

SUBVERSION

Διδάσκοντες: Π. Αγγελάτος, Δ. Ζήνδρος
Επιμέλεια διαφανειών: Δ. Ζήνδρος

Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών



Στόχος της ώρας

- Το πρόβλημα εκδόσεων
- Έλεγχος εκδόσεων
- Subversion
- Repositories
- Αμμοδοχεία
- Check out
- Commit
- Update
- Ιστορικό
- Διενέξεις
- Χρήση του subversion ως χρονομηχανή

Χρήσεις subversion

Χρησιμοποιούμε subversion ως...

- «Μηχανή χρόνου» για αρχεία και φακέλους
- Πλατφόρμα συνεργασίας με άλλους

Αλλά όχι αν...

- Θέλουμε να ανταλλάξουμε αρχεία μία φορά
 - Γι' αυτό υπάρχει το Skype, το FTP, το rsync...

Διότι υπάρχει ένα κόστος συντήρησης

Χρήσεις subversion

- Ατομικά
 - Χρονομηχανή
 - Συγχρονισμός κώδικα σε πολλούς υπολογιστές
- Ομαδικά
 - Όλα τα παραπάνω
 - Ανταλλαγή κώδικα
 - Αυτόματη διαχείριση διενέξεων
 - Ποιος έγραψε ποιον κώδικα;
 - Αντίγραφα ασφαλείας



collaboration
is everything

Server

- Υπεύθυνος για να αποθηκεύσει τον κώδικά μας
- Αποθηκεύει **όλες** τις λεπτομέρειες για τον κώδικά μας
- Συμπεριλαμβανομένου και του **ιστορικού**
- Τρέχει σε έναν υπολογιστή που είναι 24/7 ανοιχτός
- Χρησιμοποιεί το πρόγραμμα server “subversion”
- Ο κάθε client συνδέεται σε αυτόν τον server

Client

- Αντιστοιχεί σε έναν μηχανικό
- Είναι εγκατεστημένος σε έναν απλό υπολογιστή
- Αποθηκεύει ένα αντίγραφο του κώδικα
- Τρέχει το πρόγραμμα “svn”

Αμμοδοχείο

- Το αντίγραφο του κώδικα που έχει ένας μηχανικός
- **Δεν** αποθηκεύει πλήρες ιστορικό του κώδικα
- Είναι μόνο η **νεότερη** έκδοση του κώδικα

