**Ανάλυση φάσεων**

Καταρχάς οι φάσεις δεν είναι απομονωμένες για να πετύχουμε παραλληλοποίηση του έργου ώστε να τελειώσουμε όσο το δυνατόν περισσότερα τμήματα του λογισμικού μπορούμε σαν ομάδα. Με βάση αυτό και το γεγονός πως το μοντέλο κύκλου ζωής που χρησιμοποιούμε είναι το SCRUM με μικρές τροποποιήσεις για να γίνει βατό για την ομάδα μας, θα αναλύσω πως ακριβώς προχωράνε οι φάσεις.

Καταρχάς οι φάσεις συνυπάρχουν μέσα σε κάθε Sprint του μοντέλου SCRUM μας. Κάθε Sprint είναι ένα υποέργο, το οποίο υποέργο περνάει όλες τις φάσεις. Επομένως πέρα από την παραλληλοποίηση πετυχαίνουμε και μια προτυποποίηση ορισμένων διαδικασιών από όλα τα μέλη ώστε τα επόμενα Sprints να ολοκληρωθούν πιο εύκολα και γρήγορα. Θα κατηγοριοποιήσω τις φάσεις ανά sprint και θα δώσω διεξοδικά σχόλια, προβλήματα, τρόπους αντιμετώπισης και προληπτικά μέτρα για αυτά σε κάθε Sprint.

**Sprint 1**

Το πρώτο Sprint μας είναι η δημιουργία της πρώτης φάσης του λογισμικού (Login Screen και λειτουργικότητα Προσθήκης/Τροποποίησης/Διαγραφής χρήστη από τον Διαχειριστή του λογισμικού). Αυτό συνεπάγεται τα πρώτα βήματα για όλη την ομάδα. Έχουμε πρώτα από όλα τους πελάτες.

**Ανάλυση Απαιτήσεων:**

Οι πελάτες ξεκίνησαν μαζί να δουλεύουν για να βγάλουν ερωτηματολόγια και απαιτήσεις του συνολικού προγράμματος σε ένα γενικό εύρος του. Στην πορεία ξεκίνησαν να φτιάχνουν ερωτηματολόγια και απαιτήσεις για το ειδικό εύρος του πρώτου υποέργου το οποίο είναι η οθόνη σύνδεσης, η λειτουργικότητα της και η βάση δεδομένων της. Ξεκίνησαν με την δημιουργία επιπλέων ερωτηματολογίων και σχεδιασμό προτύπων ερωτηματολογίων για τα επόμενα Sprints. Από τις απαντήσεις αυτών των ερωτηματολογίων (τα οποία απαντήθηκαν μετά από διεξοδική αναζήτηση πληροφοριών στο διαδίκτυο και συγκρίσεων διαφόρων ανταγωνιστικών λογισμικών) εξαγάγανε απαιτήσεις για το λογισμικό (Λειτουργικές και Μη Λειτουργικές). Μετά από αυτή τη διαδικασία οι πελάτες σε συνεργασία με τους προγραμματιστές ξεκίνησαν να αναλύουν τις απαιτήσεις και να προβλέπουν πιθανούς κινδύνους που μπορεί να αντιμετωπιστούν για να πάρουμε κατάλληλα μέτρα αποφυγής σαν ομάδα.

Η φάση της ανάλυσης απαιτήσεων δυστυχώς πήρε περισσότερο διάστημα από όσο υπολογίστηκε και ήταν πρόβλημα το οποίο λύνεται από εδώ και στο εξής με την προτυποποίηση ορισμένων ερωτήσεων για την γρηγορότερη εξαγωγή απαιτήσεων και προδιαγραφών.

