

Version Control



ΑΝΤΩΝΗΣ ΣΙΔΗΡΟΠΟΥΛΟΣ

Βιβλιογραφία

2

- **Version Control by example, Eric Sink**
 - Διαθέσιμο on-line: <https://ericsink.com/vcbe/index.html>
(Θεωρία για Version Control + Git, SubVersion, Mercurial)
- **Wikipedia**
 - http://en.wikipedia.org/wiki/Software_configuration_management
 - http://en.wikipedia.org/wiki/Revision_control
 - http://en.wikipedia.org/wiki/Distributed_revision_control
 - http://en.wikipedia.org/wiki/Comparison_of_revision_control_software

software configuration management (SCM)

3

- is the task of tracking and controlling changes in the software
- Configuration management practices include
 - **revision control**
 - establishment of baselines
- Software Configuration Management is how you control the evolution of a software project

Στόχοι του SCM

4

- identification - Identifying configurations, configuration items and baselines.
- Configuration control - Implementing a controlled change process. This is usually achieved by setting up a change control board whose primary function is to approve or reject all change requests that are sent against any baseline.
- Configuration status accounting - Recording and reporting all the necessary information on the status of the development process.
- Configuration auditing - Ensuring that configurations contain all their intended parts and are sound with respect to their specifying documents, including requirements, architectural specifications and user manuals.
- Build management - Managing the process and tools used for builds.
- Process management - Ensuring adherence to the organization's development process.
- Environment management - Managing the software and hardware that host the system.
- Teamwork - Facilitate team interactions related to the process.
- Defect tracking - Making sure every defect has traceability back to the source.

Revision control

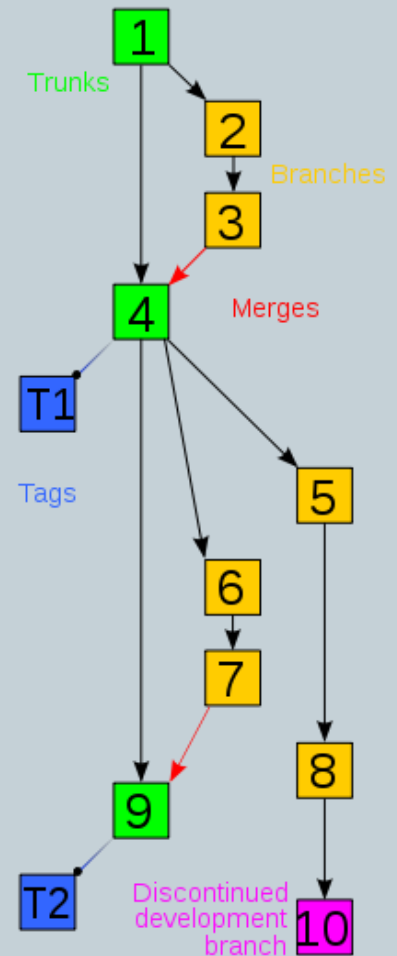
5

- == version control == source control
- is the management of changes to documents, programs, large web sites and other information stored as computer files
- Changes are usually identified by a number or letter code, termed the "revision number", "revision level", or simply "revision"

Example history tree of a revision-controlled project

6

- Κατά την ανάπτυξη ενός software (και γενικότερα αρχείου κειμένου) συνήθως δεν ακολουθείται μια «ευθεία» αλλά δημιουργούνται κλαδιά - ιδίως όταν στην ανάπτυξη αυτή συμμετέχουν πολλά άτομα.



- Version control systems
- Συνήθως είναι αδύνατο (ή πολύ δύσκολο) να αναπτυχθεί κάποιο project χωρίς την χρήση κάποιου βοηθητικού προγράμματος που να καταγράφει τις μεταβολές που έγιναν.
- Οι σουίτες γραφείου (πχ. ms-office, openoffice, kword etc.) έχουν ενσωματωμένη την δυνατότητα για καταγραφή του ιστορικού ενός αρχείου.
- Υπάρχουν εξειδικευμένα προγράμματα για αυτόν τον σκοπό (το source code είναι text αρχεία)

VCS

8

- Τα κύρια πλεονεκτήματα με την χρήση VCS:
 - Σε μια ομάδα προγραμματιστών πετυχαίνουμε παράλληλη εργασία και όχι σειριακή
 - όταν πολλοί προγραμματιστές εργάζονται πάνω στο ίδιο αρχείο χρειαζόμαστε έναν εύκολο τρόπο οι αλλαγές που κάνει ο καθένας να «συνενωθούν».
 - Αποθηκεύονται όλες οι εκδόσεις από οτιδήποτε υπήρξε οποτεδήποτε στο προτζεκτ μας.
 - ✦ ποιος το έκανε
 - ✦ πότε
 - ✦ γιατί

Source-management models

9

- Traditional revision control systems use a centralized model
 - all the revision control functions take place on a server.
- If two developers try to change the same file at the same time, without some method of managing access the developers may end up overwriting each other's work.
- Centralized revision control systems solve this problem in one of two different "source management models": **file locking** and **version merging**.

Source-management models

10

- **file locking**

- Εάν ένας χρήστης «κλειδώσει» κάποιο αρχείο, τότε κανένας άλλος δεν μπορεί να κάνει αλλαγές σε αυτό (μόνο όταν το ξεκλειδώσει)

- **version merging**

- 2 χρήστες κάνουν αλλαγές σε δικά τους αντίγραφα του ίδιου αρχείου. Κατά την ολοκλήρωση της διαδικασίας τα 2 αντίγραφα συνενώνονται σε 1.

Distributed version control System - DVCS

11

- P2P προσέγγιση
- peer's working copy of the codebase is a repository
- synchronization by exchanging patches (change-sets) from peer to peer

Distributed version control System - DVCS

12

differences from a centralized system:

- No canonical, reference copy of the codebase exists by default; only working copies.
- Common operations (such as commits, viewing history, and reverting changes) are fast, because there is no need to communicate with a central server.
- Each working copy effectively functions as a remote backup of the codebase and of its change-history, providing natural protection against data loss.

Three Generations of Version Control

13

| Generation | Networking | Operations | Concurrency | Examples |
|--------------------------------|-------------|--------------------|---------------------|---|
| First | None | One file at a time | Locks | RCS, SCCS |
| Second | Centralized | Multi-file | Merge before commit | CVS, SourceSafe, Subversion, Team Foundation Server |
| Third (DVCS – Distributed VCS) | Distributed | Changesets | Commit before merge | Bazaar, Git , Mercurial |

- **Git:** Το σύστημα που χρησιμοποιείται περισσότερο αυτή τη στιγμή.