Ομάδα ανάπτυξης

Μηχανικός #1

Αμμοδοχείο

Μηχανικός #2

Αμμοδοχείο

Μηχανικός #3

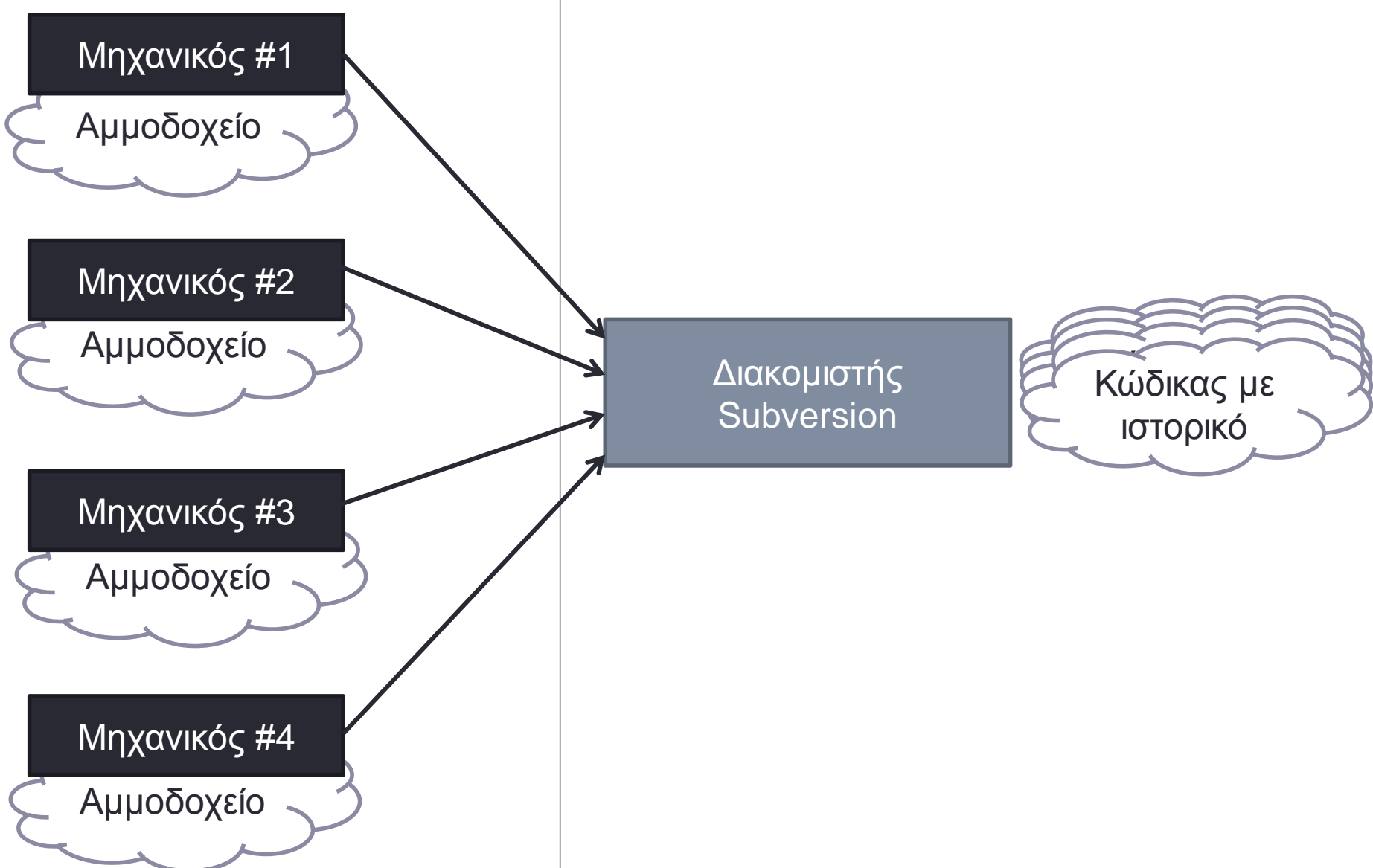
Αμμοδοχείο

Μηχανικός #4

Αμμοδοχείο

Διακομιστής
Subversion

Κώδικας με
ιστορικό



Αρχιτεκτονική client-server

- Το subversion ακολουθεί αρχιτεκτονική client-server
- Ένας server είναι υπεύθυνος για τον κώδικα
- Οι clients κρατάνε απλώς αντίγραφα
- Γι' αυτό το subversion είναι **κεντρικός έλεγχος εκδόσεων**

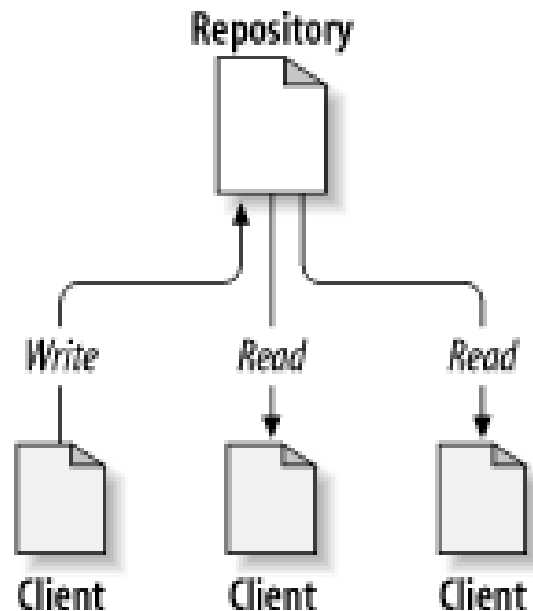
Εναλλακτικές

Υπάρχουν και άλλα συστήματα ελέγχου εκδόσεων

- CVS
 - Ο «μπαμπάς» του SVN. Δύσχρηστο.
- Git
 - Αποκεντροποιημένος έλεγχος εκδόσεων, δεν υπάρχει server.
- Mercurial

