

Τεχνολογία λογισμικού στην πράξη Μοντέλα και μέθοδοι τεχνολογίας λογισμικού

Διομήδης Σπινέλλης
Τμήμα Διοικητικής Επιστήμης και Τεχνολογίας
Οικονομικό Πανεπιστήμιο Αθηνών

dds@aueb.gr
<http://www.dmst.aueb.gr/dds>
@CoolSWEng

2021-03-09

Άσκηση (Μοντέλα και μέθοδοι τεχνολογίας λογισμικού)

Για ένα δημοφιλές-σημαντικό έργο ανοιχτού λογισμικού απαντήστε τα ακόλουθα.

- Τι είδους μοντέλο τεχνολογίας λογισμικού θα ταίριαζε περισσότερο για τη μοντελοποίησή του;
- Ποια μέθοδος τεχνολογίας λογισμικού επιλέχθηκε για την ανάπτυξή του; Μπορείτε να αντλήσετε στοιχεία από τον πηγαίο κώδικα και την τεκμηρίωση του έργου.
- Αν το σύστημα άρχιζε να αναπτύσσεται τώρα ποια μέθοδος πιστεύετε ότι θα έπρεπε να επιλεγεί;

Τι προσφέρουν τα μοντέλα και οι μέθοδοι;

- Συστηματοποίηση
- Επαναληψιμότητα
- Μοντέλα
- Τρόπο επίλυσης προβλημάτων
- Συμβολισμούς
- Διαδικασίες κατασκευής και ανάλυσης
- Μέθοδοι συστηματοποιούν
- Προδιαγραφές
- Σχεδιασμό
- Κατασκευή
- Δοκιμές

Επισκόπηση

- Μοντελοποίηση
- Είδη μοντέλων
- Ανάλυση μοντέλων
- Μέθοδοι τεχνολογίας λογισμικού

Αρχές μοντελοποίησης

- Μοντελοποιούμε τα σημαντικά
- Προσφέρουμε οπτικές
- Επιτρέπουμε την επικοινωνία μεταξύ μερών

«Κάθε μοντέλο είναι μια αφαίρεση και κατά συνέπεια όχι πλήρης παράσταση της πραγματικότητας»

Έκφραση μοντέλων

- Ένα μοντέλο απαρτίζεται από
- Οντότητες
- Σχέσεις
- Μπορεί να εκφραστεί
- Γραφικά
- Με μορφή κειμένου
- Ορισμός του τρόπου έκφρασης
- Για γραφική μορφή: μεταμοντέλο
- Για κείμενο: BNF

Ιδιότητες μοντέλων

- Πληρότητα (completeness)
- Συνέπεια (consistency)
- Ορθότητα (correctness)

(Οι δύο πρώτες ιδιότητες μπορούν να ελεγχθούν αυτόματα.)

Σχεδιασμός μέσω Συμβολαίου

Ένα μοντέλο μπορεί να ορίζει:

- Προϋποθέσεις (preconditions)
- Συνθήκες που πρέπει να ισχύουν πριν την εκτέλεση κάποιου τμήματος κώδικα (π.χ. μεθόδου).
- Μετασυνθήκες (postconditions)
- Συνθήκες που ξέρουμε πως θα ισχύουν μετά την εκτέλεση κάποιου τμήματος κώδικα.
- Αναλλοίωτα χαρακτηριστικά

Είδη μοντέλων

- Μοντέλα πληροφοριών, π.χ.
- ER
- Συσχετίσεις UML
- Μοντέλα δομής, π.χ.

- Διάγραμμα κλάσεων (class diagram) και σχέσεων
- Διάγραμμα αντικειμένων (object diagram)
- Διάγραμμα εξαρτημάτων (component diagram)
- Διάγραμμα ανάπτυξης (deployment diagram)
- Μοντέλα συμπεριφοράς, π.χ.
 - Διάγραμμα καταστάσεων (statechart diagram)
 - Διάγραμμα δραστηριοτήτων (activity diagram)
 - Διάγραμμα ακολουθίας (sequence diagram)
 - Διάγραμμα συνεργασίας (collaboration diagram)

Ανάλυση μοντέλων

- Πληρότητα
- Συνέπεια
- Ορθότητα
- Ιχνηλασιμότητα
- Αλληλεπιδράσεις

Ευρετικές μέθοδοι τεχνολογίας λογισμικού

- Δομημένη ανάλυση και σχεδιασμός
- Μέθοδοι μοντελοποίησης δεδομένων
- Αντικεμενοστρεφής ανάλυση και σχεδιασμός

