10β Αναδιάρθρωση κώδικα (Code Refactoring)

Τεχνολογία Λογισμικού

Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών
Εθνικό Μετσόβιο Πολυτεχνείο

Χειμερινό εξάμηνο 2017-18

Δρ. Κώστας Σαΐδης (saiko@di.uoa.gr)

Refactoring

Η αναδιάρθρωση του κώδικα για την επίτευξη καλύτερου εσωτερικού σχεδιασμού, χωρίς να αλλάζει η λειτουργικότητά του.

Βελτίωση των μη λειτουργικών χαρακτηριστικών

- Απλότητα
- Αναγνωσιμότητα
- Επεκτασιμότητα
- Διαχειρισιμότητα / συντηρισιμότητα

Ωρίμανση του λογισμικού

- Μετασχηματισμός του υφιστάμενου κώδικα
- Καλύτερες αφαιρέσεις (πιο απλές, πιο εκφραστικές -> πιο αποτελεσματικές)
- Καλύτερα μοντέλα αντικειμένων (object models, π.χ. classes ή interfaces)
- Καλύτερη εσωτερική αρχιτεκτονική (πιο αποτελεσματικός διαχωρισμός ενδιαφερόντων, πιο χαλαρή σύνδεση συστατικών, κ.λπ)

Όχι μονομιάς

- Απλά, μικρά micro-refactorings
- Που βελτιώνουν σιγά-σιγά και τμηματικά τον κώδικα
- Χωρίς να αλλάζουν τη λειτουργικότητά του

Αυτοματισμοί

- Πολλά IDEs παρέχουν αυτοματοποιημένες refactoring διαδικασίες (τα micro-refactorings που λέγαμε προηγουμένως)
- Έυκολη, γρήγορη και χωρίς λάθη εφαρμογή σε μεγάλο κώδικα
- Θα τα δούμε στο παράδειγμα

Πώς καταλαβαίνω ότι ο κώδικάς μου χρειάζεται refactoring;

Οσμές (code smells)

- Αρνητικά συμπτώματα/επιφανειακές ενδείξεις στον κώδικα που μπορεί να οφείλονται κάποιο βαθύτερο πρόβλημα.
- Ποιοτικές "αντενδείξεις" που υποκρύπτουν την παραβίαση μιας βασικής αρχής αφαίρεσης ή κάποιας καλής σχεδιαστικής πρακτικής

Τα code smells δεν είναι bugs

Παραδείγματα

- Επαναλαμβανόμενος κώδικας
- Μεγάλες κλάσεις
- Μεγάλες μέθοδοι
- Πολλές παράμετροι σε μια μέθοδο
- Παραβίαση της δυνατότητας αντικατάστασης (σε υποτύπους)
- κ.ά

Code smells & τεχνικό χρέος

Τα code smells μπορεί να υποδηλώνουν τους παράγοντες που αυξάνουν το τεχνικό χρέος.

Ας θυμηθούμε από την διάλεξη 3.

Τεχνικό χρέος (technical debt)

Το κόστος της πρόσθετης δουλειάς που θα απαιτηθεί από την επιλογή μιας εύκολης και γρήγορης υλοποίησης αντί για την εφαρμογή της συνολικά καλύτερης λύσης.

Refactoring & τεχνικό χρέος

Το refactoring μειώνει το τεχνικό χρέος (μειώνει το κόστος συντήρησης και επέκτασης του λογισμικού).

Τεχνικές refactoring

- Move method or field
- Pull up (move to superclass) - Pull down (move to subclass)
- Rename method or field
- Extract class (move part of existing code into a new class)
- Extract method (introduce a new method to hold a part of existing code)
- Encapsulate field (getters/setters)
- Generalized types (generics - polymorphism)
- κ.ά

Παράδειγμα

Refactoring in practice

Βέλτιστη πρακτική

- Αυτόματα Unit tests
- Πολλές, μικρές, συχνές σχεδιαστικές επαναλήψεις (iterations)
- Μικρά refactorings
- Χρήση IDE για επιτάχυνση της διαδικασίας εφαρμογής τους

Refactoring: μέρος της διαδικασίας σχεδιασμού και ανάπτυξης του λογισμικού κατά την Agile μεθοδολογία