Ένα σύστημα αρχείων

- Το subversion μοιάζει με διακομιστή αρχείων
- Οι clients **γράφουν** και **διαβάζουν** από το repository



Ιστορικό

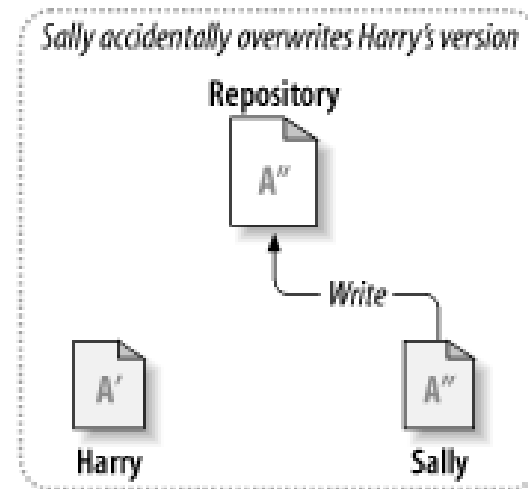
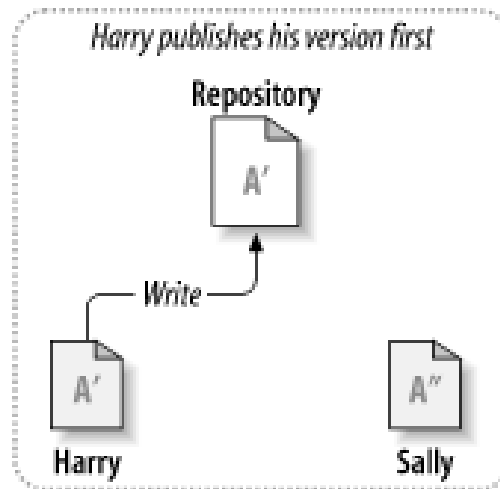
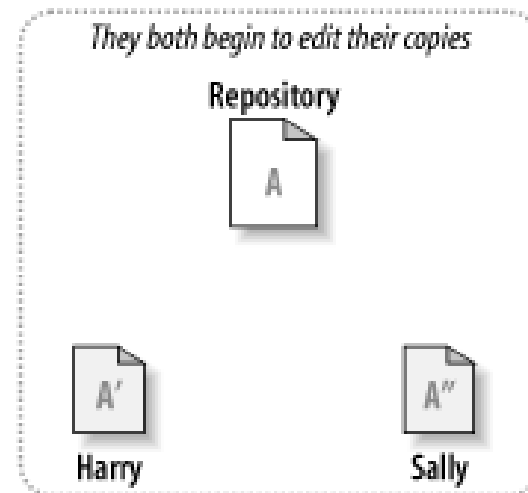
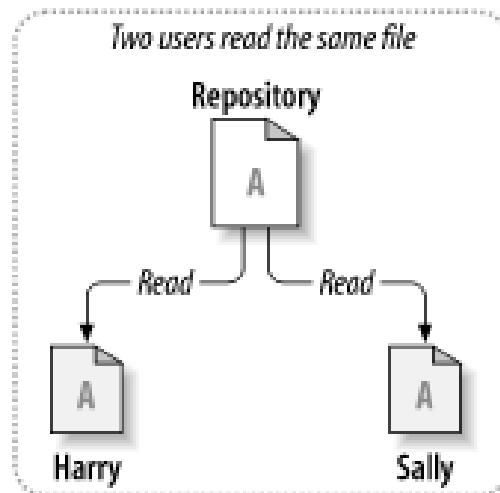
- Διαφέρει από ένα σύστημα αρχείων
- **Θυμάται** όλες τις αλλαγές:
 - Δημιουργία αρχείων
 - Αλλαγές αρχείων
 - Αλλαγές φακέλων
 - Διαγραφές
 - Μετακινήσεις
 - Τα πάντα

