

Team Plan

ECLIPSE™

Patras 2022-23

Editor: Σάββας Γεράσιμος 1072475

Contributor: Καπογιάννης Κωσταντίνος 1072521

Reviewers: Μπέσιας Σπυρίδων 1072524

Σερταρίδης Ηλίας 1072480

Μέλη ομάδας:

Καπογιάννης Κωνσταντίνος 1072521 4^ο έτος

Μπέσιας Σπυρίδων 1072524 4^ο έτος

Σάββας Γεράσιμος 1072475 4^ο έτος

Σερταρίδης Ηλίας 1072480 4^ο έτος

Λαμβάνοντας υπόψη τις μεθοδολογίες Scrum και Kanban συνθέσαμε μια δική μας μεθοδολογία που στηρίζεται στη μέθοδο Kanban. Συγκεκριμένα θα υπάρχουν διάφορα υπο-έργα τα οποία θα επιλέγονται για υλοποίηση από τα μέλη της ομάδας μας, ανάλογα με βάση τα παραδοτέα. Ταυτόχρονα θα εργάζονται δυάδες σε διαφορετικά κομμάτια ενός παραδοτέου με editor, contributor και τα υπόλοιπα δύο μέλη θα έχουν τον ρόλο των reviewers. Μάλιστα η ομάδα μας θα λειτουργεί με βάση την agile, καθώς κατά την διάρκεια υλοποίησης ενός υπο-έργου μπορούν να προκύψουν νέες ιδέες, προβληματισμοί και συνεπώς να χρειαστεί να ανατρέξουμε σε προηγούμενα ολοκληρωμένα ή παράλληλα υπο-έργα για διορθώσεις ή προσθήκες. Θα υπάρχουν ορισμένες τροποποιήσεις όπως για παράδειγμα τα υποχρεωτικά έργα για τα οποία δεν θα υφίσταται καθυστέρηση και αυτά θα είναι οι αναφορές για τα παραδοτέα. Επίσης θα λαμβάνουν χώρα κάθε εβδομάδα ορισμένα meeting και θα γίνονται αξιολογήσεις ανά τακτά χρονικά διαστήματα.

Η γλώσσα υλοποίησης που θα χρησιμοποιήσουμε για το back-end και για τα γραφικά είναι η Java, για τη βάση δεδομένων η SQL και για τα τεχνικά κείμενα το Word.

Στα εργαλεία που χρησιμοποιούμε συγκαταλέγονται:

Για Pert chart χρησιμοποιούμε το Miro

<https://miro.com/app/dashboard/>

Για Gantt chart χρησιμοποιούμε το Monday

<https://support.monday.com/hc/en-us/articles/360015643840-The-Gantt-Chart-View-and-Widget>

Για Sql χρησιμοποιούμε το Mysql

<https://www.mysql.com/>

Για Java-Back-end χρησιμοποιούμε το IntelliJ

<https://www.jetbrains.com/idea/>

Για Mockup/Design χρησιμοποιούμε το Figma

<https://www.figma.com/>

Για το διάγραμμα ανάθεσης δυναμικού χρησιμοποιούμε το draw.io

<https://app.diagrams.net/>

Παρακάτω φαίνονται αναλυτικά όλα τα υπο-έργα και οι διάφορες φάσεις του project του εξαμήνου στις οποίες πρέπει να υλοποιηθούν, καθώς και ο κανονικός χρόνος διάρκειας του κάθε υπο-έργου, λάβαμε υπόψη τον κάθε μήνα με βάση την περίοδο που θα υλοποιηθεί το project του μαθήματος και λάβαμε υπόψη μας τα σαββατοκύριακα(αρχικά λάβαμε υπόψη και τις αργίες αλλά λόγω περιορισμών των διαθέσιμων προγραμμάτων ήταν αδύνατο). Όσον αφορά το κρίσιμο μονοπάτι σε ορισμένες περιπτώσεις που κάποια υπο-έργα ολοκληρώνονταν την ίδια μέρα, θεωρήθηκε κρίσιμο μονοπάτι αυτό που οδηγούσε σε ένα υπο-έργο που είναι προαπαιτούμενο για τα παραδοτέα:

1. Deliverables 1:

- TY1-Project-description-V0.1 (4 εργάσιμες μέρες).
- TY2-Team-plan-V0.1 (3 εργάσιμες μέρες).
- TY3-Project-plan-V0.1 (3 εργάσιμες μέρες).
- TY4-Risk-assesment-V0.1 (1 εργάσιμες μέρες).