Centralized Version Control

14

- 18 basic operations you can do with a version control system

1. create
2. checkout
3. commit
4. update
5. add
6. edit
7. delete
8. rename

9. move
10. status
11. diff
12. revert
13. log
14. tag
15. branch
16. merge
17. resolve
18. lock

Create

15



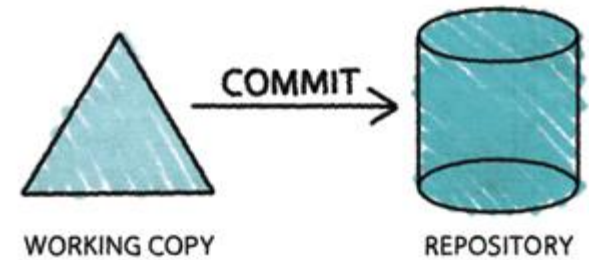
REPOSITORY

- **Create a Repository**
- **Repository**
 - είναι μια 2-διάστατη «οντότητα»: filesystem x time
 - σε ένα repository τίποτε δεν διαγράφεται. ακόμη και αν ο χρήστης διαγράψει ένα αρχείο, το repository μεγαλώνει, διότι αποθηκεύεται η ενέργεια.

- **create a working copy**
- **working copy:**
 - τοπικό αντίγραφο ενός repository επάνω στο οποίο δουλεύει ένας προγραμματιστής.
 - Θα ήταν «αδύνατο» πολύ προγραμματιστές να κάνουν τροποποιήσεις απευθείας επάνω στο repository.
 - Ο κάθε προγραμματιστής έχει το δικό του “working copy”.
 - Το “working copy” δεν είναι μόνο ένα snapshot του repository, περιέχει και άλλες πληροφορίες.
 - ✦ Ο χρήστης Α κάνει ένα “working Copy” (το ονομάζουμε WCA).
 - ✦ ο Α προσθέτει κώδικα στο WCA. Το WCA περιέχει «νέα» πληροφορία.
 - ✦ άλλοι χρήστες ενημερώνουν το Repository. Το WCA περιέχει «νέα» και «παλιά» πληροφορία. πρέπει να ενημερωθεί κατά ένα τμήμα και να ενημερώσει το Repository.

Commit

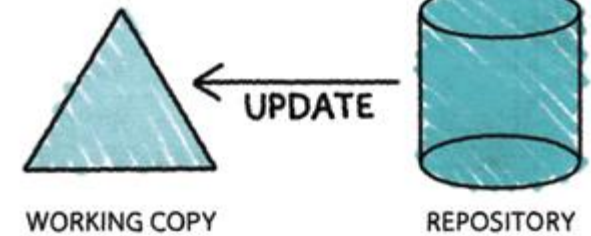
17



- Η ενέργεια ενημέρωσης ενός Repository με τις αλλαγές που έχουν γίνει σε ένα WorkingCopy.
- Ισοδύναμοι όροι: checkin, ci, install, submit or record
- όλα τα νέα VCS κάνουν αυτή την ενέργεια «ατομικά» (atomically).
 - Από ένα changeSet ή θα πετύχουν όλες οι αλλαγές στο Repository ή καμία.
- Συνηθίζεται να απαιτείται να καταγραφεί από τον χρήστη ένα μήνυμα-σχόλιο για τις αλλαγές που κάνει commit.

Update

18



- Η ενέργεια ενημέρωσης ενός **WorkingCopy** με τις αλλαγές που έχουν γίνει σε ένα **Repository**.
- όταν δημιουργείται ένα **WorkingCopy**, αποθηκεύεται και η έκδοση του **Repository** από το οποίο προήλθε (ας την ονομάσουμε **Rx**). Αυτή η έκδοση συνήθως ονομάζεται **parent** (γονική).
- Κατά το **update** εφαρμόζονται στο **WorkingCopy** όλα τα **changeSets** που εφαρμόστηκαν στο **Repository** από την έκδοση **Rx** και μετά.

Add, Edit, Delete, Rename, Move

19

- **όλες αυτές οι ενέργειες καταγράφονται σε ένα ChangeSet.**
- **Rename:**
 - Κάποια VCS δεν έχουν υποστήριξη για rename. Θα το υλοποιήσουν ως Delete και Add.
 - Υπάρχει διαφορά στο χειρισμό του από διάφορα VCS.
- **Move:**
 - Σε κάποια συστήματα θεωρείται ταυτόσημο με το rename.

Status, Diff

20

- Status: Δείχνει περιληπτικά τις αλλαγές που έχουν γίνει σε ένα Working Copy.
- Diff: Δείχνει με λεπτομέρειες τις αλλαγές που έχουν γίνει σε ένα Working Copy.
(γραμμή- γραμμή για κάθε αρχείο)

Revert

21

- Επαναφορά ενός Working Copy στο σημείο που ξεκίνησε και διαγραφή του τρέχοντος ChangeSet.

- Εμφάνιση ΟΛΩΝ των ενεργειών που έχουν γίνει σε ένα Repository. για κάθε ενέργεια εμφανίζονται και τα:
 - ποιος έκανε την ενέργεια (αλλαγή)
 - πότε
 - γιατί (το μήνυμα που κατέγραψε ο χρήστης)

Tag

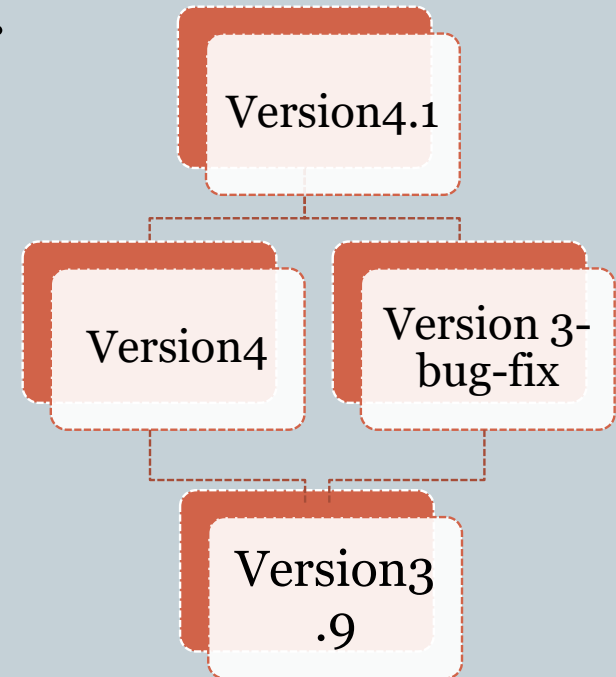
23

- ένα όνομα για μια συγκεκριμένη έκδοση στο Repository. Συνήθως για κάποια σημαντική έκδοση.
- Το όνομα πρέπει να αντιπροσωπεύει την έκδοση, πχ:
 - LAST_VERSION_BEFORE_THE_BIG_BUG
 - FINISHED_SUPPORT_OF_DATABASE_BACKUP
 - Firefox3.2
 - Firefox3.3.2
 - etc.

