Εργασία στο Μάθημα της Τεχνολογίας Λογισμικού  
Παραδοτέα και διαδικασία εξέτασης

Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών  
Εθνικό Μετσόβιο Πολυτεχνείο

Χειμερινό εξάμηνο 2019-2020

Διδάσκοντες: Ν. Παπασπύρου, Β. Βεσκούκης, Κ. Σαΐδης

# Παραδοτέα

## Δομή φακέλων στο Git repository της ομάδας

Η δομή των φακέλων στο git repository της κάθε ομάδας θα πρέπει να ακολουθεί την εξής δομή:

* Φάκελος **back-end**: περιέχει τον κώδικα της back-end εφαρμογής.
* Φάκελος **cli-client**: περιέχει τον κώδικα της cli εφαρμογής.
* Φάκελος **documentation**: περιέχει την τεκμηρίωση της εργασίας (συνολικά).
* Φάκελος **front-end**: περιέχει τον κώδικα της front-end εφαρμογής (για 4μελείς ομάδες).
* Φάκελος **mobile-app**: περιέχει τον κώδικά της mobile εφαρμογής (για 5μελείς ομάδες που θα το επιλέξουν).

Σε περίπτωση που το κρίνετε απαραίτητο, μπορείτε να προσθέσετε νέους φακέλους στην παραπάνω δομή (για παράδειγμα, λόγω δημιουργίας κάποιου component/module που (επανα)χρησιμοποιείται σε δύο ή περισσότερες εφαρμογές).

## Λίστα παραδοτέων

Η λίστα των παραδοτέων του μαθήματος φαίνεται στον παρακάτω πίνακα. Τα παραδοτέα θα πρέπει να καταχωριστούν υποχρεωτικά στον αντίστοιχο φάκελο του github repository κάθε ομάδας, με ημ/νία καταχώρισης (commit date) προγενέστερη της προθεσμίας. Δεν θα υπάρξει άλλη διαδικασία υποβολής της εργασίας.

| **Παραδοτέο** | **Φάκελος**  **github** | **Ομάδα 3 ατόμων** | **Ομάδα 4 ατόμων** | **Ομάδα 5 ατόμων** |
| --- | --- | --- | --- | --- |
| Documentation - Diagrams |  |  |  |  |
| Εγγραφο StRS - Stakeholders Requirements Specification | documentation | NAI | | |
| Εγγραφο SRS - Software Requirements Specification | 2 Use Cases | 3 Use Cases | 3 Use Cases |
| Διαγράμματα UML Activity | αντίστοιχα με τα Use Cases | | |
| Διαγράμματα UML Sequence | αντίστοιχα με τα Use Cases | | |
| Διαγράμματα UML Deployment | NAI | | |
| Διαγράμματα UML Component | ΝΑΙ | | |
| Διαγράμματα UML Class | NAI | | |
| Source code - implementation |  |  |  |  |
| Πηγαίος κώδικας εφαρμογής για εισαγωγή, διαχείριση και πρόσβαση σε πρωτογενή δεδομένα (backend) | back-end | NAI | | |
| RESTful API | back-end | NAI | | |
| Command line interface (CLI) | cli-client | NAI | | |
| Εκτελέσιμη μορφή (build & deploy your code from source) |  | back-end, cli-client | back-end, cli-client, front-end | back-end, cli-client, front-end, mobile-app |
| Back-end unit tests | back-end | ΝΑΙ | | |
| CLI unit tests | cli-client | ΝΑΙ | | |
| Back-end functional tests | back-end | ΝΑΙ | | |
| CLI functional tests | cli-client | ΝΑΙ | | |
| Front-end παρουσίασης δεδομένων σε περιβάλλον web | front-end |  | ΝΑΙ | ΝΑΙ |
| Front-end tests | front-end |  | ΠΡΟΑΙΡΕΤΙΚΑ | ΠΡΟΑΙΡΕΤΙΚΑ |
| Mobile εφαρμογή παρουσίασης δεδομένων σε smartphone | mobile-app |  |  | ΝΑΙ (\*) |
| Mobile app tests | mobile-app |  |  | ΠΡΟΑΙΡΕΤΙΚΑ |
| Λειτουργίες χρέωσης του REST API | back-end, cli-client, front-end |  |  | ΝΑΙ (\*) |
| Ανάλυση επιδόσεων του REST API ("benchmarking") | back-end, cli-client, front-end |  |  | ΝΑΙ (\*) |

