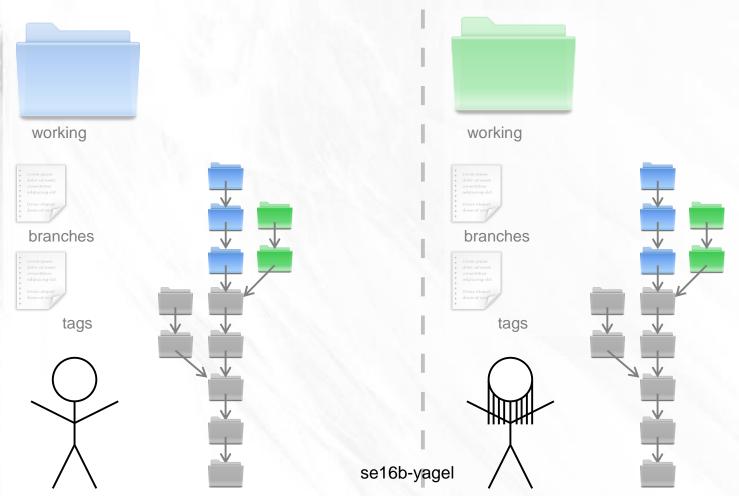




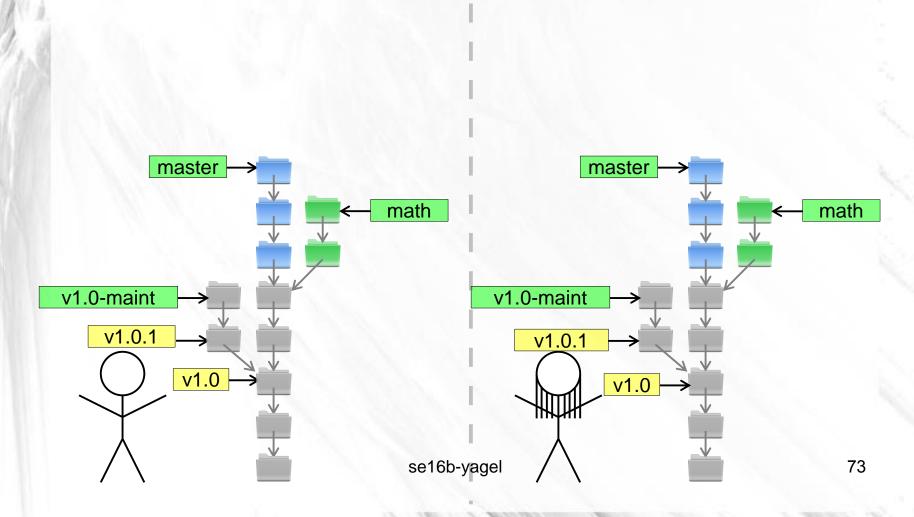
Offline



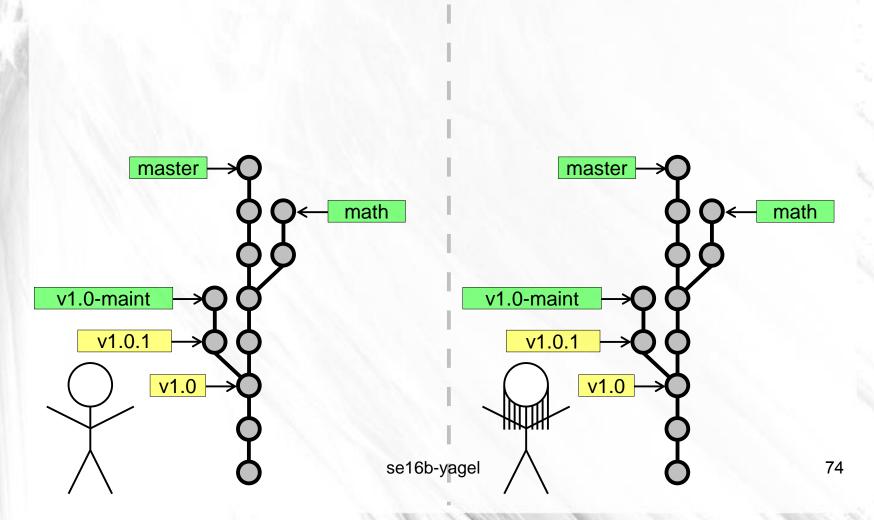
(simpler drawings)



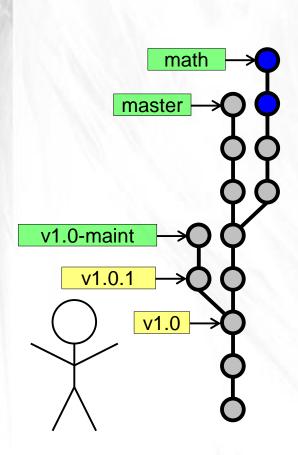
(simpler drawings)

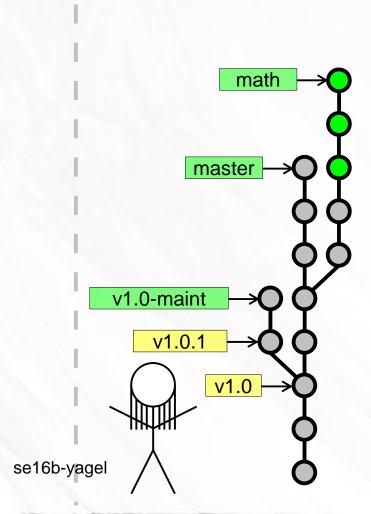


(simpler drawings)



Merges

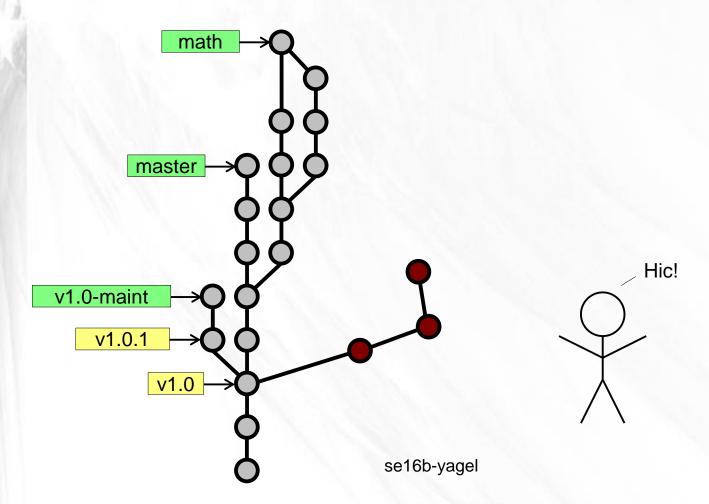


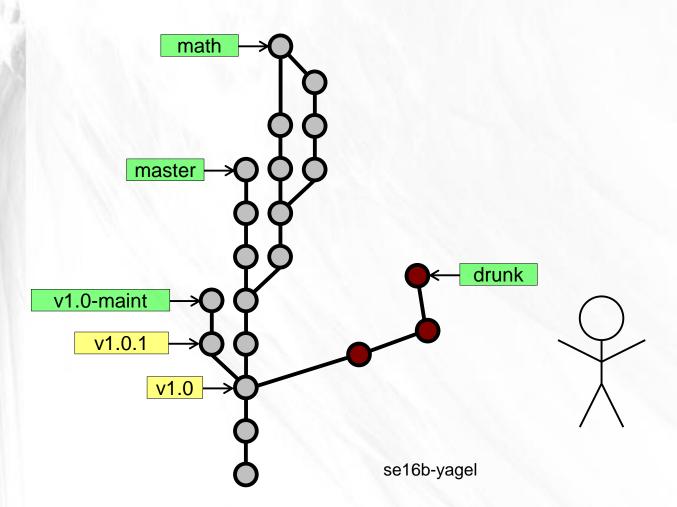


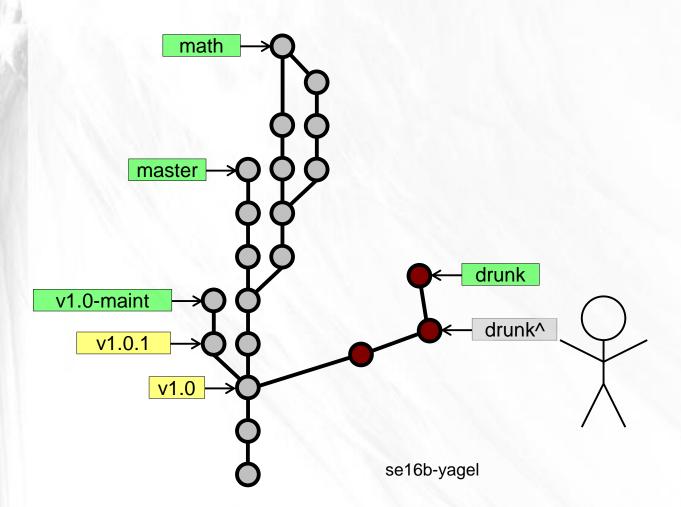
Merges math math master master v1.0-maint v1.0-maint v1.0.1 v1.0.1 v1.0 v1.0 se16b-yagel