Branch

24

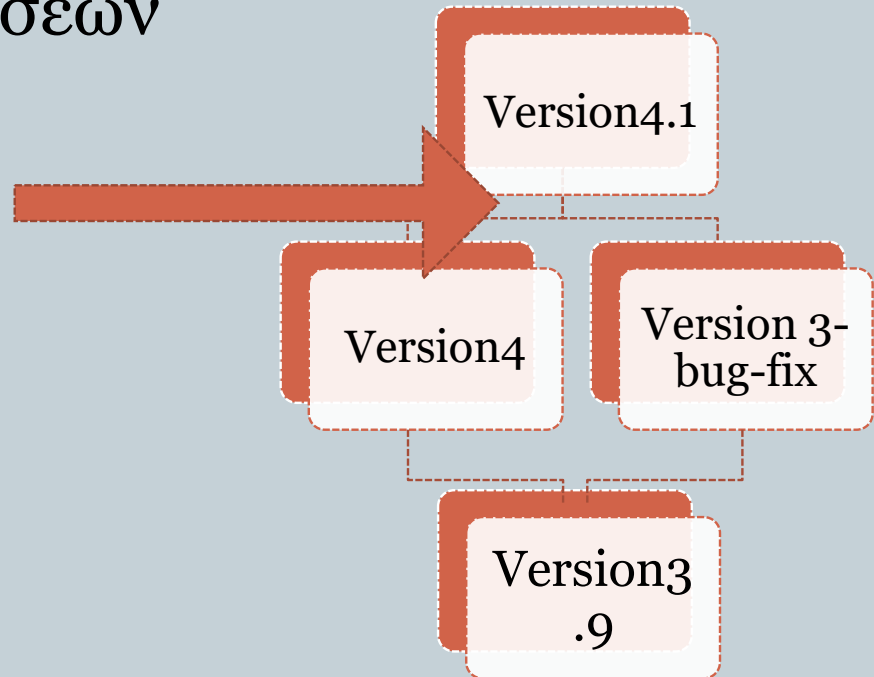
- μια νέα «γραμμή» ανάπτυξης.
 - πχ:



Merge

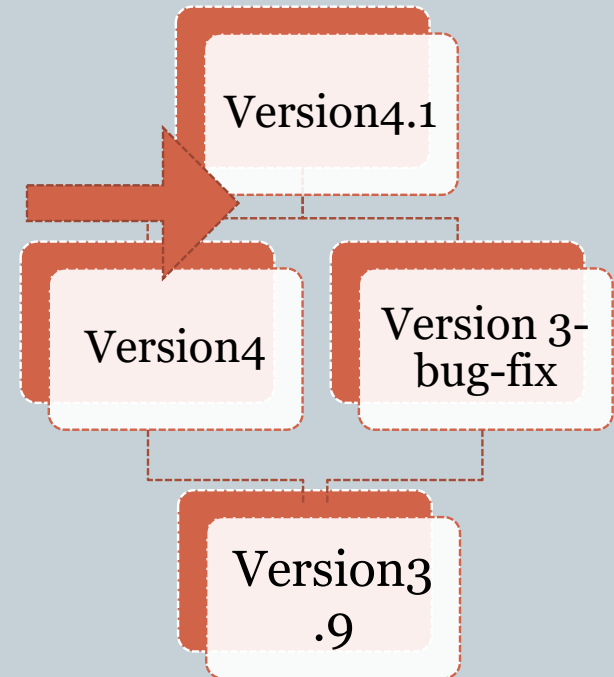
- Εφαρμογή τροποποιήσεων ενός branch σε άλλο

○ πχ:



Resolve

- Διαχείριση των διενέξεων (conflicts)
 - Η ενέργεια του merge θέλουμε να γίνεται αυτόματα. Υπάρχουν όμως περιπτώσεις που δεν μπορεί το σύστημα να αποφασίσει πως θα γίνει το merge. Τότε χρειάζεται η επέμβαση του χρήστη για να λύσει την διένεξη.





- Κλείδωμα ενός αρχείου.
 - Στα νέα συστήματα VCS δεν χρειάζεται η χρήση του. Το VCS μπορεί να αποφασίσει πώς να χειριστεί την σύγχρονη πρόσβαση και αλλαγές.
 - Μπορεί να χρειάζεται στην περίπτωση binary αρχείων (πχ. images) όπου το VCS δεν μπορεί να «συνενώσει» 2 εκδόσεις μιας εικόνας.

Ορολογία - σύνοψη

28

- **Baseline:** An approved revision of a document or source (baselines,===labels===tags)
- **Branch:** A set of files under version control may be branched or forked at a point in time so that, from that time forward, two copies of those files may develop at different speeds or in different ways independently of each other.
- **Change:** A change (or diff, or delta) represents a specific modification to a document under version control.
- **Change list:** a changelist, change set, or patch identifies the set of changes made in a single commit.
- **Checkout :** A check-out (or co) is the act of creating a local working copy from the repository.
- **Commit:** A commit (checkin, ci or, more rarely, install, submit or record) is the action of writing or merging the changes made in the working copy back to the repository.
- **Conflict:** A conflict occurs when different parties make changes to the same document, and the system is unable to reconcile the changes. A user must resolve the conflict by combining the changes, or by selecting one change in favour of the other.

Ορολογία - σύνοψη

29

- **Delta compression:** retains only the differences between successive versions of files. This allows for more efficient storage of many different versions of files.
- **Export:** exporting is the act of obtaining the files from the repository. It is similar to checking-out except that it creates a clean directory tree without the version-control metadata used in a working copy. This is often used prior to publishing the contents, for example.
- **Head (tip):** refers to the most recent commit.
- **Import:** is the act of copying a local directory tree (that is not currently a working copy) into the repository for the first time.
- **Mainline:** Similar to trunk, but there can be a mainline for each branch.
- **Merge:** A merge or integration is an operation in which two sets of changes are applied to a file or set of files
- **Repository:** is where files' current and historical data are stored, often on a server. Sometimes also called a depot.