(\*) Ένα από τα τρία

Όλα τα σενάρια ελέγχων (unit and functional tests) θα πρέπει να εκτελούνται αυτοματοποιημένα από το build εργαλείο σας.

## Μορφότυποι Παραδοτέων

Η μορφή των παραδοτέων φαίνεται στον παρακάτω πίνακα.

|  |  |  |
| --- | --- | --- |
| Παραδοτέο | Μορφότυπος | Παρατηρήσεις |
| Documentation - Diagrams |  |  |
| Έγγραφα StRS & SRS | docx, odt ή άλλη ανοικτή **επεξεργάσιμη** μορφή, σύμφωνα με το πρότυπο που διατέθηκε |  |
| Διαγράμματα UML | Αρχείο Visual Paradigm (vpp) | Θα πρέπει να παραδοθεί **\*ένα\*** αρχείο visual paradigm (vpp) με όλα τα διαγράμματα. Θα πρέπει να υπάρχει κατάλληλη ονομασία των διαγραμμάτων ώστε να είναι εύκολα ανιχνεύσιμη η ένταξή τους στα κείμενα, όπου γίνεται.  Επίσης, στα διαγράμματα είναι επιθυμητές αναφορές (παραπομπές) στα αντίστοιχα συστατικά στοιχεία πηγαίου κώδικα, όπου αυτό έχει νόημα, καθώς και συνδέσεις μεταξύ τους. |
| Source code - implementation |  |  |
| Πηγαίος κώδικας για το σύνολο των λειτουργιών που υλοποιήσατε | Το σύνολο των πηγαίων αρχείων σύμφωνα με το περιβάλλον, εργαλεία κλπ που θα χρησιμοποιήσετε. | Θα πρέπει να περιλάβετε στο αρχείο README σε κάθε επιμέρους εφαρμογής, παρέχοντας μια συνοπτική περιγραφή των συστατικών της εφαρμογής και των βημάτων που απαιτούνται για το στήσιμο του περιβάλλοντος ανάπτυξης. |

# Προθεσμίες

Το πρώτο σκέλος των παραδοτέων (τα έγγραφα και διαγράμματα τεκμηρίωσης) θα υποβληθεί δύο φορές: η πρώτη έκδοση θα παραδοθεί πριν την έναρξη της εξεταστικής και η δεύτερη -και οριστική- έκδοση θα υποβληθεί μετά το πέρας αυτής.

Το δεύτερο σκέλος των παραδοτέων (πηγαίος κώδικας, υλοποίηση) θα υποβληθεί άπαξ μετά την οριστική υποβολή των παραδοτέων τεκμηρίωσης.

Για όλες τις προθεσμίες, λαμβάνεται υπόψη η ημερομηνία καταχώρησης (commit) στο github repository της ομάδας σας.

## 1η έκδοση τεκμηρίωσης

Η πρώτη έκδοση των εγγράφων και διαγραμμάτων τεκμηρίωσης (documents & diagrams) θα πρέπει να παραδοθεί το αργότερο **έως τα μεσάνυκτα της Κυριακής 2 Φεβρουαρίου 2020** (με βάση, πάντα, την ημερομηνία καταχώρισης των αρχείων στο git repository της ομάδας σας). Σε ότι αφορά τα διαγράμματα, δεν απαιτείται πρόσθετη περιγραφή, παρά μόνο το κάθε διάγραμμα αυτό καθ’ αυτό (δηλαδή, μπορείτε να καταχωρίσετε τα διαγράμματά σας στο git απευθείας ως αρχεία εικόνας).