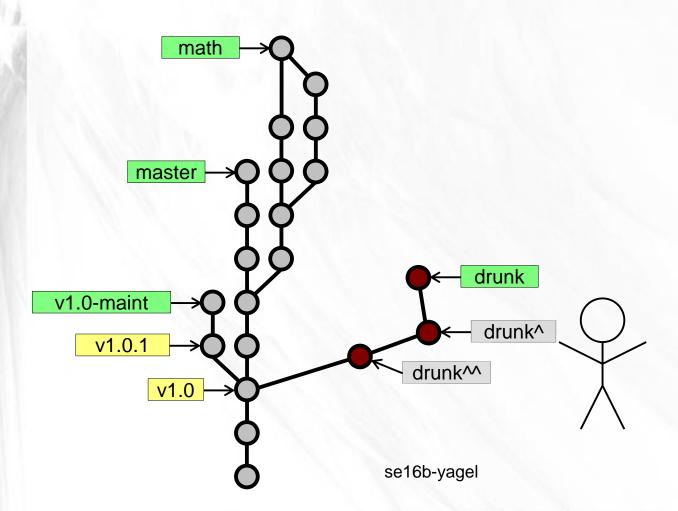
Merges math math master master v1.0-maint v1.0-maint v1.0.1 v1.0.1 v1.0 v1.0 se16b-yagel

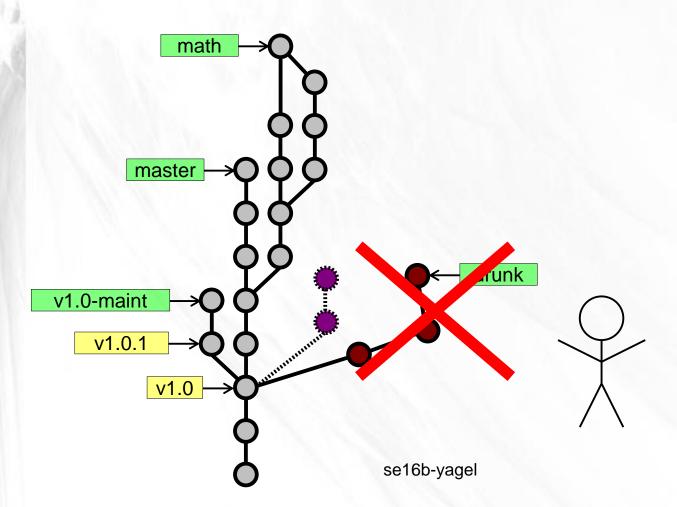
Merges math math master master v1.0-maint v1.0-maint v1.0.1 v1.0.1 v1.0 v1.0 se16b-yagel