Ορολογία - σύνοψη

30

- **Resolve:** The act of user intervention to address a conflict between different changes to the same document.
- **Reverse integration :** The process of merging different team branches into the main trunk of the versioning system.
- **Revision (version):** A version is any change in form.
- **Share:** The act of making one file or folder available in multiple branches at the same time. When a shared file is changed in one branch, it is changed in other branches.
- **Tag:** A tag or label refers to an important snapshot in time, consistent across many files. These files at that point may all be tagged with a user-friendly, meaningful name or revision number.
- **Trunk:** The unique line of development that is not a branch (sometimes also called Baseline or Mainline)
- **Update:** An update (or sync) merges changes made in the repository (by other people, for example) into the local working copy.
- **Working copy:** The working copy is the local copy of files from a repository, at a specific time or revision. All work done to the files in a repository is initially done on a working copy, hence the name. Conceptually, it is a sandbox.

DVCS

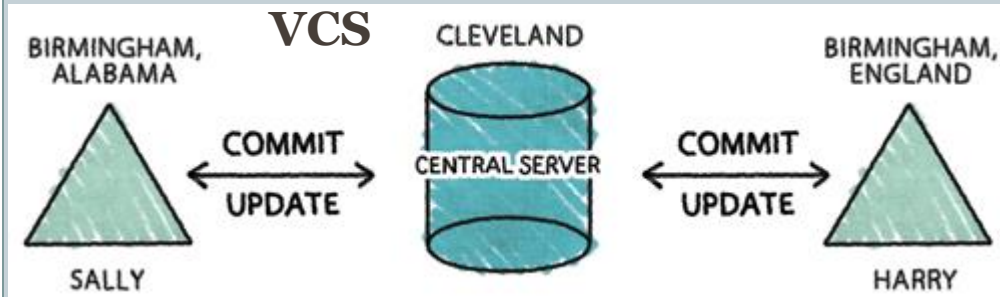
32

- Distributed (Decentralized) Version Control Systems
- 3 ακόμη ενέργειες σε σχέση με VCS
 - clone
 - push
 - pull

Clone

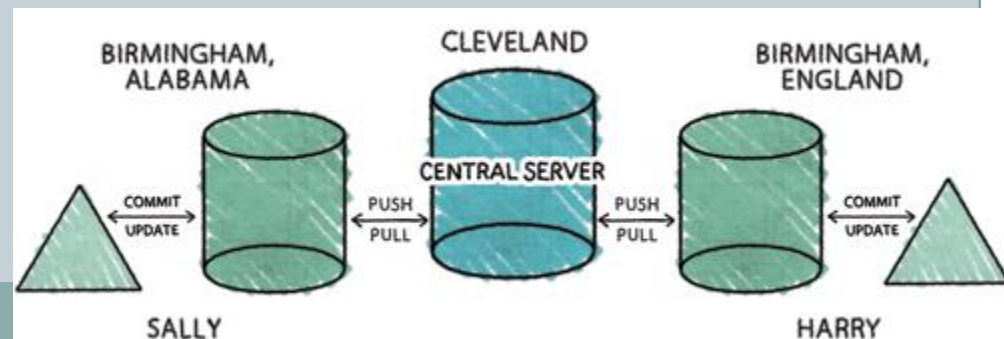
33

VCS



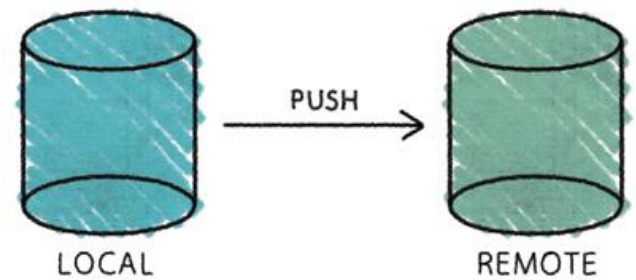
- Κατά το Clone, δημιουργείται ένα αντίγραφο ενός Repository.
- Σε ένα DVCS έχουμε πολλά Repositories. Συνήθως ο κάθε χρήστης έχει το δικό του.

DVCS



Push

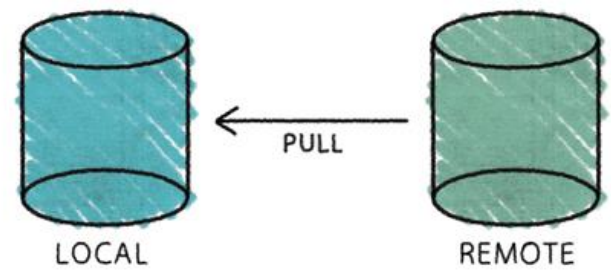
34



- Με το push ενημερώνουμε ένα Remote Repository με τις αλλαγές που έχουν γίνει στο Local Repository.
 - ΠΡΟΣΟΧΗ: όχι με τις αλλαγές που έχουν γίνει στα αρχεία... για να στείλουμε τις αλλαγές των αρχείων θα πρέπει πρώτα να γίνει Commit στο local repository.
 - ΠΡΟΣΟΧΗ2: Μετά το push τα 2 repositories μπορεί να μην είναι ίδια.
 - ✦ Υπάρχει περίπτωση να έχουμε τοπικά επιπλέον δεδομένα (διότι πχ δεν κάναμε push όλα τα branches)
 - ✦ Υπάρχει περίπτωση να έχουμε επιπλέον δεδομένα στο Remote Repository διότι:
 - είτε Δεν είχαμε λάβει όλα τα στοιχεία από το repository
 - είτε έκαναν push και άλλοι χρήστες από άλλα repositories

Pull

35



- Με το pull ενημερώνουμε ένα Local Repository με τις αλλαγές που έχουν γίνει στο Remote Repository.
- ΠΡΟΣΟΧΗ: Μετά το pull τα 2 repositories μπορεί να μην είναι ίδια.
 - ✦ Υπάρχει περίπτωση να έχουμε τοπικά επιπλέον δεδομένα (διότι πχ δεν κάναμε push όλα τα branches)
 - ✦ Υπάρχει περίπτωση να έχουμε επιπλέον δεδομένα στο Remote Repository διότι:
 - είτε Δεν είχαμε λάβει όλα τα στοιχεία από το repository
 - είτε έκαναν push και άλλοι χρήστες από άλλα repositories

Pull & Push

36

- Ούτε με το push αλλά ούτε με το pull υπάρχει η εγγύηση ότι τα 2 repositories έχουν ακριβώς την ίδια πληροφορία.
- Για να πετύχουμε απόλυτο συγχρονισμό σε 2 repositories θα πρέπει πρώτα να κάνουμε pull "τα πάντα" και μετά push "τα πάντα".