Το πρόβλημα

- Η ομάδα έχει ένα αρχείο index.php
- Ο **Harry** έχει το νεότερο index.php
- Η **Sally** έχει επίσης το νεότερο index.php
- Ο Harry κάνει κάποιες αλλαγές στο αρχείο του
- Η Sally κάνει επίσης κάποιες αλλαγές στο αρχείο της

- Πώς ανταλλάσσουν αυτές τις αλλαγές;
- Πώς δεν **διαγράφει** ο ένας τις αλλαγές του άλλου;

Το πρόβλημα



Η ιδέα του κλειδώματος

- Όταν κάποιος πρόκειται να επεξεργαστεί ένα αρχείο, το **κλειδώνει** ώστε να μην το αλλάξει κάποιος άλλος
- Στη συνέχεια **αλλάζει** το αρχείο
- Όταν στέλνει τις αλλαγές του, **ξεκλειδώνει** το αρχείο για να το επεξεργαστεί κάποιος άλλος
- Όταν κάποιος πάει να κλειδώσει ένα αρχείο που είναι ήδη **κλειδωμένο**, δημιουργείται **σφάλμα**

Προβλήματα με το κλείδωμα

- Κάποιος μπορεί να **ξεχάσει** να ξεκλειδώσει
- Συχνά μπορούμε δύο άτομα να κάνουμε αλλαγές **στο ίδιο** αρχείο χωρίς πρόβλημα, αλλά το κλείδωμα μας το **απαγορεύει** και μας καθυστερεί
- Το κλείδωμα **ενός** αρχείου δεν σημαίνει ότι οι αλλαγές μας θα είναι συμβατές με αλλαγές σε **άλλα**

Η λύση του τοπικού αντιγράφου

- Το subversion δεν χρησιμοποιεί κλειδώματα
- Όλοι διαβάζουν τις νεότερες εκδόσεις του αρχείου από το repository
- Δημιουργούν αντίγραφα στο τοπικό τους αμμοδοχείο
- Επεξεργάζονται ο καθένας το αντίγραφό του ανεξάρτητα
- **Ανεβάζει** ο πρώτος τις αλλαγές του στο repository
- Όταν ο δεύτερος **ανεβάσει** αλλαγές, αυτό αποτυγχάνει
- Θα πρέπει να **κατεβάσει** πρώτα τις αλλαγές του άλλου
- Κατά το κατέβασμα αλλαγών, ενώνονται οι δύο αλλαγές
- Στη συνέχεια μπορεί να ανεβάσει την νέα έκδοση

Επικοινωνία svn client-server

- Ανάγνωση:
 - Είτε από όλους (δημόσιος/ανοιχτός κώδικας)
 - Είτε από συγκεκριμένη ομάδα
- Πιστοποίηση για ανάγνωση ή εγγραφή:
 - Με κωδικό πρόσβασης
 - Με RSA κλειδί (ασφαλέστερο)
- Πρωτόκολλα ανταλλαγής κώδικα:
 - http
 - https (κρυπτογραφημένο)
 - svn
 - svn+ssh (κρυπτογραφημένο)

Repository URL

- Αντιπροσωπεύει ένα μέρος όπου είναι αποθηκευμένος ο κώδικας
- Περιλαμβάνει:
 - Το πρωτόκολλο (http://, svn://, https://, κλπ.)
 - Το όνομα χρήστη του μηχανικού
 - Το όνομα του διακομιστή
 - Το όνομα του repository
- π.χ. `svn+ssh://dionyziz@example.com/repository`

Checkout

- Checkout ονομάζεται η διαδικασία δημιουργίας αμμοδοχείου

(παράδειγμα)

Commit

- Οι αλλαγές στο αμμοδοχείο δεν δημοσιεύονται στους άλλους
- Είναι οι δικές σου, προσωπικές αλλαγές, μέχρι να αποφασίσεις να τις δημοσιεύσεις
- Η διαδικασία δημοσίευσης με την ομάδα σου ονομάζεται “commit”