Τυπικές μέθοδοι τεχνολογίας λογισμικού

- Γλώσσες προδιαγραφών
- Τυπικός έλεγχος (formal verification)
- Λογική τεκμηρίωση (logical inference)

Πρωτότυπα

- Είδη
- Σε χαρτί
- Εκτελέσιμη προδιαγραφή
- Κώδικας που θα πεταχτεί
- Κώδικας που θα εξελιχθεί
- Στόχος
- Προδιαγραφές
- Αρχιτεκτονική
- Αλγόριθμος
- UI

Ευέλικτες μέθοδοι

- Rapid Application Development (RAD)
- Ακραίος προγραμματισμός (XP)
- Scrum
- Feature-Driven Development (FDD)

Μανιφέστο για την ευέλικτη ανάπτυξη λογισμικού

Ανακαλύπτουμε καλύτερους τρόπους ανάπτυξης λογισμικού στην πράξη και βοηθάμε τους άλλους να κάνουν το ίδιο. Αυτή η δραστηριότητα μας έχει οδηγήσει στο να αξιολογούμε:

- Τα άτομα και τις αλληλεπιδράσεις πάνω από τις διαδικασίες και τα εργαλεία
- Το λογισμικό που λειτουργεί πάνω από την εκτενή τεκμηρίωση
- Την συνεργασία με τον πελάτη πάνω από τις συμβατικές διαπραγματεύσεις
- Την ανταπόκριση στην αλλαγή πάνω από την τήρηση ενός προδιαγεγραμμένου σχεδίου

Χαρακτηριστικά του ακραίου προγραμματισμού

- Ο προγραμματισμός σε ζευγάρια
- Η συγγραφή ελέγχων μονάδος πριν από τον κώδικα.
- Η συνεχής ολοκλήρωση και ο συνεχής έλεγχος του κώδικα (πολλές φορές μέσα στην ημέρα).
- Η βαθμιαία εξέλιξη του σχεδίου του έργου.
- Η γρήγορη παραγωγική χρήση του συστήματος για την εκμείωση των απαιτήσεων με το μεγαλύτερο όφελος για τον πελάτη.

Χαρακτηριστικά της Scrum

- Scrum master διοικεί τις ενέργειες
- Sprint < 30 ημέρες
- Product Backlog Item
- Προϋπολογισμός
- Επιλογή
- Καθημερινές συναντήσεις

Διεργασία FDD

(Feature-Driven Development)

- Μοντέλο του προϊόντος
- Κατάλογος χαρακτηριστικών
- Πλάνο υλοποίησης χαρακτηριστικών

- Σχέδιο ανά κύκλο
- Ανά χαρακτηριστικό:
- Υλοποίηση
- Έλεγχος
- Ολοκλήρωση

Προετοιμασία για το επόμενο μάθημα (1)

- Διαβάστε το κεφάλαιο 10 του SWEBOOK v 3.0
- Άσκηση (Ποιότητα λογισμικού)

Για ένα δημοφιλές-σημαντικό έργο ανοιχτού λογισμικού απαντήστε τα ακόλουθα.

- Πώς γίνεται έλεγχος της ποιότητας λογισμικού που αναπτύσσεται;
- Ποιοι παράγοντες επίδρασης και ποια χαρακτηριστικά ποιότητας είναι σημαντικότερα για τις απαιτήσεις ποιότητας λογισμικού;
- Πώς μετράται η ποιότητα του λογισμικού που αναπτύσσεται;

Μπορείτε να αντλήσετε στοιχεία από τον πηγαίο κώδικα και την τεκμηρίωση του έργου. Αν δε βρίσκετε στοιχεία για κάποια από τα παραπάνω αυτό ίσως σημαίνει ότι δεν έγιναν στα πλαίσια του έργου που μελετάτε. Πως πιστεύετε ότι θα έπρεπε να γίνουν;

Προετοιμασία για το επόμενο μάθημα (2)

- Βίντεο (Ποιότητα λογισμικού: Διαβεβαίωση ποιότητας λογισμικού)
https://www.youtube.com/watch?v=hMfPCdF07hA&list=PLZTe0pWS8OYv3KYWJTZT_3chbtpXBoiOz

Άδεια διανομής

Εκτός αν αναφέρεται κάτι διαφορετικό, όλο το πρωτότυπο υλικό της σελίδας αυτής του οποίου δημιουργός είναι ο Διομήδης Σπινέλλης παρέχεται σύμφωνα με τους όρους της άδειας Creative Commons Αναφορά-Παρόμοια διανομή 3.0 Ελλάδα.