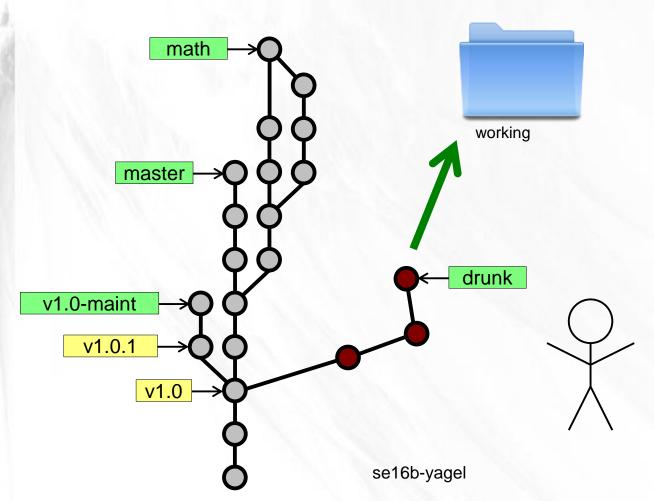


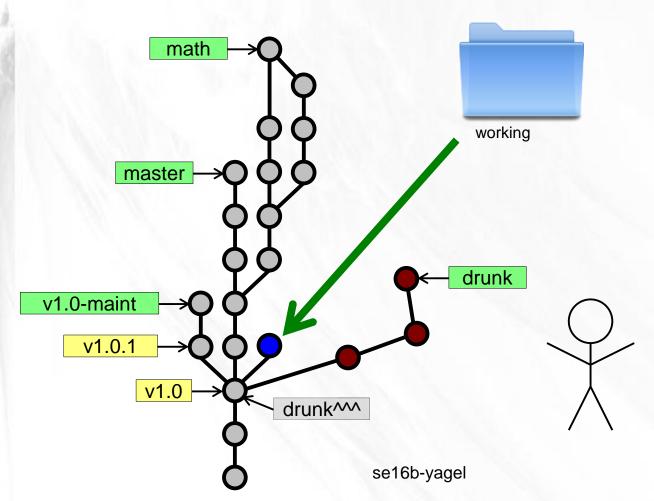


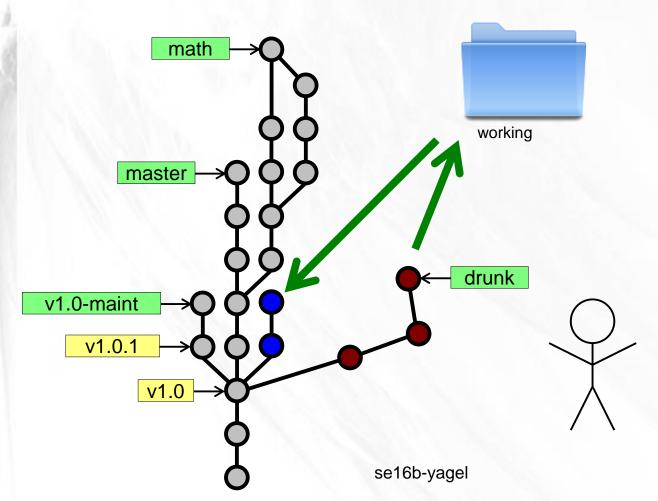


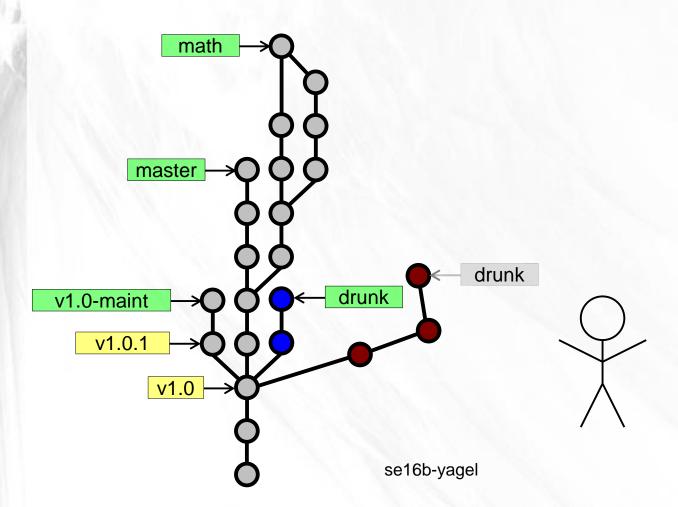


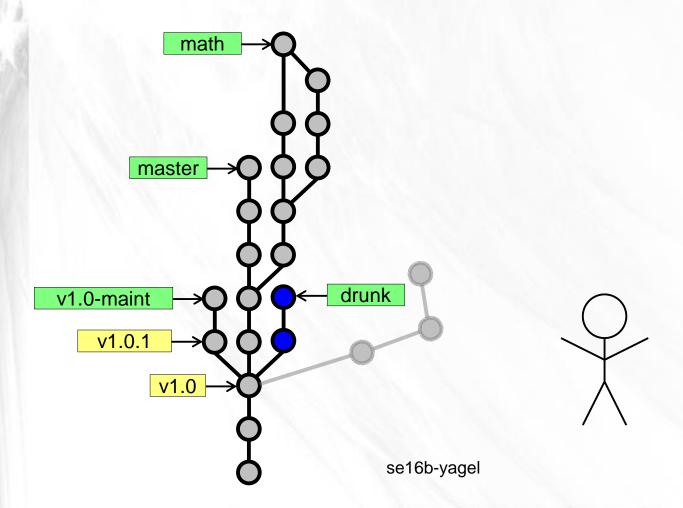


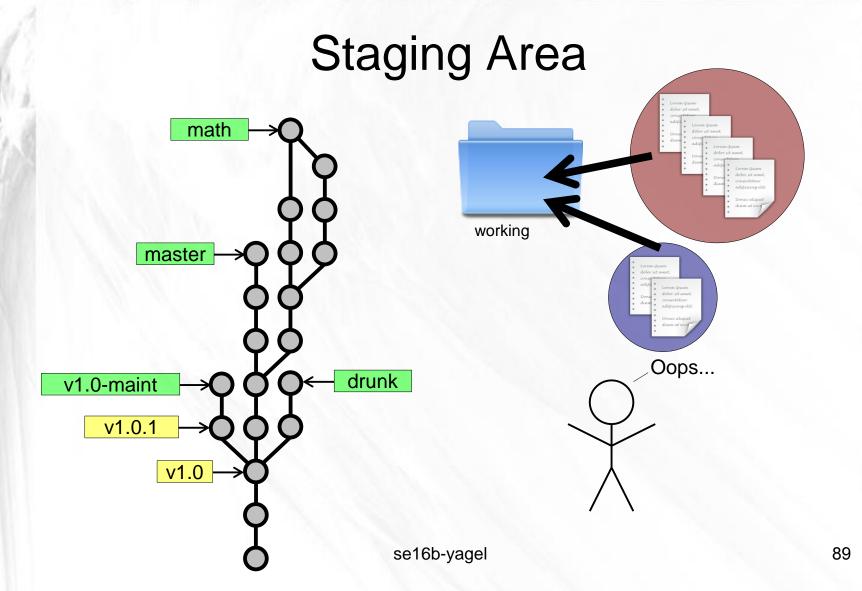




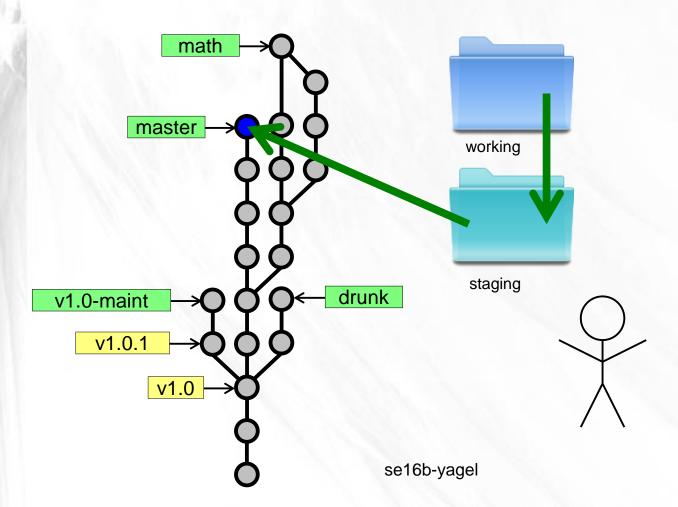


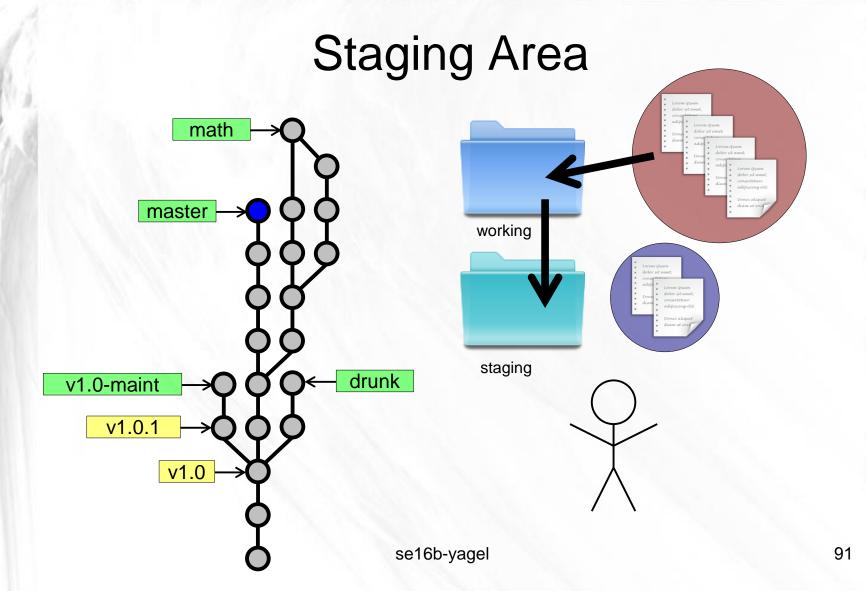


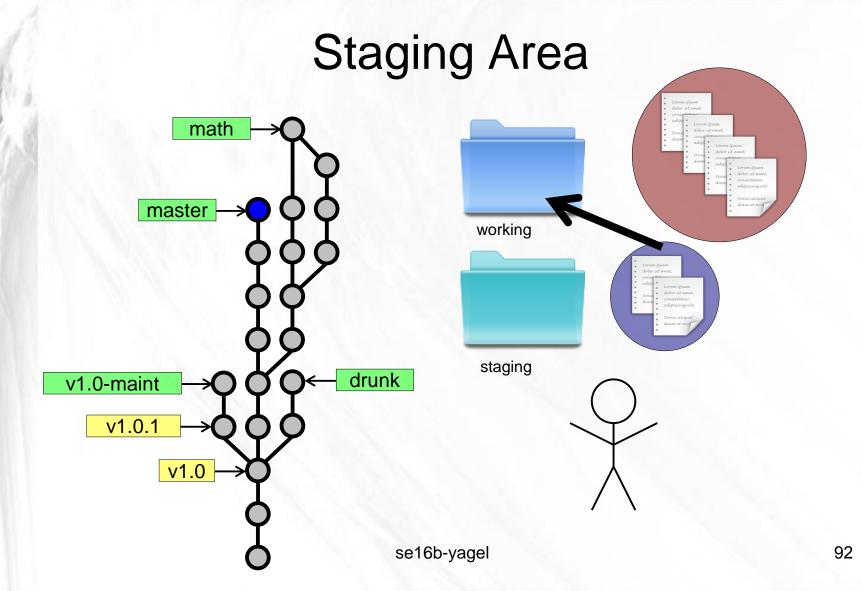




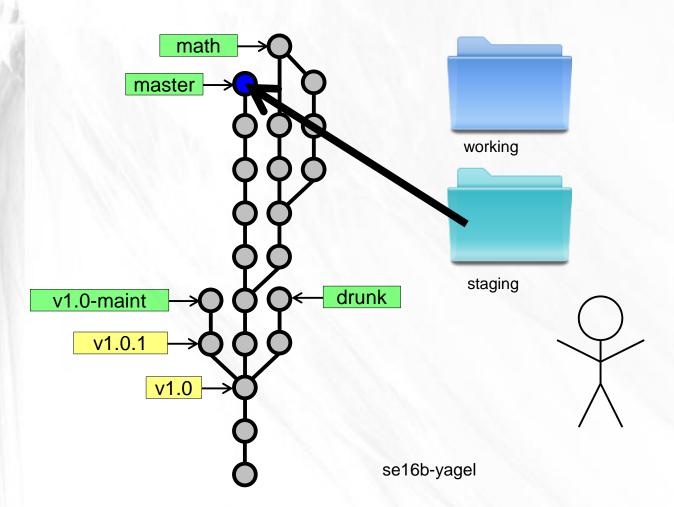
Staging Area



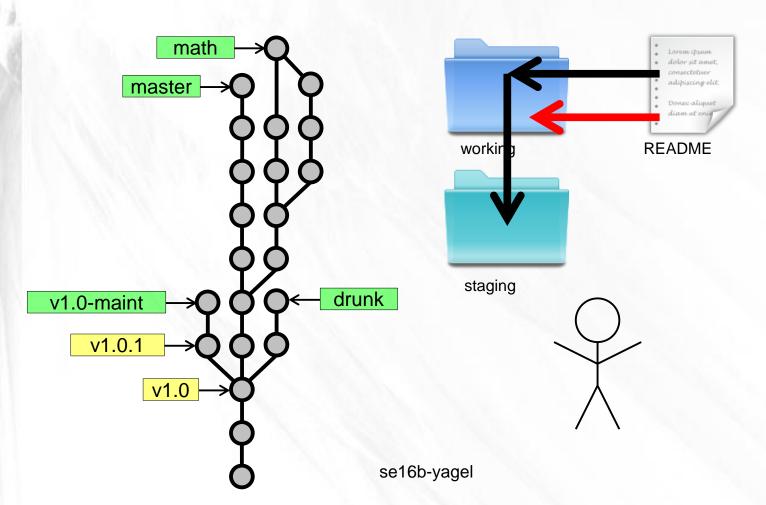




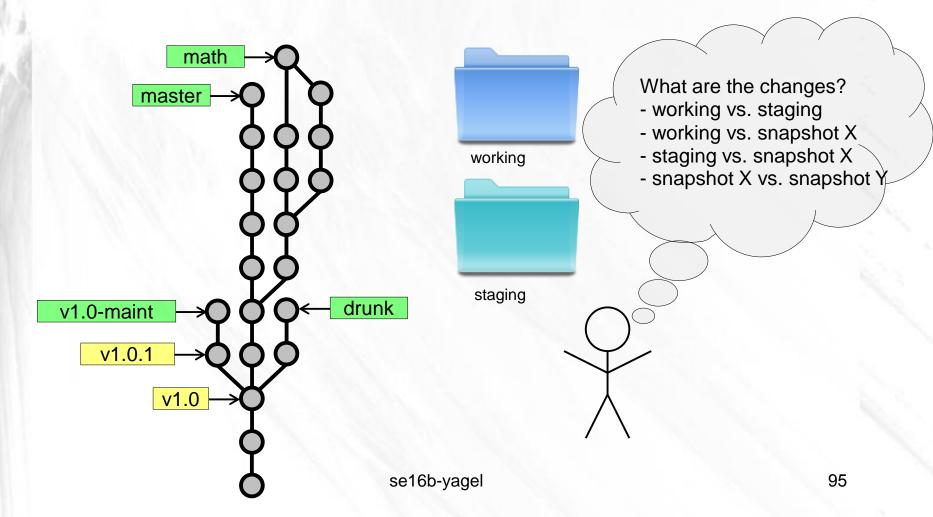
Staging Area

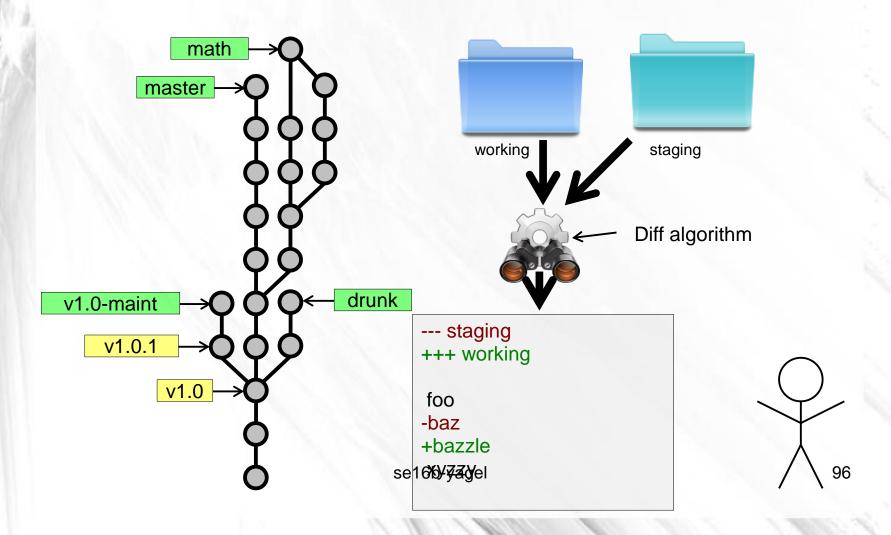


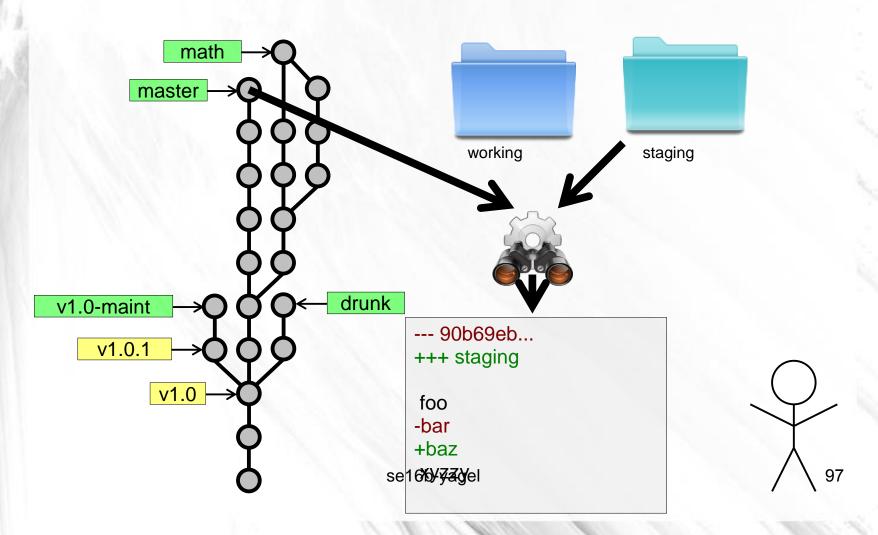
Staging Area

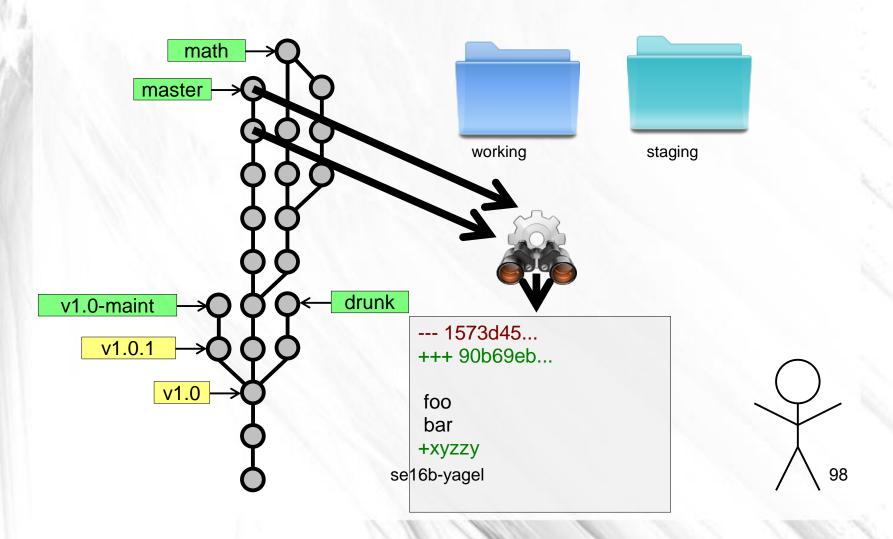


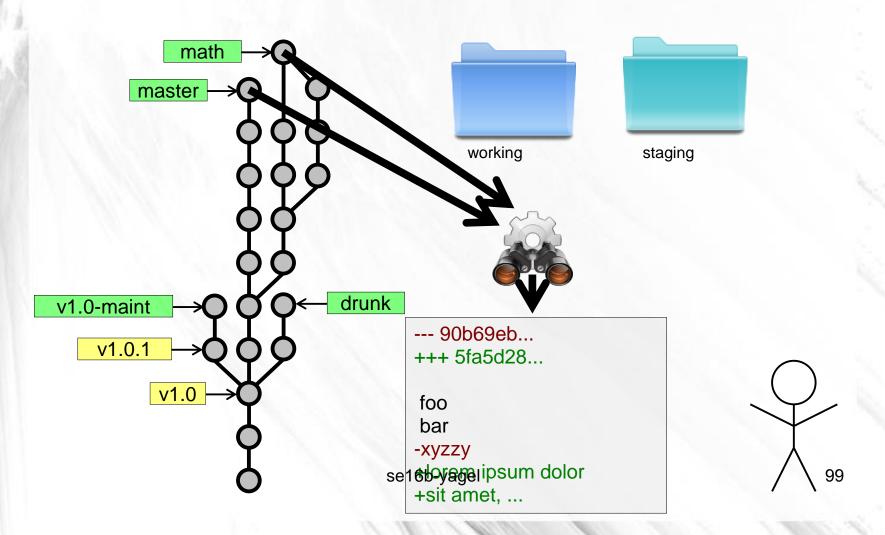
94

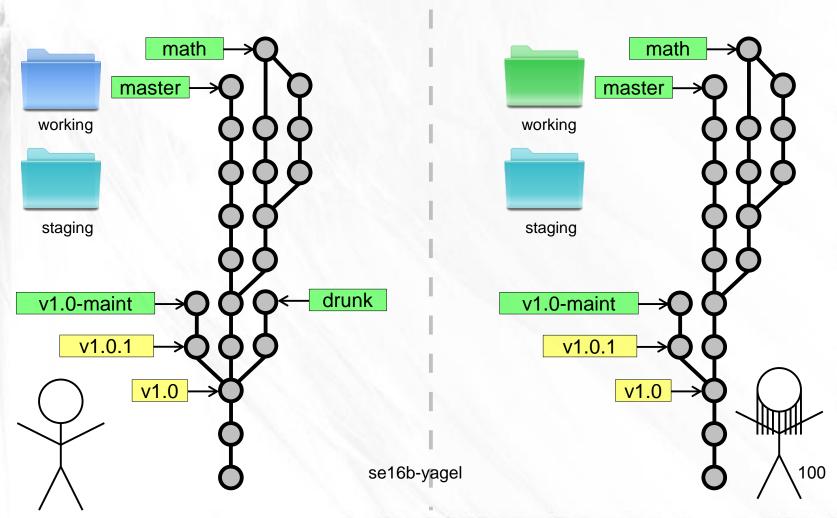