(παράδειγμα)

Update

- Οι αλλαγές που **ανέβασαν** οι άλλοι λαμβάνονται μόνο όταν το αποφασίσεις
- Το **κατέβασμα** αλλαγών λέγεται update

(παράδειγμα)

Κρυφός φάκελος .svn

- Υπάρχει μέσα σε κάθε φάκελο του αμμοδοχείου
- Αποθηκεύει meta-πληροφορίες που χρειάζεται το svn

Τι μπορεί να συμβεί

1. Το αρχείο στο αμμοδοχείο είναι **ίδιο** με το αρχείο στο repository. Update και commit δεν κάνουν τίποτα.
2. Το αρχείο στο αμμοδοχείο έχει **αλλάξει τοπικά**. Update δεν κάνει τίποτα, commit στέλνει τις αλλαγές.
3. Το αρχείο στο repository είναι πιο καινούργιο. Commit δεν κάνει τίποτα, update ενημερώνει το τοπικό αρχείο.
4. Το αρχείο στο repository είναι πιο καινούργιο, αλλά το αρχείο **έχει αλλάξει και τοπικά**. Το commit αποτυγχάνει, επειδή πρέπει πρώτα να γίνει update. Το update ενώνει τις αλλαγές.

Εκδόσεις

- Κάθε φορά που συμβαίνει ένα commit, αυξάνεται ο **αριθμός έκδοσης**
- Ο αριθμός έκδοσης είναι **κοινός** για όλα τα αρχεία
- Ξεκινά από το 1 και αυξάνεται κατά 1 κάθε φορά
- Αν αλλάξω **ένα** αρχείο, αυξάνεται ο αριθμός έκδοσης όλων των αρχείων
- Ο αριθμός έκδοσης αφορά το repository, όχι συγκεκριμένα αρχεία

Το commit και το update ξεχωρίζουν

- Μπορώ να κάνω commit ένα αρχείο χωρίς να κάνω update τα υπόλοιπα
- Έτσι, κάποια αρχεία μπορεί να έχουν διαφορετικές εκδόσεις
- π.χ. μπορεί να κάνω commit το index.php στην έκδοση 315
- ενώ δεν έχω κάνει update το download.php και να παραμείνει στην έκδοση 300

Subversion clients

- **svn**
 - Client σε κονσόλα για Windows, Linux, Mac
- **TortoiseSVN**
 - Client με γραφικό περιβάλλον για Windows
- Πολλές άλλες επιλογές για εργαλεία με γραφικά και κονσόλα

Δουλεύοντας με το CLI svn

- **svn help** εντολή
- Εξηγεί την σύνταξη μίας εντολής

- Π.χ.
- `svn help commit`
- `svn help update`

Checkout

svn checkout URL

Κύκλος εργασίας με svn

(1) Ενημέρωση

Κάνε **svn update** για να πάρεις τις αλλαγές των άλλων

Κύκλος εργασίας με svn

(2) Αλλαγές

Κάνε τις αλλαγές στον κώδικα που θέλεις και δοκίμασε ανεξάρτητα στο αμμοδοχείο σου.

Εκτός από αλλαγές αρχείων μπορεί να περιλαμβάνει και προσθήκες, διαγραφές, αντιγραφές, μετακινήσεις αρχείων.

svn add; svn delete; svn copy; svn move

Κύκλος εργασίας με svn

(2) Προβολή αλλαγών

Δες τις αλλαγές σου πριν τις ανεβάσεις.

Αυτό είναι εφικτό με svn **status** και svn **diff**.

Κύκλος εργασίας με svn

(3) Ακύρωση αλλαγών

Αν δεν σου αρέσουν οι αλλαγές που έκανες, μπορείς να τις ακυρώσεις.

Αυτό είναι εφικτό με svn **revert**.