DAGs (Directed Acyclic Graphs)

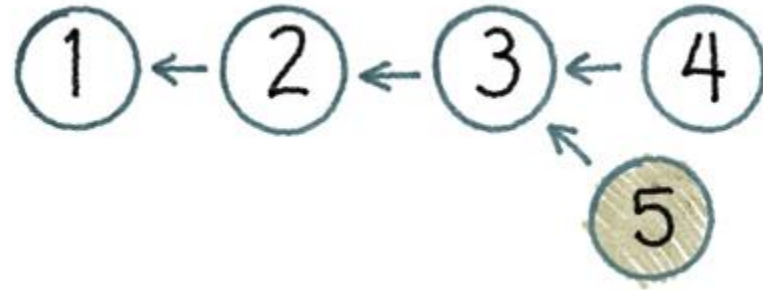
37



- κατά την ανάπτυξη μιας εφαρμογής, θα "θέλαμε" να έχουμε μια και μόνο γραμμή εκδόσεων.
- Αναπαριστούμε το ιστορικό των εκδόσεων με ένα DAG.
 - κόμβοι: εκδόσεις
 - ακμές: δηλώνουν εξάρτηση της μιας έκδοσης από την προηγούμενη. πχ: $2 \rightarrow 1$ σημαίνει ότι η έκδοση 2 είναι επέκταση στην 1.

DAGs

38

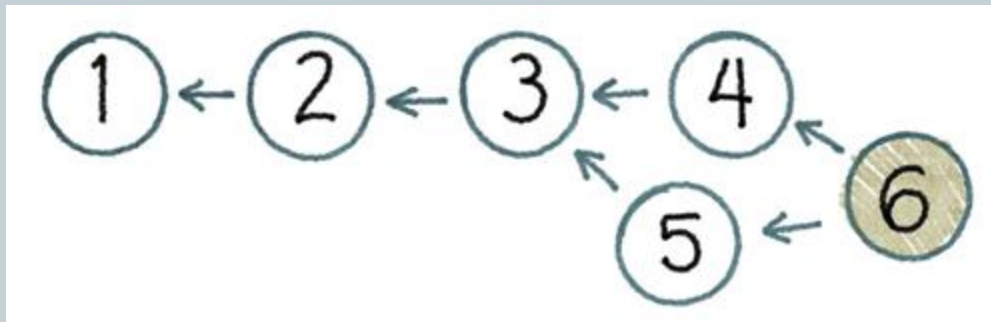


- Το να διατηρηθεί όμως "γραμμή" είναι αδύνατο όταν στο ίδιο project εργάζονται πολλοί προγραμματιστές συγχρόνως.
- έστω ο Α πήρε την έκδοση 3 και κάνει προσθήκες.
- πριν ολοκληρώσει ο Α, θέλει να εργαστεί και ο Β. Παίρνει λοιπόν και αυτός την έκδοση 3.
- ο Α ολοκληρώνει, και "στέλνει" την έκδοσή του (4)
- ο Β ολοκληρώνει, και "στέλνει" και αυτός την έκδοσή του (5)
- Η (5) όμως δεν είναι συνέχεια της 4 αλλά της 3 !!!! ποια είναι η τελευταία έκδοση του προγράμματός μας?

DAGs

39

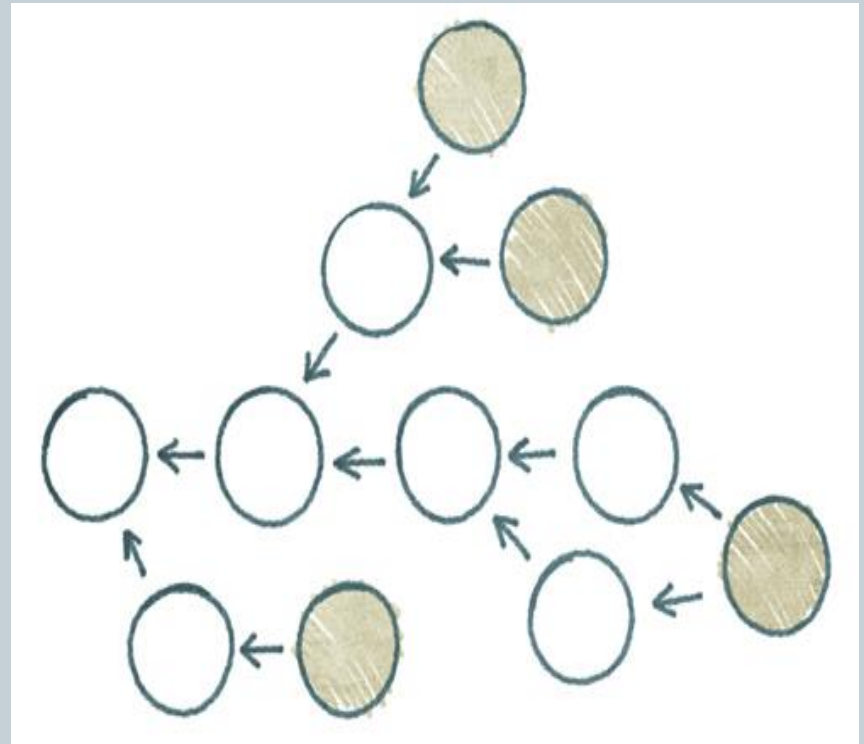
- Τότε πρέπει να δημιουργηθεί μια νέα έκδοση που να περιέχει τις βελτιώσεις της 4 αλλά και της 5.
- αυτό μπορεί να γίνει σχετικά εύκολα (ακόμη και χειροκίνητα) !!!!



DAGs

40

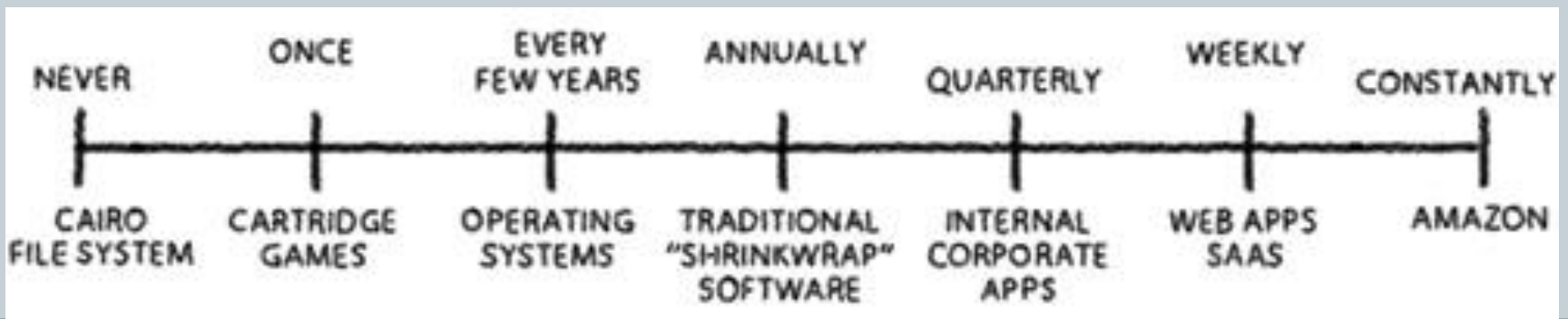
- Τι γίνεται όμως αν έχουμε πολλούς προγραμματιστές?
- Το merging μπορεί να επιτευχθεί από το DVCS επειδή διατηρείται το DAG.
- Προσοχή όμως: δεν είναι καλό να το "παρακάνουμε" με τα παρακλάδια.