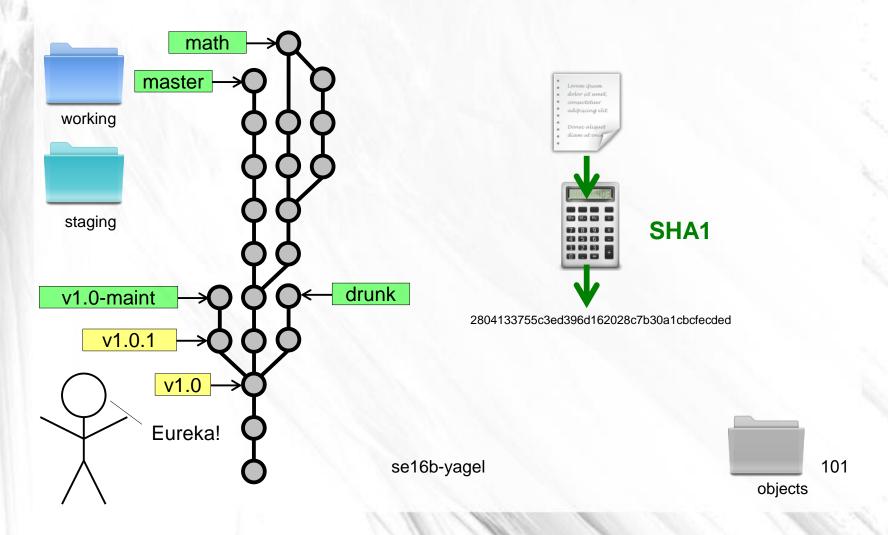


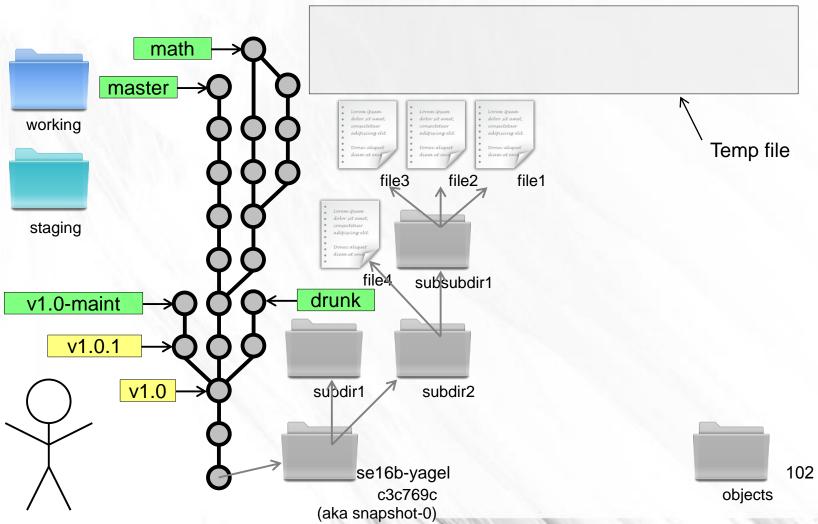


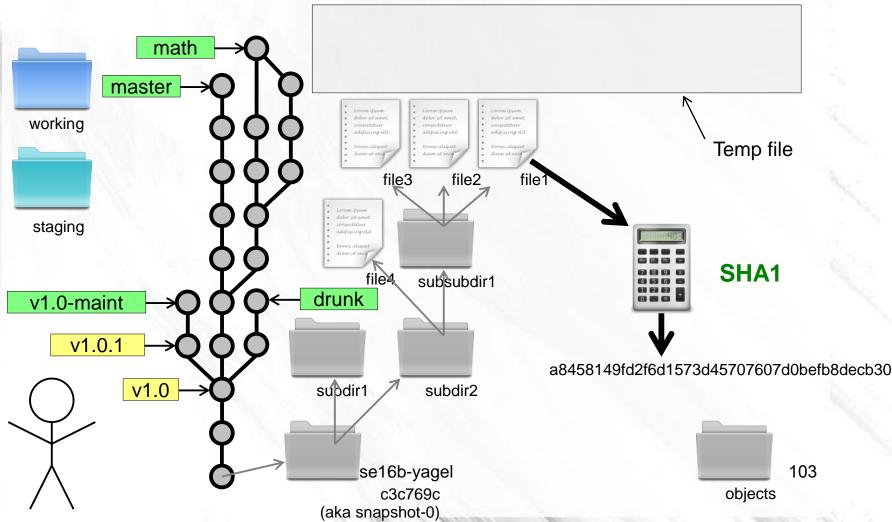


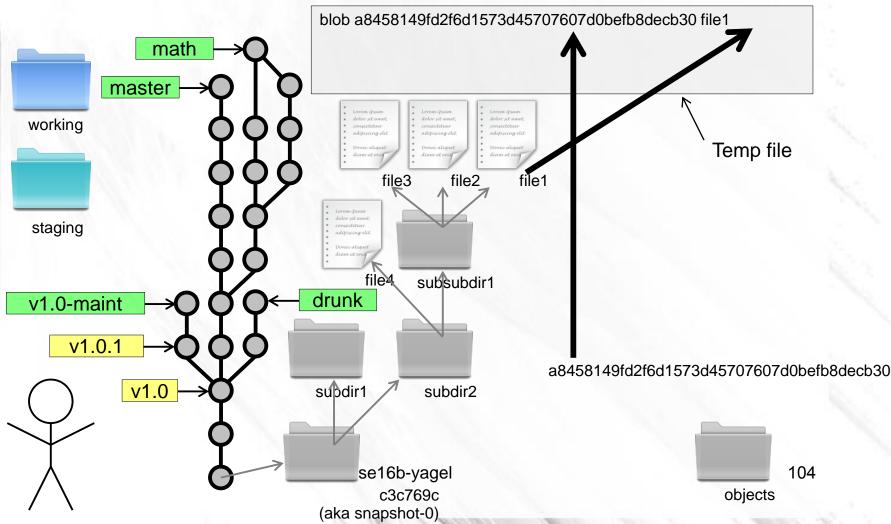


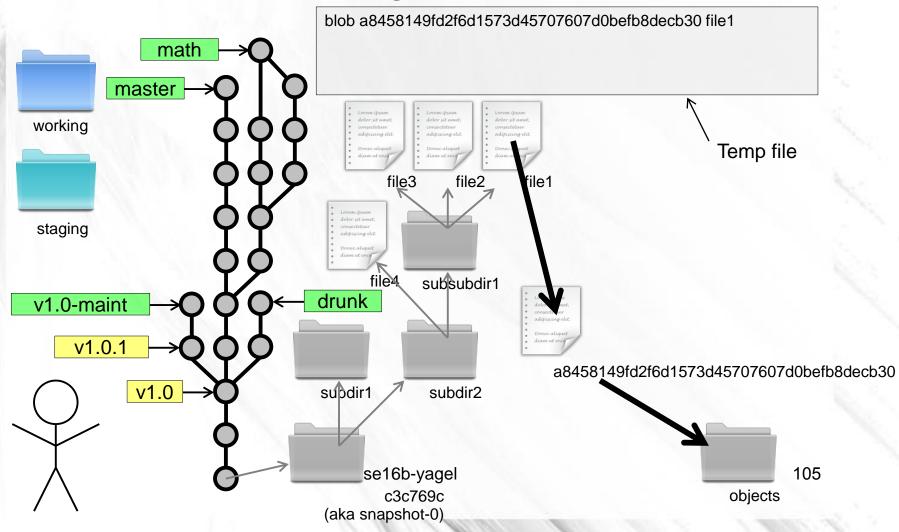


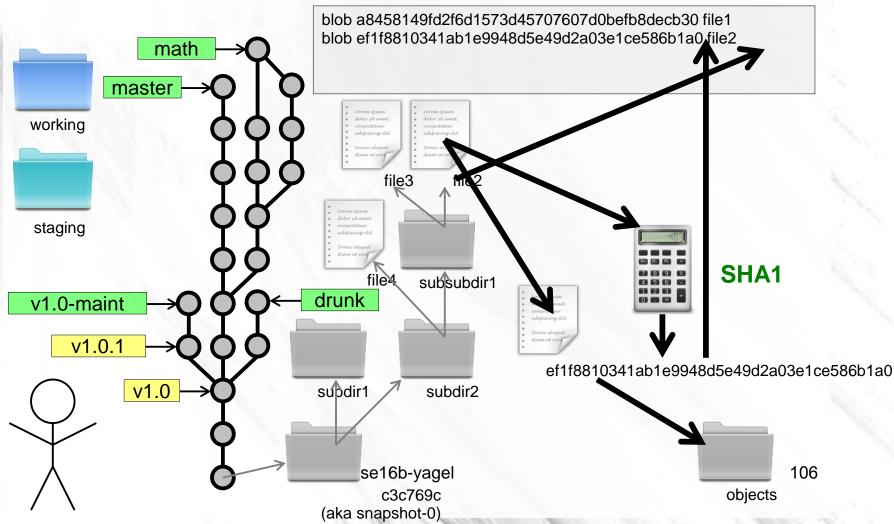


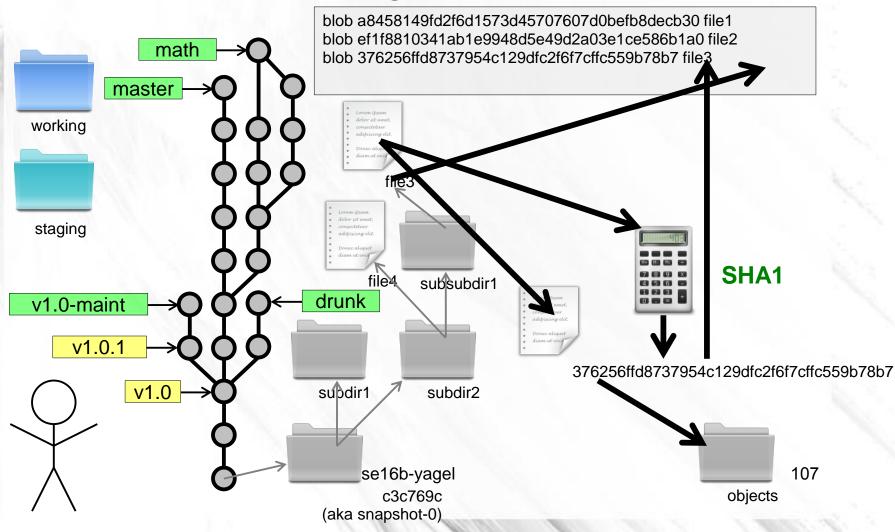


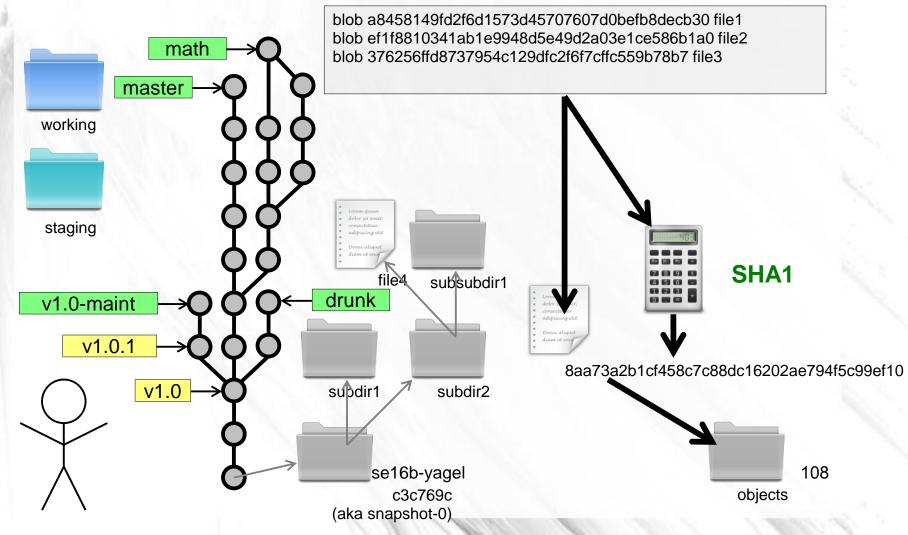


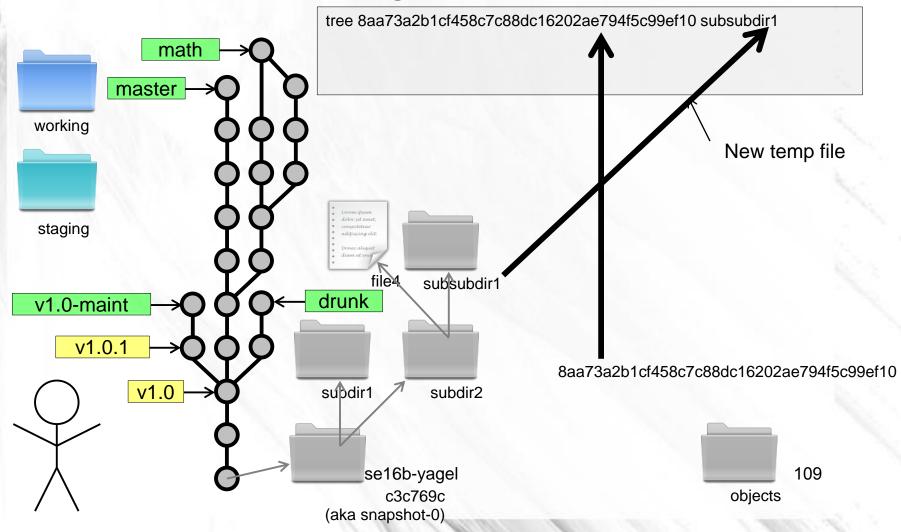


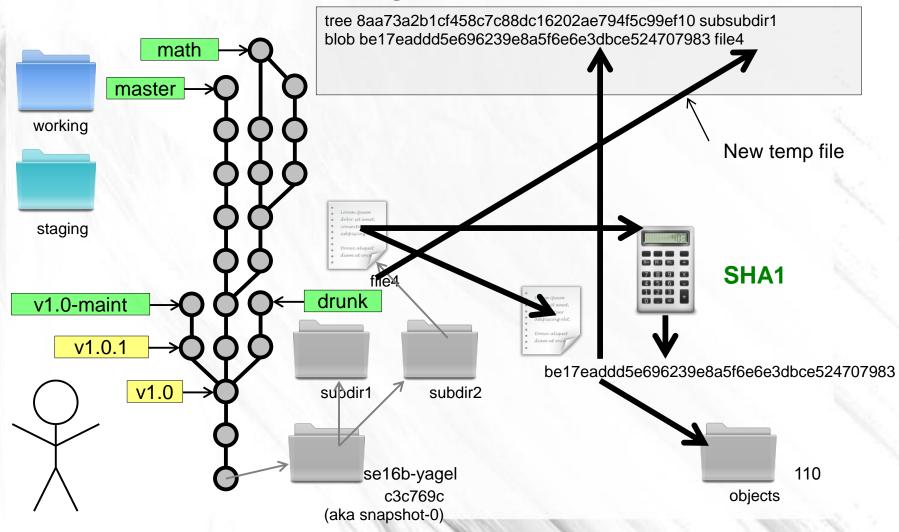


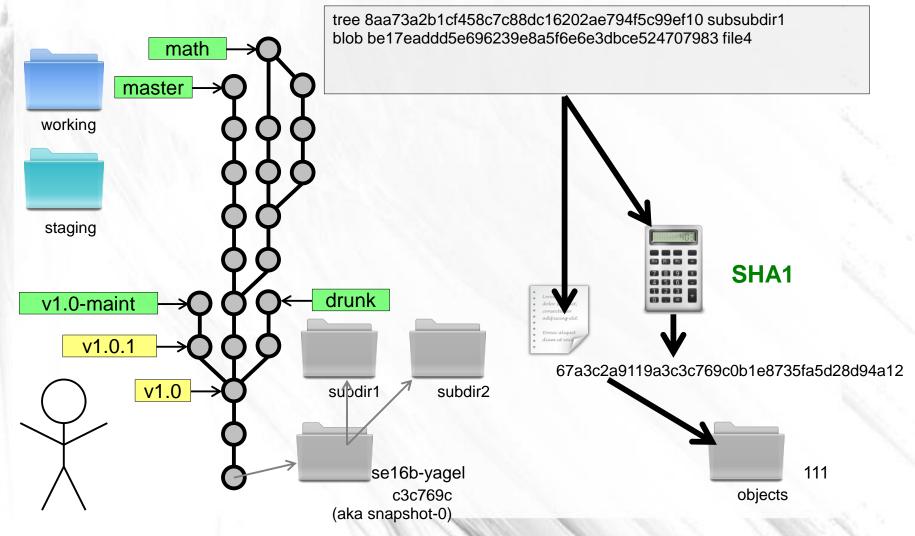


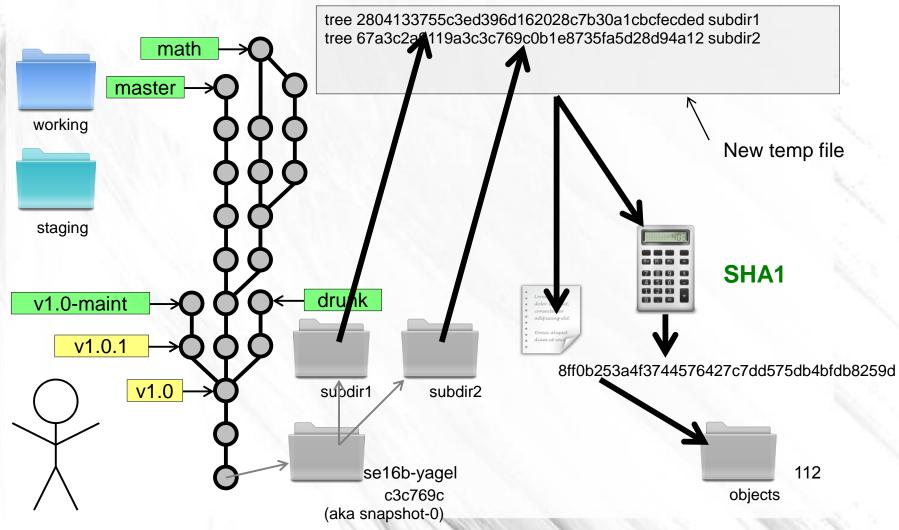


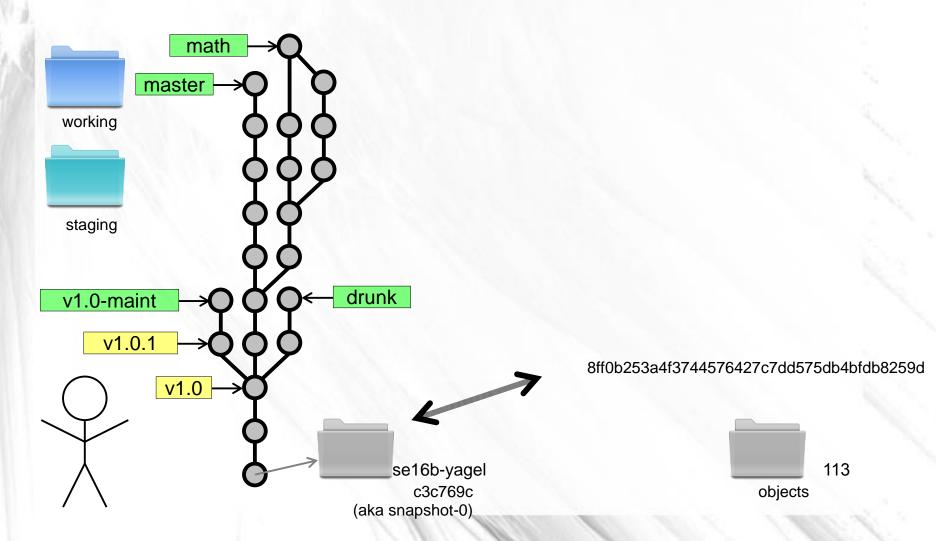


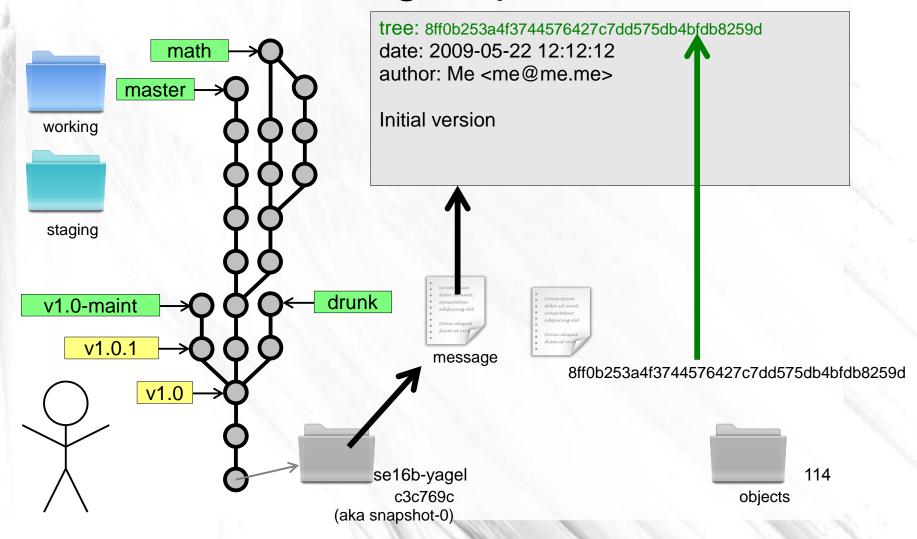


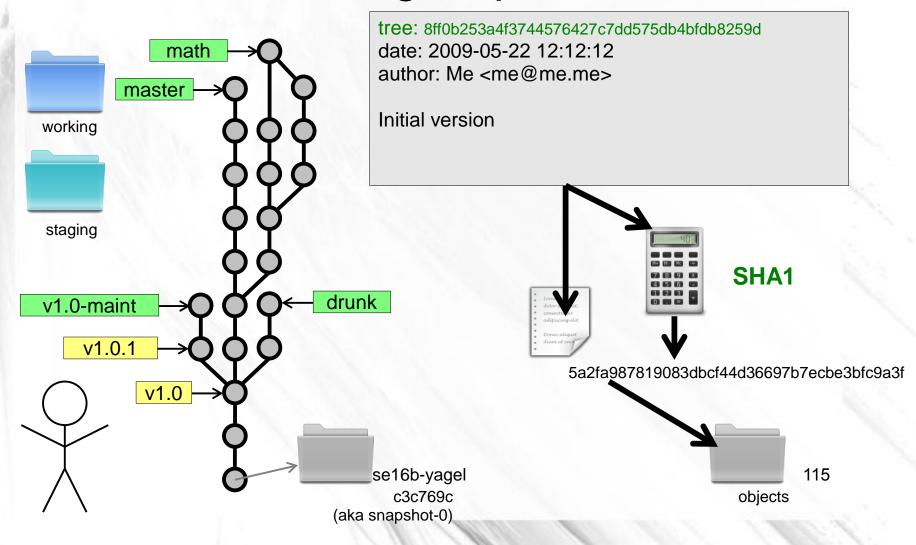


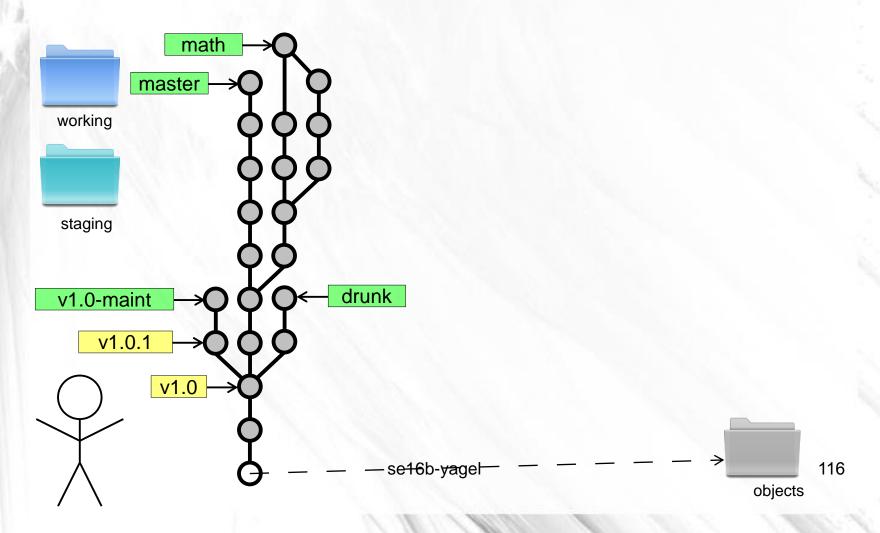


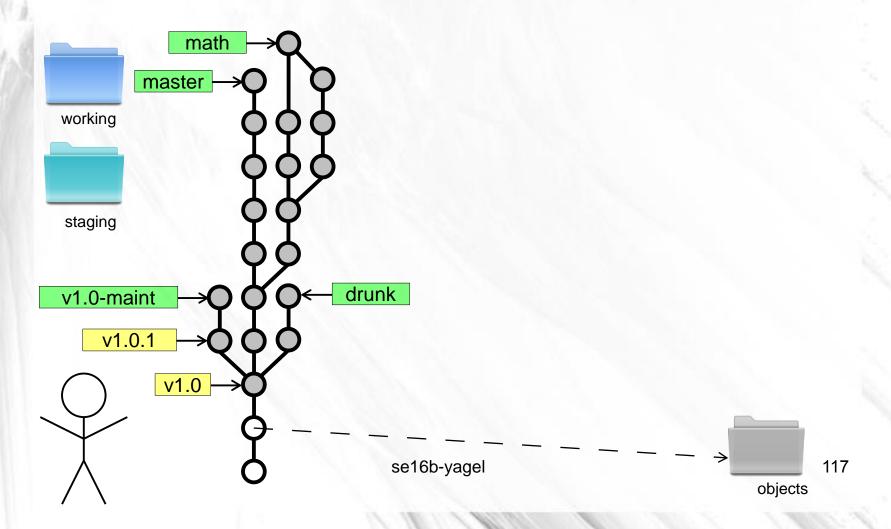