## 2η έκδοση τεκμηρίωσης (τελική)

Η τελική έκδοση των εγγράφων και διαγραμμάτων τεκμηρίωσης (documents & diagrams) θα πρέπει να παραδοθεί το αργότερο **έως τα μεσάνυκτα της Πέμπτης 27 Φεβρουαρίου 2020.** Μπορείτε να υποβάλλετε και τροποποιημένες εκδοχές των εγγράφων και των διαγραμμάτων, δεδομένου ότι έχετε κάνει σημαντικές τροποποιήσεις σε σχέση με την πρώτη τους έκδοση, καταγράφοντας ρητά σε ένα ξεχωριστό αρχείο **documentation/updates.md** μια ευσύνοπτη περιγραφή τους.

## Πηγαίος κώδικας – εξέταση εργασίας

Ο πηγαίος κώδικας της εργασίας πρέπει να έχει θα παραδοθεί στις 09:00 της ημέρας προφορικής εξέτασης, **για όλες τις ομάδες ανεξάρτητα από την ώρα εξέτασης κάθε ομάδας**. Το ακριβές πρόγραμμα εξέτασης των ομάδων θα ανακοινωθεί εντός του Φεβρουαρίου και **η διαδικασία θα λάβει χώρα στο διάστημα μεταξύ Παρασκευής 28 Φεβρουαρίου 2020 και Παρασκευής 6 Μαρτίου 2020 μεταξύ των ωρών 09:00 – 17:00**.Η ένταξη των ομάδων στα διαθέσιμα χρονοπαράθυρα (διάρκειας 30’) θα γίνει με τυχαίο τρόπο και δεν θα είναι δυνατό να αλλάξει

# Χρόνος, διάρκεια και διαδικασία εξέτασης

Κάθε ομαδική εξέταση θα διαρκεί 30’ (αυστηρά). Η παρουσία όλων των μελών της ομάδας είναι υποχρεωτική στην εξέταση (σπουδαστές που δεν θα εμφανιστούν στην εξέταση δε θα λάβουν βαθμό στην εργασία).

Η διαδικασία θα είναι η εξής:

* Κάθε ομάδα προσέρχεται με το δικό της laptop, στο οποίο έχουν προ-εγκατασταθεί και ήδη τεθεί σε λειτουργία όλα τα απαραίτητα για τη λειτουργία συστατικά της εφαρμογής (τα οποία χρησιμοποιούνται ως έχουν, δηλαδή δεν έχει γραφεί ο κώδικάς τους στο πλαίσιο της εργασίας), όπως είναι, για παράδειγμα, η βάση δεδομένων.
* Στα πρώτα 5’ :
  + Γίνεται checkout του κώδικα από το git repository της ομάδας.
  + Γίνεται εκτέλεση του build του κώδικα καθώς και των functional tests αυτού, τα οποία θα πρέπει να επιτυγχάνουν χωρίς σφάλματα.
  + Παράγονται τα application artifacts, γίνεται το deployment αυτών και εκκίνηση της εφαρμογής.
* Στα επόμενα 20’:
  + Γίνεται παρουσίαση (demonstration) των λειτουργιών της εφαρμογής.
* Στα τελευταία 5’:
  + Γίνεται εκτέλεση των αυτοματοποιημένων ελέγχων στο REST API της εφαρμογής και της batch import λειτουργίας της.

Στις αρχές Φεβρουαρίου θα διατεθεί και μια κοινή σουίτα ελέγχων (σενάρια ελέγχου σε κοινό data set) για να σας βοηθήσει να δοκιμάσετε τα σενάρια ελέγχου που θα εκτελεστούν κατά την εξέταση.