Release

41

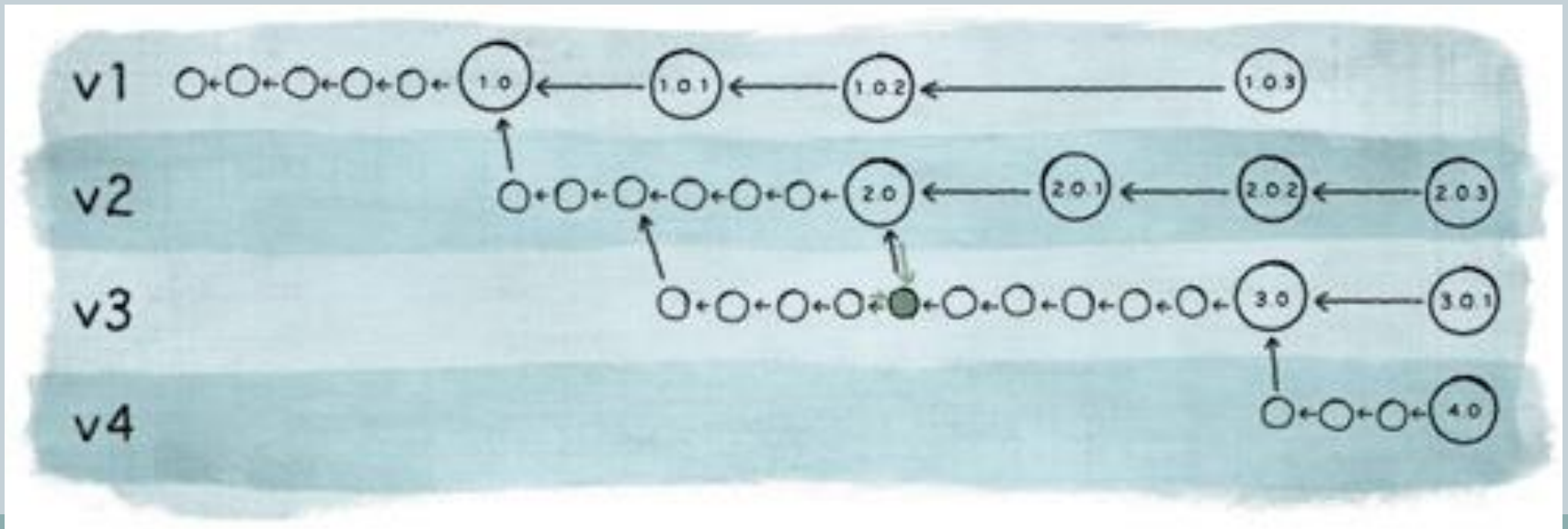
- Μια έκδοση που "δίνουμε" προς τον έξω κόσμο.
- Το πόσο συχνά δημιουργούνται Releases εξαρτάται από πολλούς παράγοντες.



Release

42

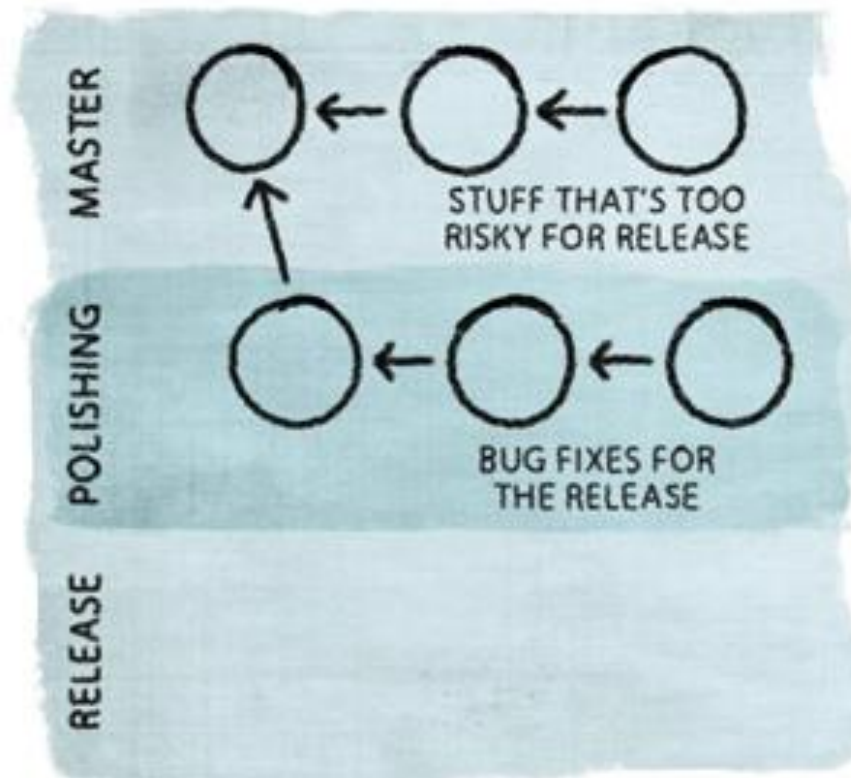
- Προβληματικό σχήμα:
 - Η έκδοση 2.0.2 δεν περιλαμβάνει τις "βελτιώσεις" της 1.0.1