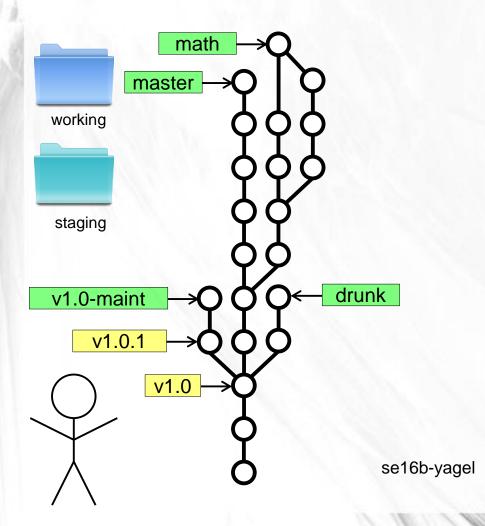






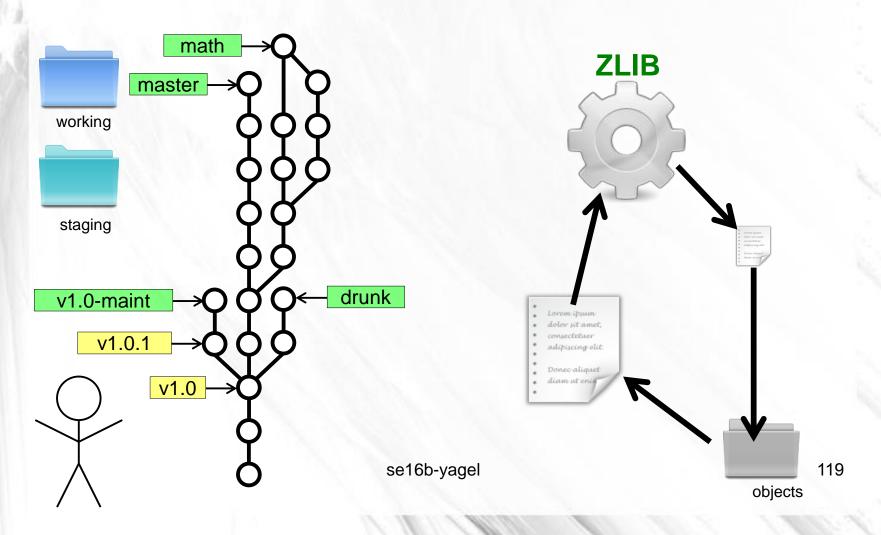




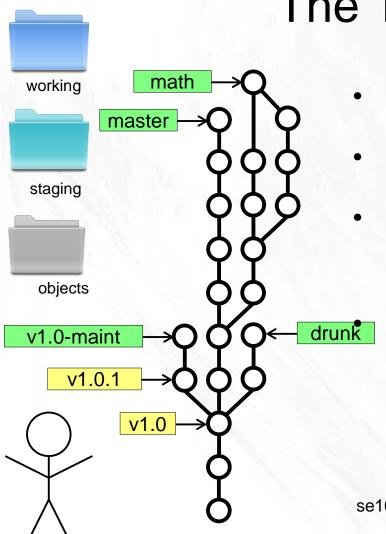




Compressing Blobs



The True Git



- . TADAA!
- This is pretty much Git
- Nicer command line tools for all these operations

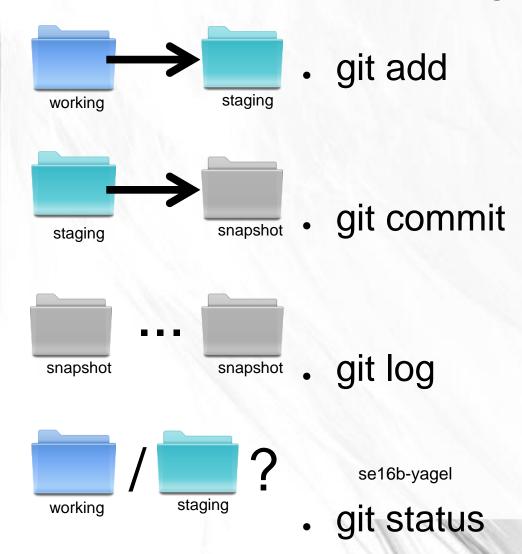
Many, many other tools

Commands: Getting Started

- First, tell Git who you are:
 - git config --global user.name "My Name"
 - git config --global user.email "my@email.address"
- Get help:
 - git <command> -h
 - git help <command>
- Start a new Git repository:
 - git init

121

Commands: Making snapshots



git commit -a

Add the simple scripts I used to do a me
Merge the new object model thing from I
[PATCH] Switch implementations of merg
[PATCH] Port fsck-cache to use parsing fu
[PATCH] Port rev-tree to parsing function
[PATCH] Implementations of parsing function
[PATCH] Header files for object parsing
[PATCH] fix bug in read-cache.c which lo
[PATCH] Fix confusing behaviour of upda
Make "commit-tree" check the input obje
Make "parse_commit" return the "stfl. 22
Do a very simple "merge-base" that finds
Make "rev-tree.c" use the new-and-impro

gith

Commands: Diffing



Commands: Branches & Tags

- git branch
- git branch
branch>

}

git checkout -b ...

- git checkout
branch>
- git tag -l

Commands: Fetching & Merging

• git remote add <name> <URL>

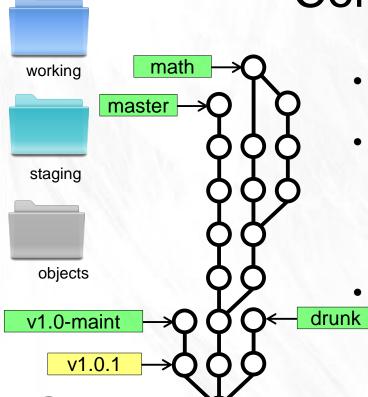
• git fetch <name>



git pull

• git merge <name>/<branch>

Conclusion



v1.0

- Keep this parable in mind