2. Deliverables 2:

- TY5-Use-cases-V0.1 (6 εργάσιμες μέρες).
- TY6-Domain-model-V0.1 (3 εργάσιμες μέρες).

3. Deliverables 3:

- TY7-Robustness-diagrams-V0.1 (6 εργάσιμες μέρες).
- TY8-Domain-model-V0.2 (3 εργάσιμες μέρες).
- TY9-Use-cases-V0.2 (3 εργάσιμες μέρες).

4. Deliverables 4:

- TY10-Sequence-diagrams-V0.1 (4 εργάσιμες μέρες).
- TY11-GUI-Design (3 εργάσιμες μέρες).
- TY12- Domain-model-V0.3 (3 εργάσιμες μέρες).
- TY13-Back-end-Design (3 εργάσιμες μέρες).
- TY14-Database-Design (3 εργάσιμες μέρες).

5. Deliverables 5:

- TY15-Class-diagrams-V0.1 (3 εργάσιμες μέρες).
- TY16-Database-Implementation (5 εργάσιμες μέρες).
- TY17-Back-end-Implementation (10 εργάσιμες μέρες).
- TY18-GUI-Implementation (10 εργάσιμες μέρες).
- TY19-Test-Cases-V0.1 (5 εργάσιμες μέρες).

6. Deliverables 6:

- TY20-Final-Versions (8 εργάσιμες μέρες).
-

Κατόπιν φαίνεται η ονομασία του κάθε υπο-έργου που αναφέραμε παραπάνω:

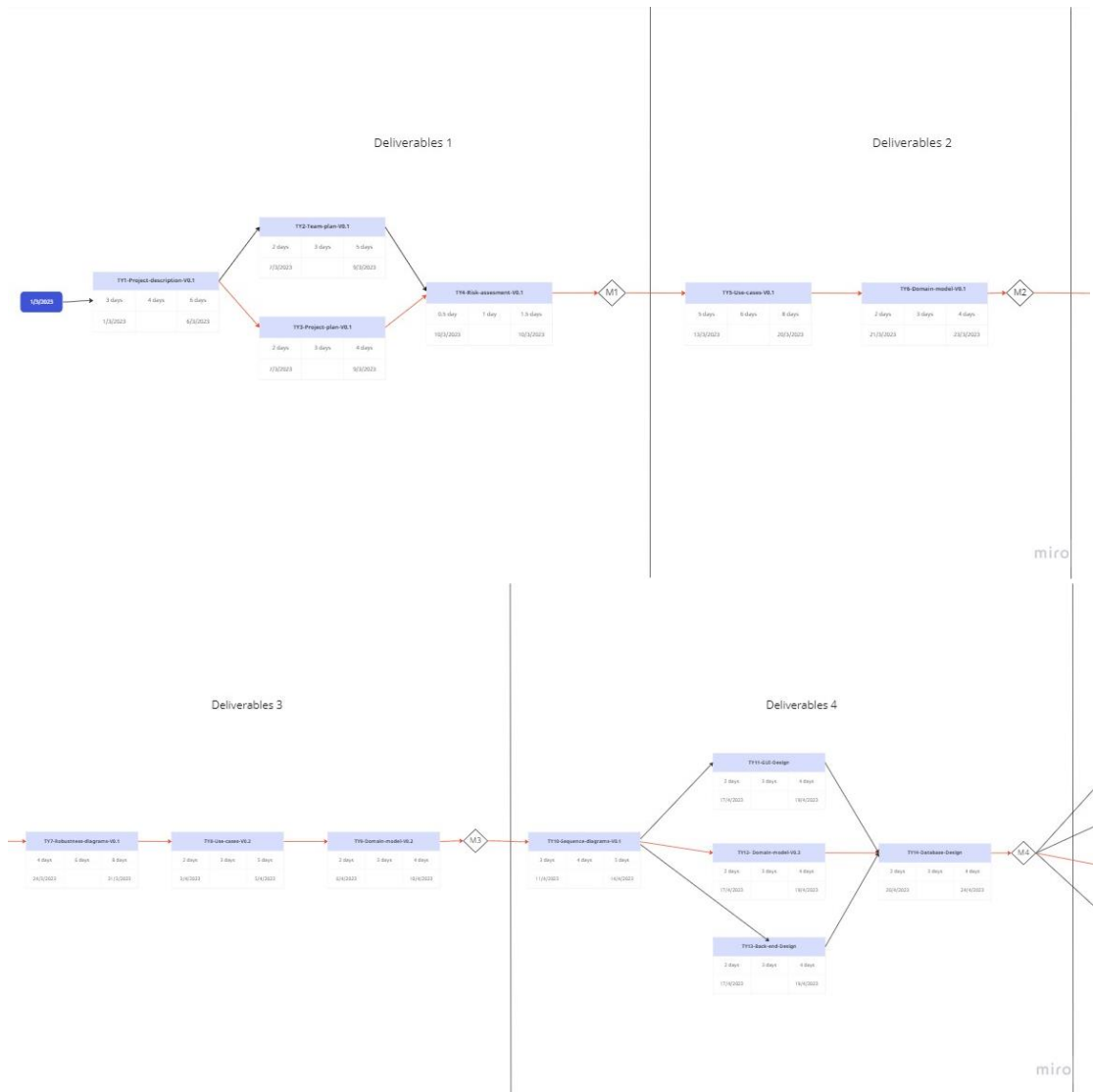
TY1	Project-description-V0.1
TY2	Team-plan-V0.1
TY3	Project-plan-V0.1
TY4	Risk-assesment-V0.1
TY5	Use-cases-V0.1
TY6	Domain-model-V0.1
TY7	Robustness-diagrams-V0.1
TY8	Use-cases-V0.2
TY9	Domain-model-V0.2
TY10	Sequence-diagrams-V0.1
TY11	GUI-Design
TY12	Domain-model-V0.3
TY13	Back-end-Design
TY14	Database-Design
TY15	Class-diagrams-V0.1
TY16	Database-Implementation
TY17	Back-end-Implementation
TY18	GUI-Implementation
TY19	Test-Cases-V0.1
TY20	Final-Versions