Polishing Branch

43

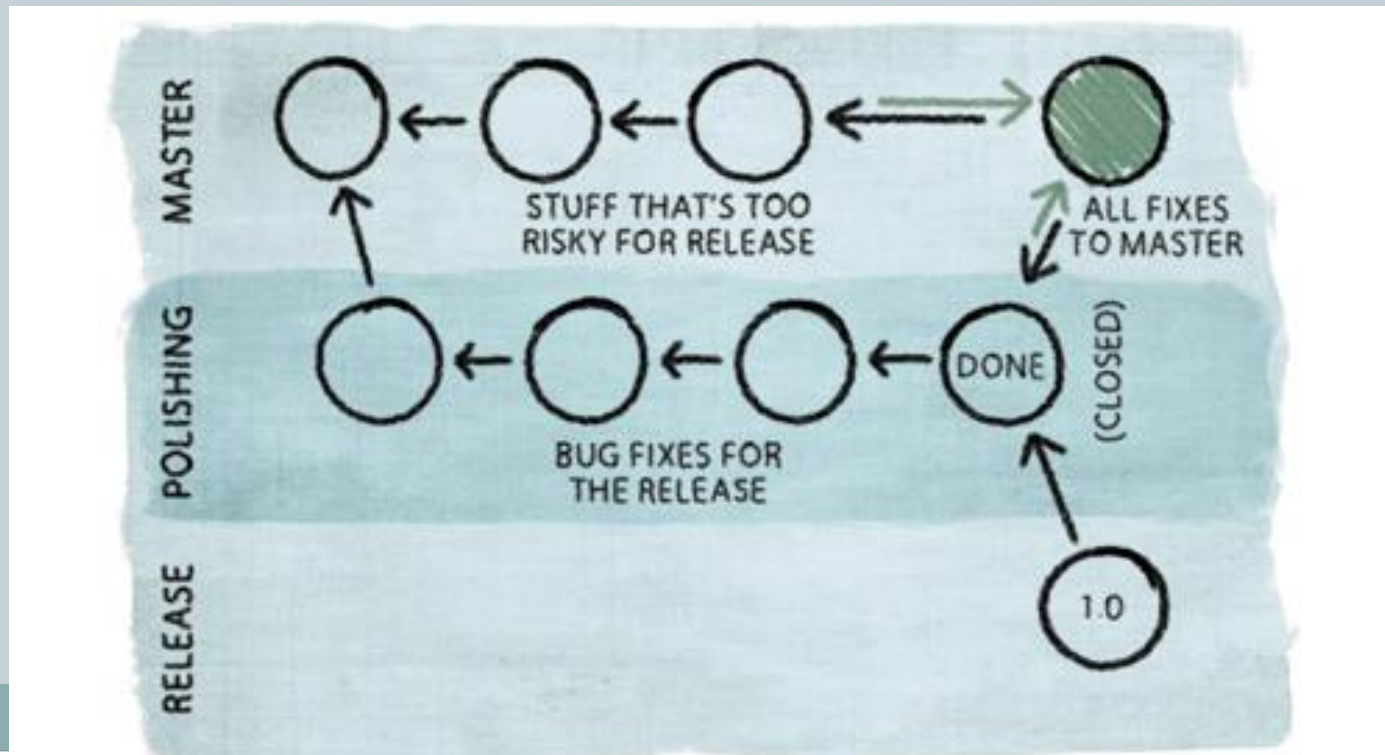
- Η έκδοση πάνω στην οποία προετοιμάζεται ένα Release.



Release Branch

44

- Αφού ολοκληρωθούν οι διορθώσεις, δημιουργείται το Release και συγχρόνως ενσωματώνονται οι αλλαγές στην κύρια έκδοση.

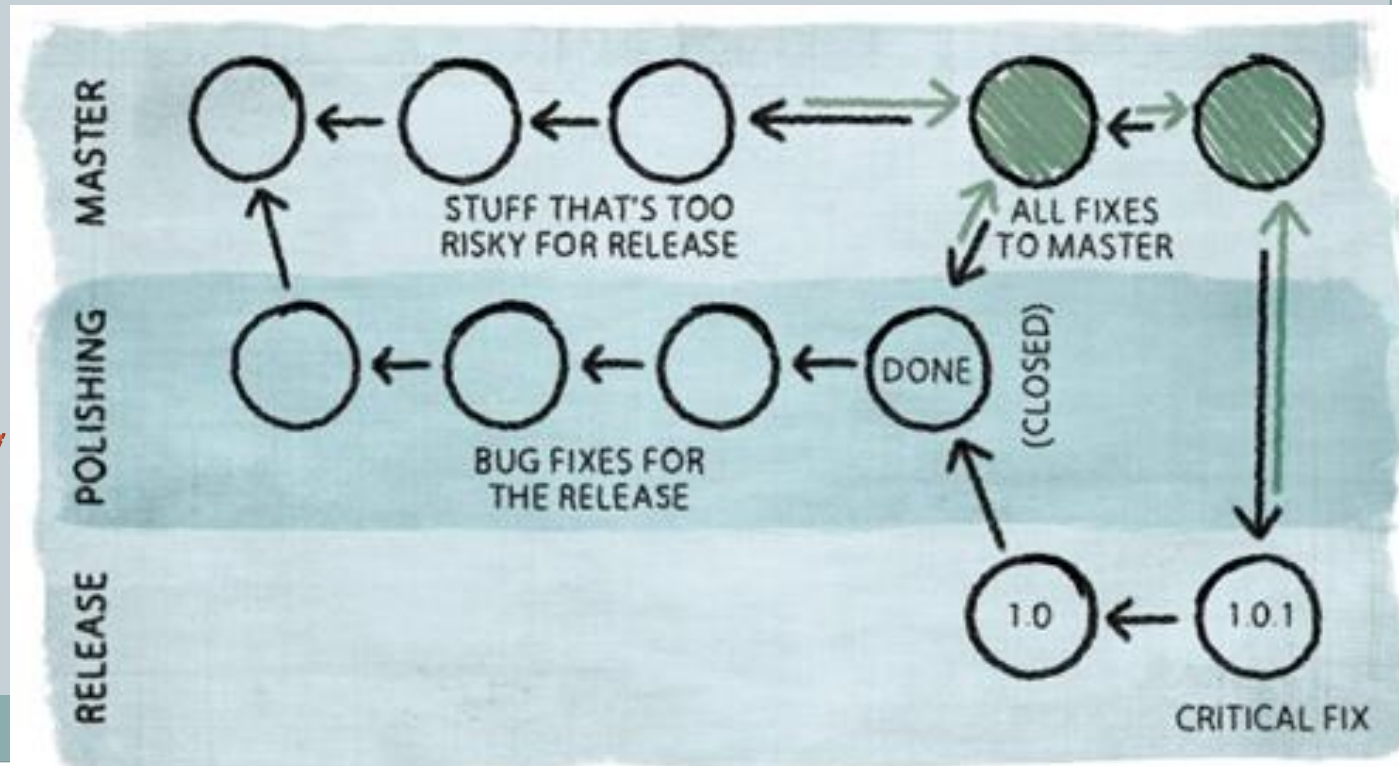


Release Branch

45

- Το Release Branch μπορεί να εξακολουθεί να διορθώνεται αλλά συγχρόνως ενσωματώνονται οι αλλαγές στην κύρια έκδοση.