- Git is simple and powerful

One more thing:

git reflog

Where to go next?

- Git homepage: http://git-scm.com
- Pro Git: http://git-scm.com/book
- Git Reference: http://gitref.org
- GitHub: http://github.com
- Gitorious: http://gitorious.org

Questions?

Thanks for your attention!

 These slides are available at: <u>https://github.com/jherland/git_parable</u>

Reach me at <<u>johan@herland.net</u>>



Git Clients (Windows)

- CLI Shell: Git Bash
- Windows Explorer Shell
- Github Desktop (+powershell)
- Bitbucket SourceTree
- IDE Integration
 - Visual Studio (VS2013+ native)
 - Eclipse Egit
 - IntelliJ, Webstrom embedded
 - Brackets plugin

<u>הדגמה</u> \ שב 3- טיפים

- http://gitimmersion.com/
- http://learn.github.com/ (education.. free student account)
- http://help.github.com/create-a-repo/ Local user settings: git config user.name <user> git config user.email user@example.com
- git pull
- git add: add / stage
- git commit –a == add+commit

Git Workflows - שיטות

- Centralized Workflow
- Feature Branch Workflow
- Gitflow Workflow
- Forking Workflow (OSS)
 - ?בפרויקט <u>Pull request</u> •

2.0

2.2

- :Nuget דוגמא: תהליך העבודה בפרויקט Contributing a Bug Fix or Feature
 - קישורים נוספים בויקי

בפעם הבאה

- פרויקט: פיתוח בסבבים
 - 3 'משימה <u>אישית</u> מס' 3
- בדיקות, פיתוח מונחה מפרטים
- שעה שלישית מעבדה (משימה אישית 4, להביא מחשבים)
 - בהמשך
 - git תרחישים נוספים עם ? ? בקרת גרסאות
 - עקרונות תיכון מונחה עצמים ועוד –

לסיכום

- תהליך: בקרת תצורה וגרסאות
 - git / github : כלים •
 - git flow שיטות: למשל •