

Τεχνολογία λογισμικού στην πράξη

Απαιτήσεις λογισμικού

Διομήδης Σπινέλλης
Τμήμα Διοικητικής Επιστήμης και Τεχνολογίας
Οικονομικό Πανεπιστήμιο Αθηνών

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

2023-03-13

Παρουσιάσεις ομάδων

- Εξαγωγή απαιτήσεων για ένα σημαντικό-δημοφιλές έργο ανοιχτού λογισμικού.
- Συγκεκριμένοι στόχοι:
- Αναγνώριση ενδιαφερόμενων μερών
- Αναγνώριση λειτουργικών και μη λειτουργικών απαιτήσεων
- Μελέτη συνεπειών αλλαγής απαιτήσεων
- Ομάδες
- `/(^_^)\(^_^)(^_^) (^_^)(^_^)(^_^)`
- CaffeineOverflow
- Codebusters
- Error 404

Ιδιότητες των απαιτήσεων

- Επαληθεύσιμες
- Προτεραιότητα
- Κατάσταση
- Αναγνωριστικό

Διαχωρισμός απαιτήσεων

- Άξονας:
- Απαιτήσεις από το προϊόν
- Απαιτήσεις από τη διεργασία ανάπτυξης
- Άξονας:
- Λειτουργικές απαιτήσεις
- Μη λειτουργικές απαιτήσεις
- Άξονας:

- Απαιτήσεις συστήματος
- Απαιτήσεις λογισμικού
- Αναδυόμενες απαιτήσεις (από όλο το σύστημα)

Η διεργασία εκμείωσης απαιτήσεων

- Συνεχίζεται σε όλο τον κύκλο ζωής του λογισμικού
- Θέτει τις απαιτήσεις κάτω από έλεγχο σχηματισμών
- Προσαρμόζεται στο λογισμικό και το προϊόν

Ποιοι έχουν λόγο στις απαιτήσεις;

- Χρήστες
- Πελάτες
- Αναλυτές της αγοράς
- Ρυθμιστικές αρχές
- Μηχανικοί λογισμικού (πολύ σημαντικό)

Πηγές απαιτήσεων

- Οι στόχοι του έργου
- Γνώση της περιοχής (domain knowledge)
- Οι έχοντες λόγο
- Επιχειρησιακοί κανόνες (business rules)
- Επιχειρησιακό περιβάλλον (π.χ. μη λειτουργικές προδιαγραφές)
- Οργανωσιακό περιβάλλον

Τρόποι εκμείωσης απαιτήσεων

- Συνεντεύξεις
- Σενάρια
- Πρότυπα (prototypes)
- Οργανωμένες συναντήσεις (facilitated meeting)
- Παρατήρηση
- Ιστορίες χρηστών
- Άλλοι
- Ανάλυση του ανταγωνισμού
- Βάσεις λαθών, τηλεμετρία
- Σχόλια χρηστών
- Ανάλυση πηγών γνώσης του πεδίου

Γιατί να αναλύσουμε τις προδιαγραφές

- Για να εντοπίσουμε συγκρουόμενες

- Για να δούμε τα όρια του λογισμικού και την αλληλεπίδρασή του με το περιβάλλον
- Για να αναπτύξουμε προδιαγραφές συστήματος σε προδιαγραφές λογισμικού

Παράδειγμα καθορισμού ορίων

Google Public DNS is not, however, any of the following:

- A top-level domain (TLD) name service.
- A DNS hosting or failover service.
- An authoritative name service.
- A malware-blocking service

Ανάλυση: κατηγοριοποίηση

- Λειτουργική / μη λειτουργική
- Προέλευση
- Από προδιαγραφή υψηλού επιπέδου
- Από αναδυόμενη απαίτηση
- Από άμεση πηγή
- Διεργασία ή προϊόν
- Εύρος
- Σταθερότητα ή αστάθεια
- (γιατί είναι σημαντικό;)

Ανάλυση: αφηρημένη μοντελοποίηση

- Π.χ. UML
- Case diagram
- Data flow model
- State model
- Object model
- Data model
- ...
- Επιλογή μοντέλου
- Είδος προβλήματος
- Εξειδίκευση του μηχανικού ανάπτυξης
- Απαιτήσεις από τη διεργασία ανάπτυξης

Ανάλυση απαιτήσεων: άλλα στοιχεία

- Αρχιτεκτονικός σχεδιασμός
- Κατανομή απαιτήσεων
- Διαπραγμάτευση απαιτήσεων (και ιεράρχηση)

- Τυπική ανάλυση

Καθορισμός προδιαγραφών

- Καθορισμός του συστήματος
- Καθορισμός προδιαγραφών συστήματος
- Καθορισμός προδιαγραφών λογισμικού

Επικύρωση προδιαγραφών

- Επιθεώρηση
- Υλοποίηση πρωτοτύπου
- Επικύρωση του μοντέλου
- Έλεγχοι αποδοχής

Πρακτικά θέματα

- Η διεργασία προδιαγραφής των απαιτήσεων είναι επαναληπτική
- Οι απαιτήσεις πρέπει είναι υπό έλεγχο αλλαγών
- Το σημαντικότερο στοιχείο μιας απαίτησης είναι το προσδιοριστικό της
- Πρέπει να τεκμηριώνεται πώς έχει προέλθει μια απαίτηση

Προετοιμασία για το επόμενο μάθημα (1)

- Διαβάστε το κεφάλαιο 2 του SWEBOK v 3.0
- Άσκηση (Αρχιτεκτονική)
- Επιλέξτε ένα (σημαντικό-δημοφιλές) έργο ανοιχτού λογισμικού και εξετάστε στοιχεία του σχεδιασμού του:
- αναζητήστε αρχές καλής σχεδίασης λογισμικού
- αναγνωρίστε θέματα - παράγοντες σχεδίασης που αποκτούν ιδιαίτερη βαρύτητα στο συγκεκριμένο έργο
- επιλέξτε μια στρατηγική σχεδίασης και ταιριαστή σημειογραφία που θεωρείτε κατάλληλες για να αναπαραστήσετε μια στατική όψη του σχεδίου λογισμικού. Μπορείτε να χρησιμοποιήσετε αφαίρεση, μία από τις αρχές καλής σχεδίασης λογισμικού, για να επικοινωνήσετε εκείνες τις οντότητες που θεωρείτε πιο σημαντικές.

Προετοιμασία για το επόμενο μάθημα (2)

- Βίντεο (Σχεδιαστικά πρότυπα): https://www.youtube.com/watch?v=vNHpsC5ng_E

Συμβουλές για την άσκηση:

- Συμβουλευτείτε την τεκμηρίωση και τον πηγαίο κώδικα του έργου (για έργα ανοιχτού λογισμικού στο Github μπορείτε να κατεβάσετε τον πηγαίο

τους κώδικα σε μορφή zip).

- Συνδυάστε πληροφορίες υψηλού επιπέδου (τι προσφέρει το σύστημα στους χρήστες) με δομικά στοιχεία (πακέτα, συστατικά) για να πετύχετε το σκοπό σας.
- Επιλέξτε έργα που χρησιμοποιούν οικείες για εσάς τεχνολογίες (π.χ. Java).

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.