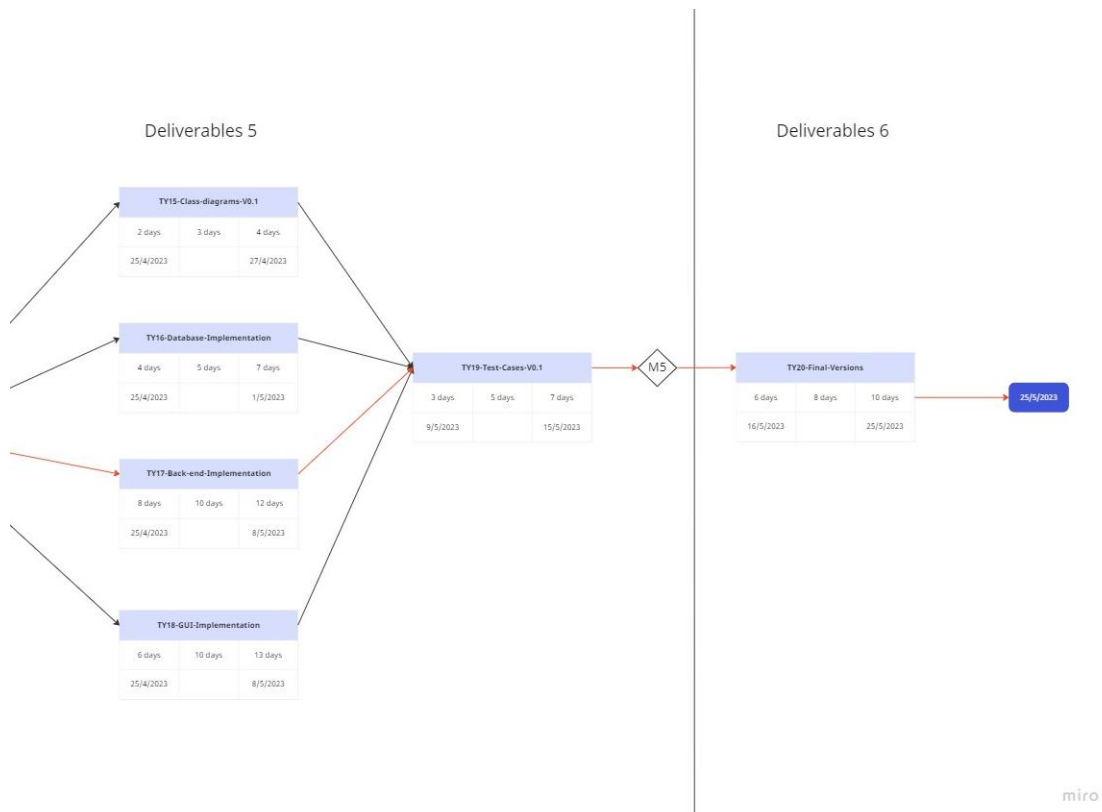
Μετάπειτα κατασκευάζουμε ένα βοηθητικό πίνακα με τα διαφορετικά υπο-έργα, τις εξαρτήσεις(προαπαιτούμενα υπο-έργα), καθώς και τον κανονικό, το χειρότερο και το καλύτερο χρόνο διεκπεραίωσης του κάθε υπο-έργου:

activity	TY1	TY2	TY3	TY4	TY5	TY6	TY7	TY8	TY9	TY10	TY11	TY12	TY13
predecessors	---	---	TY1 TY2	TY1 TY2	TY3 TY4	TY3 TY4	TY5 TY6	TY7	TY7	TY7	TY8 TY9 TY10	TY11	TY12
Expected time	4 days	3 days	3 days	1 day	6 days	3 days	6 days	3 days	3 days	4 days	3 days	3 days	3 days
Best time	3 days	2 days	2 days	0.5 day	5 days	2 days	4 days	2 days	2 days	3 days	2 days	2 days	2 days
Worst time	6 days	5 days	4 days	1.5 days	8 days	4 days	8 days	5 days	4 days	5 days	4 days	4 days	4 days
activity	TY14	TY15	TY16	TY17	T18	TY19	TY20						
predecessors	---	---	TY1 TY2	TY1 TY2	TY3 TY4	TY3 TY4	TY5 TY6						
Expected time	3 days	3 days	5 days	10 days	10 days	5 days	8 days						
Best time	2 days	2 days	4 days	8 days	10 days	3 days	6 days						
Worst time	4 days	4 days	7 days	12 days	13 days	7 days	10 days						

Διάγραμμα PERT:

Έχοντας κάνει την ανάλυση του έργου σε υπο-έργα και έχοντας μια αρχική εκτίμηση των χρονικών πλαισίων και των εξαρτήσεων μεταξύ τους, μπορούμε να προχωρήσουμε στην κατασκευή του διαγράμματος PERT, προκειμένου να δούμε γραφικά τις εξαρτήσεις και τις παραλληλοποιήσεις των υπο-έργων. Το διάγραμμα φαίνεται παρακάτω:





Διάγραμμα Gantt:

Έχοντας υλοποιήσει το διάγραμμα PERT, είναι αρκετά εύκολο να σχεδιάσουμε το διάγραμμα Gantt, το οποίο φαίνεται παρακάτω:

