

# ΕΚΠΑΙΔΕΥΣΗ ΣΤΙΣ ΕΠΙΧΕΙΡΗΣΕΙΣ

21

- ▶ Στις μεγάλες επιχειρήσεις γίνεται συστηματικά
  - ▶ Σεμινάρια
  - ▶ Coaching από Senior Engineers
  - ▶ Προσκεκλημένοι ομιλητές
- ▶ Στις μικρές επιχειρήσεις (π.χ. StartUps)
  - ▶ Γίνεται καθημερινά λόγω μικρού μεγέθους ομάδας
  - ▶ Συνήθως η ομάδα έχει χτιστεί γύρω από κοινές γνώσεις

# Υλοποίηση σε ομάδες...

22

- ▶ Καθοδήγηση των Junior Engineers από Senior Engineers
  - ▶ Coaching
  - ▶ Seminars / Training
- ▶ Διαδικασίες για version control και team working
  - ▶ Code repository
  - ▶ Version control
  - ▶ Revert changes
- ▶ Σίγουρα code reviews
- ▶ Πιθανότατα code inspections

# Version Control

23

- ▶ Ένα σύστημα που κρατάει αρχείο αλλαγών στον κώδικα
  - ▶ Εταιρικό
  - ▶ Public (π.χ. GitHub)
- ▶ Παραδείγματα:
  - ▶ Στην έκδοση 2.04 διαπιστώσαμε σημαντικά λάθη παλινδρόμησης και επιστρέφουμε στην έκδοση 2.03
  - ▶ Ο κώδικας του Νίκου θα γίνει review από τον Κώστα (την ώρα που ο Νίκος κοιμάται γιατί ο Κώστας είναι στο Los Angeles)

# Features ενός Version Control System

24

- ▶ Ποιος έκανε αλλαγές, τι αλλαγές έκανε και πότε
- ▶ Έλεγχο και διαχείριση πρόσβασης στον κώδικα
- ▶ Διαχείριση (task management), επικοινωνία με «πελάτη» και feature requests
- ▶ Συζήτηση (forum) και τεκμηρίωση κώδικα ενός έργου (wikis)
- ▶ Εργαλεία για bug tracking

# GitHub

25

- ▶ Είναι το μεγαλύτερο web-based code repository και version control system
- ▶ Έχει free και enterprise έκδοση
- ▶ Δεν είναι το μόνο!

# Φιλοσοφία του GitHub

1/2 26

- ▶ Οι χρήστες υλοποιούν κώδικα στους υπολογιστές τους (offline)
  - ▶ Push
  - ▶ Pull
- ▶ Commit
  - ▶ Τι άλλαξε
  - ▶ Ποια ήταν η προηγούμενη έκδοση
- ▶ Repository (repo)
  - ▶ Ιστορικό όλων των commit
  - ▶ Cloning for teamwork

# Φιλοσοφία του GitHub

2/2

27

- ▶ Pull = κατεβάζω στον υπολογιστή μου commits που δεν έχω
- ▶ Push = ανεβάζω τις αλλαγές που έκανα στο repository
- ▶ Branches
  - ▶ Master branch του έργου, αλλά...
  - ▶ ... και πολλά άλλα branches
  - ▶ ... που μπορούν να γίνουν merge

## Committing changes to a pull request branch created from a fork

28

- ▶ Σε κάποιο repository εσείς έχετε access
- ▶ Κάποιος έκανε ένα pull request και σας έδωσε δικαίωμα
- ▶ Δεν υπάρχουν branch restrictions που να σας εμποδίζουν να κάνετε commit
- ▶ Κάνετε fork και δημιουργείτε ένα clone του head branch
- ▶ Κάνετε αλλαγές και όταν κρίνετε ότι είναι ώριμος ο κώδικάς σας
- ▶ Κάνετε commit τις αλλαγές στο head branch του pull request (push your changes up to the original pull request)

## Υλοποίηση σε ομάδες...

29

- ▶ Καθοδήγηση των Junior Engineers από Senior Engineers
  - ▶ Coaching
  - ▶ Seminars / Training
- ▶ Διαδικασίες για version control και team working
  - ▶ Code repository
  - ▶ Version control
  - ▶ Revert changes
- ▶ Σίγουρα code reviews ←
- ▶ Πιθανότατα code inspections

## Code Reviews

1/2

30

- ▶ Θέλουμε να έχουμε καλύτερη ποιότητα κώδικα
- ▶ Και να ελαχιστοποιήσουμε το programmer's effect (π.χ. το λάθος που μπορεί να έχει κάνει ένας Junior Engineer)

Τυπικά τα code reviews είναι:

- ▶ A constructive review of a fellow developer's code.
- ▶ A required sign-off from another team member before a developer is permitted to check in changes or new code.