Κύκλος εργασίας με svn

(4) Ενημέρωση ξανά

Κατέβασε τις αλλαγές που έκαναν οι άλλοι στο μεταξύ με **svn update**.

Αυτό περιλαμβάνει πιθανώς αλλαγές στα αρχεία που άλλαξες κι εσύ. Συνεπώς ενδέχεται να πρέπει να διαχειριστείς διενέξεις.

Κύκλος εργασίας με svn

(5) Ανέβασμα αλλαγών

Ανέβασε τις αλλαγές σου με **svn commit**.

Κύκλος εργασίας με svn

Συνήθως η διαδικασία περιορίζεται:

- 1) `svn update`
- 2) Επεξεργασία αρχείων
- 3) `svn commit`

Αλλαγές δομής αρχείων

- Όταν χρησιμοποιούμε svn, οι **προσθήκες, διαγραφές, αντιγραφές, και μετακινήσεις** αρχείων και η **δημιουργία φακέλου** δεν γίνονται με το συνηθισμένο τρόπο.
- Πρέπει να γίνονται μέσω του svn
- Αυτό διότι θέλουμε η ομάδα μας να **ενημερωθεί** και για αυτές τις αλλαγές.
- Επιπλέον θέλουμε και αυτές οι αλλαγές να έχουν **ιστορικό**.

Αλλαγές δομής αρχείων

- **svn add**
 - Προσθέτει ένα αρχείο
- **svn rm**
 - Διαγράφει ένα αρχείο
- **svn mv**
 - Μετακινεί ένα αρχείο
- **svn cp**
 - Αντιγράφει ένα αρχείο
- **svn mkdir**
 - Δημιουργεί ένα φάκελο

Αλλαγές δομής αρχείων

- Τα `svn add`, `svn rm`, `svn mv`, `svn cp`, `svn mkdir` δρουν τοπικά. Θα πρέπει πάλι να κάνουμε `svn commit` για να σταλούν οι αλλαγές δομής που κάναμε.

Προβολή αλλαγών

- Θέλουμε να προβάλλουμε τις αλλαγές που έχουμε κάνει τοπικά πριν τις ανεβάσουμε.
- **svn status**
 - Δείχνει τα αρχεία που άλλαξαν. Μας δείχνει με συγκεκριμένο τρόπο τι έχει συμβεί:
A = Αρχείο προστέθηκε
M = Αρχείο άλλαξε
D = Αρχείο διαγράφηκε
? = Αρχείο υπάρχει, αλλά δεν έχει προστεθεί
- **svn diff**
 - Δείχνει τι άλλαξε μέσα στα αρχεία

Ακύρωση αλλαγών

- Θέλουμε να ακυρώσουμε αλλαγές που κάναμε τοπικά και τελικά να μην τις ανεβάσουμε.
- **svn revert**
- **(παράδειγμα)**

Πού θα βρω SVN server?

Αν το Project μου είναι **ανοιχτού κώδικα...**

Δωρεάν:

- SourceForge.net
- Google Project Hosting

Αν το Project μου είναι **κλειστού κώδικα...**

Δωρεάν:

- Στήνω το δικό μου svn server

Εμπορικές λύσεις:

- Beanstalk, κλπ.

Διενέξεις

- Ο Α και ο Β κάνουν **svn update** στο αρχείο index.php
- Ο Α **αλλάζει** το index.php
- Ο Β **αλλάζει** το index.php
- Ο Α κάνει **svn commit** το index.php
- Ο Β κάνει **svn commit** το index.php αλλά **αποτυγχάνει**

Διενέξεις: Συνήθης περίπτωση

- Ο A και ο B έχουν **συμβατές** αλλαγές
- π.χ. A στην αρχή του αρχείου, B στο τέλος του αρχείου
- Τότε το `svn` ενώνει τις αλλαγές αυτόματα
- Ο B κάνει **svn update**
- Οι αλλαγές του A και του B ενώνονται στην τοπική έκδοση του B
- Ο B κάνει **svn commit**
- Πλέον το repository έχει την ενωμένη έκδοση των δύο
- Ο A κάνει **svn update** και κατεβάζει την ενωμένη έκδοση