Δεν θέλουμε το DAG να ξεφύγει από το να μοιάζει με γραμμή ή να δίνει την αίσθηση μιας κατεύθυνσης (όχι πολλές)

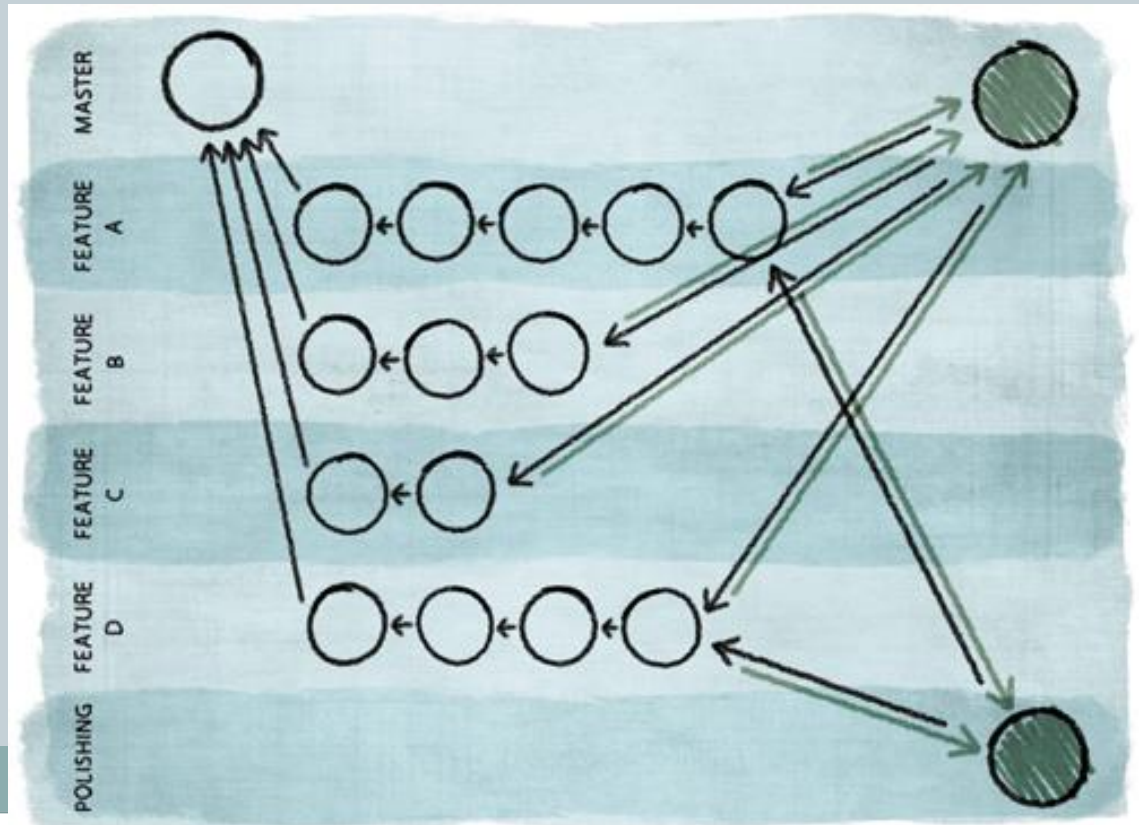


Feature Branches

46

- Μερικές φορές οι δυνατότητες ενός προγράμματος είναι ανεξάρτητες μεταξύ τους.... Τότε μπορεί να επιλέξουμε να δημιουργήσουμε ένα Branch ανα δυνατότητα.

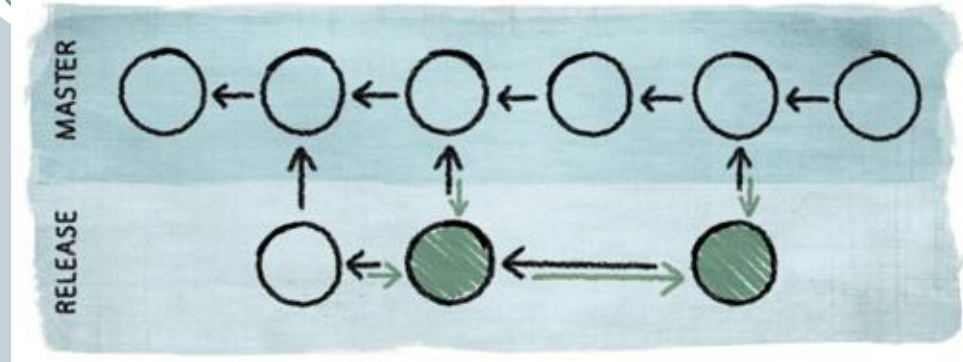
Παρόλα αυτά
το DAG δίνει
την αίσθηση
ανάπτυξης
προς μια
κατεύθυνση.



Web case

47

- With a Web-based application, the notion of a “release” is different, so the way we use the VCS is different.



- For web developers, “release” means deploying new code to the website. That new code immediately becomes available to all users.
- The website has only one version of the software on it. Everybody is always using the latest version. There is no obligation to continue providing support to users who chose to stay with an older version instead of upgrading to the new release.
- Users are generally unaware of versioning. Version numbers are not part of the user’s consciousness. People who talk at conferences about their experience with Google Calendar do not ask each other which version they are running.
- For web developers, there isn’t much overhead in doing a release, so they can deploy new versions with new features and fixes as often as makes sense. Some major websites deploy changes multiples times per day.

Χρήση Github

48

- Το github είναι ένα site που επιτρέπει στους χρήστες του την δημιουργία git repositories. Παρέχει πρόσβαση μέσω web.
- Εγγραφή στο site. Οι δωρεάν λογαριασμοί κάνουν μόνο public repositories.
- Student account. Οδηγίες στη σελίδα: <https://help.github.com/en/github/teaching-and-learning-with-github-education/applying-for-a-student-developer-pack> . Για ακαδημαϊκή χρήση, δίνονται κάποιες επιπλέον δυνατότητες.
- Ψάξτε να γραφτείτε στο “Organization”: iee-ihu-gr-course1941. Θα το χρησιμοποιήσουμε για να ομαδοποιήσουμε τα projects.

Εργαλεία στο pc μας

49

- Git!!!: είναι command line σύνολο εργαλείων.
- Github: Server που επιτρέπει την δημιουργία git repositories.
- Github Desktop: Γραφικό περιβάλλον ειδικά για το github. Περιλαμβάνει εσωτερικά την εγκατάσταση του git.
 - - “Κρύβει” αρκετές δυνατότητες του git.
 - + Έτοιμη ρύθμιση για το github (και μόνο).
- Git Extensions: Γραφικό περιβάλλον γενικά για git.
 - + Περιέχει όλες τις δυνατότητες του git.
 - + Δείχνει γραφικά το DAG
 - - Δύσκολο (?) το config του github.
 - - Πρέπει να εγκατασταθούν: git + kdiff3 ξεχωριστά.