# Code Reviews

2/2

31

- ▶ Συμμετέχουν
  - ▶ Ο συντάκτης του κώδικα (π.χ. Junior Engineer)
  - ▶ Reviewer (π.χ. Senior Engineer, Team Leader)
- ▶ Μπορεί να είναι δια ζώσης, εξ αποστάσεως ή και offline
- ▶ Είναι iterative διαδικασία
  - ▶ Review => recoding
  - ▶ Και νέο review μέχρι να γίνει αποδεκτός ο κώδικας
- ▶ Αφορά έλεγχο κώδικα και λογικής (συνήθως σε σχέση με γνωστά ποιοτικά κριτήρια)

# Γιατί Code Reviews;

32

- ▶ Αναγκάζει τον αρχικό προγραμματιστή να προβληματιστεί για τις επιλογές του (και να κάνει καλύτερες επιλογές)
- ▶ Είναι μια σημαντική μορφή «εκπαίδευσης» (pairing a Junior Engineer με ένα Experienced Engineer)
- ▶ Οι έμπειροι μηχανικοί εμπλέκονται σε όλες τις φάσεις ανάπτυξης (π.χ. Team Leader), άρα
  - ▶ Αποκτούν καλύτερη εποπτική εικόνα του έργου
  - ▶ Μειώνει αχρείαστη επανάληψη λειτουργιών
- ▶ Εμπλέκει και τον reviewer στην ευθύνη του κώδικα που υποβλήθηκε.

# Code Reviews at Google

33

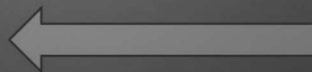
"All code that gets submitted needs to be reviewed by at least one other person, and either the code writer or the reviewer needs to have readability in that language. Most people use Mondrian to do code reviews, and obviously, we spend a good chunk of our time reviewing code."

-- Amanda Camp, Software Engineer, Google

# Υλοποίηση σε ομάδες...

34

- ▶ Καθοδήγηση των Junior Engineers από Senior Engineers
  - ▶ Coaching
  - ▶ Seminars / Training
- ▶ Διαδικασίες για version control και team working
  - ▶ Code repository
  - ▶ Version control
  - ▶ Revert changes
- ▶ Σίγουρα code reviews
- ▶ Πιθανότατα code inspections



# Code Inspections

1/3

35

- ▶ Είναι πιο «βαριά» διαδικασία από τα code reviews
- ▶ Είναι δύσκολη διαδικασία ειδικά στην αντικειμενοστραφή ανάπτυξη, γιατί...
  - ▶ ... θέματα όπως inheritance, dynamic binding, non-local effects και polymorphism κάνουν τον object oriented κώδικα μη γραμμικό και κατά συνέπεια πιο δύσκολο να επιθεωρηθεί.
- ▶ Συμμετέχουν πολλά μέλη της ομάδας ανάπτυξης
- ▶ Δεν εξετάζεται μόνο ο κώδικας, αλλά όλα τα διαγράμματα από το στατικό και δυναμικό μοντέλο.

# Code Inspections

2/3

36

- ▶ Γίνεται σε μια συνάντηση που έχει προγραμματιστεί επαρκώς (το υλικό έχει διατεθεί σε όλους από πριν)
- ▶ Έχει καθορισμένους ρόλους και καθορισμένη διαδικασία
- ▶ Ρόλοι
  - ▶ Author
  - ▶ Moderator
  - ▶ Reader (συνήθως είναι ο author)
  - ▶ Recorder
  - ▶ Inspectors



# Code Inspections

3/3

37

- ▶ Διαδικασία
  - ▶ Planning
  - ▶ Overview Meeting
  - ▶ Preparation
  - ▶ Inspection Meeting
  - ▶ Rework (refactoring)
  - ▶ Follow-up

## Σε ένα τυπικό Code Inspection

38

Θα εξεταστούν:

- ▶ Τα use cases με την αναλυτική περιγραφή των ροών
- ▶ Το sequence diagram για κάθε use case
- ▶ Το class diagram
- ▶ Ο κώδικας
- ▶ Τα test cases για κάθε περίπτωση

# Παραδείγματα Code Inspection

39

Ακολουθούν κάποια παραδείγματα ερωτήσεων που εξετάζουν οι επιθεωρητές σε ένα τυπικό code inspection.

Υπάρχουν πολλά διαθέσιμα και προφανώς κάθε επιχείρηση καθορίζει τα δικά της standards και έχει τις δικές τις λίστες ελέγχου.

# Specification and design

40

Παράδειγμα ερωτήσεων  
από ένα code inspection

- ▶ Has all functionality specified been implemented?
- ▶ Is there any excess functionality not specified?
- ▶ Does all of the specified functionality execute correctly?

# Initialization and declarations

41

Παράδειγμα ερωτήσεων  
από ένα code inspection

- ▶ Are all variables properly initialized before use?
- ▶ Are all variables and class members of the proper type and mode?
- ▶ Are all variables declared in the proper scope?
- ▶ Is a constructor called when a new object is required?

# Algorithms

42

Παράδειγμα ερωτήσεων  
από ένα code inspection

- ▶ Are the order of operations, operator precedence, and parentheses correct?
- ▶ Is division by zero possible?
- ▶ Is there any integer division subject to improper rounding or truncation?
- ▶ Is there any implicit type conversion?
- ▶ Are comparison and Boolean operators correct?