Διενέξεις: Δύσκολη περίπτωση

- Ο Α και ο Β έχουν **ασυμβίβαστες** αλλαγές
- π.χ. έχουν αλλάξει το ίδιο τμήμα του αρχείου
- Συνήθως όταν προσπαθούν να λύσουν το ίδιο πρόβλημα
- Η διένεξη πρέπει να λυθεί από άνθρωπο
- Ο Β κάνει **svn update**
- Το subversion τον ειδοποιεί για **μη επιλύσιμη διένεξη** (conflict)

Διενέξεις: Δύσκολη περίπτωση

- Ο Β επιλέγει:
 - Να απορρίψει τις αλλαγές του Β εντελώς και να κρατήσει την έκδοση Α
 - Να απορρίψει τις αλλαγές του Α εντελώς και να κρατήσει την έκδοση Β
 - Να φτιάξει «στο χέρι» μία έκδοση που περιλαμβάνει **τις αλλαγές και των δύο**
- Αφού επιλέξει, χρησιμοποιεί το **svn resolve** για να δηλώσει ότι το πρόβλημα λύθηκε
- Κάνει **svn commit** την έκδοση που επέλεξε
- Ο Α κάνει **svn update** και κατεβάσει την νέα έκδοση

Μηνύματα commit

- Δηλώνουν τον λόγο που άλλαξε ένα αρχείο
- Σύντομο περιγραφικό μήνυμα που βλέπει η ομάδα
- Χρήσιμο όταν κοιτάμε το ιστορικό

π.χ. “Resolved password length bug”

π.χ. “Added ability to change password“

Χρονομηχανή

- Το svn λειτουργεί ως χρονομηχανή
- Μπορούμε να δούμε την λίστα με όλες τις αλλαγές χρησιμοποιώντας **svn log**
- Περιλαμβάνει:
 - Ποιος άλλαξε ένα αρχείο
 - Πότε το άλλαξε
 - Την περιγραφή που είχε δώσει
- Μπορεί να τρέξει σε ένα συγκεκριμένο αρχείο ή σε φάκελο για να δούμε όλο το ιστορικό του

Χρονομηχανή

- Μπορούμε να «δούμε» μία παλιότερη έκδοση ενός αρχείου
- Με **svn update --revision** μεταβαίνουμε σε μία παρελθοντική έκδοση του αρχείου. Όσο είμαστε «στο παρελθόν» δεν μπορούμε να κάνουμε αλλαγές.
- Με απλό **svn update** επανερχόμαστε στο παρόν.

**TOP
SECRET**

Tips για επιτυχία στο λογισμικό

- Κάντε χρήση **version control**
- Κρατήστε μία **λίστα με bugs**
 - Για αρχή ένα .txt
 - Αργότερα trac, PivotalTracker, mantis, bugzilla, κλπ.
- **Διορθώστε** τα bugs πριν γράψετε νέο κώδικα
- Γράψτε ένα **πλάνο** με στόχους
 - Για αρχή σε ένα .txt
 - Και κρατήστε το **ενήμερο**
- Γράψτε πρώτα το API σας **σε κείμενο**
 - **Πριν** αρχίσετε να γράφετε κώδικα

Μάθαμε

- Το πρόβλημα εκδόσεων
- Έλεγχος εκδόσεων
- Subversion
- Repositories
- Αμμοδοχεία
- Check out
- Commit
- Update
- Ιστορικό
- Διενέξεις
- Χρήση του subversion ως χρονομηχανή



Συγχαρητήρια!

**Μπορείτε πλέον να συνεργάζεστε με
άλλους μηχανικούς λογισμικού!**

Την επόμενη φορά...

- Workshop ανάπτυξης πλήρους εφαρμογής
- Εφαρμογή για photo sharing
- Flickr-style

- **Στις 28 Ιανουαρίου 2011:**
- **Παρουσίαση τελικών εργασιών**
- **Προσκαλέστε τους φίλους σας! :-)**