**Ανάλυση επικινδυνότητας:**

Βλέποντας τις αρχικές απαιτήσεις αρχίσαμε να σκεφτόμαστε συλλογικά πιθανούς κινδύνους, και πως θα τους αποτρέψουμε από το να εμφανιστούν. Ένας από αυτούς ήταν η βάση δεδομένων στην φάση υλοποίησης. Η αρχική ιδέα ήταν η βάση δεδομένων να είναι Sqlite ώστε να μην επιβαρύνουμε τον υπολογιστή του χρήστη με επιπλέον τρίτα συστήματα dbms. Τελικά μετά από μια ανάλυση καταλήξαμε στο γεγονός πως δεν μπορούμε να το αποφύγουμε οπότε για να αποφύγουμε προβλήματα υλοποίησης και προβλήματα συντήρησης αποφασίσαμε συλλογικά να χρησιμοποιήσουμε βάση δεδομένων με dbms (Oracle XE). Ένας ακόμα κίνδυνος που αντιμετωπίσαμε σε πιο ολικό εύρος της ανάπτυξης του λογισμικού μας ήταν η καθυστέρηση που υπέστη το Sprint και η πιθανή καθυστέρηση λήξης του Sprint. Ενδέχεται να καθυστερήσει το Sprint να λήξει και να φύγουμε εκτός χρονοδιαγράμματος λόγω έλλειψης εμπειρίας, το οποίο ενδέχεται να καθυστερήσει το συνολικό έργο μας αν δεν κάνουμε κάτι για να το αποτρέψουμε. Από την στιγμή που ξέρουμε σε τι οφείλεται αυτή η καθυστέρηση (Έλλειψη εμπειρίας) μπορούμε να το αντιμετωπίσουμε σχετικά εύκολα. Αφού αυτό είναι το πρώτο Sprint και αποκτούμε την εμπειρία που θα χρειαστούμε για μετά, ξεκινάμε να δημιουργούμε πρότυπα σε κάθε φάση ώστε να τα έχουμε έτοιμα για τα επόμενα Sprints και να μην χρειαστεί να ξαναγίνει η διαδικασία από την αρχή με αποτέλεσμα να μην ξαναβγούμε εκτός χρονοδιαγράμματος (Καθυστερήσουμε διότι αν τελειώνουμε πιο γρήγορα δεν είναι πρόβλημα)

**Υλοποίηση κώδικα**

Σε αυτήν την φάση του Sprint αντιμετωπίσαμε ένα πρόβλημα με την βάση δεδομένων το οποίο έπρεπε να λυθεί άμεσα αλλιώς δεν μπορεί να προχωρήσει το project. Η βάση δεδομένων που επιλεχτεί από εμάς (Oracle XE) δεν υποστηρίζει ελληνικά οπότε αυτό μας αφήνει με δύο διαθέσιμες επιλογές, εκ των οποίων η μία είναι η βιώσιμη. Η μία λύση θα ήταν να αλλάξουμε βάση δεδομένων πράγμα το οποίο δεν μας συμφέρει λόγω έλλειψης γνώσεων με άλλες βάσεις δεδομένων, επομένως πρέπει να χρησιμοποιήσουμε αυτά που έχουμε για να προλάβουμε να φέρουμε εις πέρας το έργο σε ένα τόσο μικρό χρονικό διάστημα. Επομένως η επόμενη λύση που προτάθηκε και επιλέχτηκε είναι η μη χρήση ελληνικής συμβολοσειράς στα δεδομένα εισόδου. Το User Inteface θα είναι κανονικά στα ελληνικά αλλά τα δεδομένα που θα εισάγει ο χρήστης θα πρέπει να είναι με λατινικούς χαρακτήρες για να αποφύγουμε προβλήματα με την βάση δεδομένων. (Αν είχαμε περισσότερο χρόνο θα το εξερευνούσαμε περαιτέρω το πρόβλημα και θα δοκιμάζαμε άλλα εργαλεία για την υλοποίηση του λογισμικού μας)

**Τελικός Έλεγχος**

Η σχεδίαση των ελέγχων ξεκίνησε αρκετά νωρίς, πιο συγκεκριμένα ξεκίνησε μόλις βγήκαν οι προδιαγραφές σαν ιδέα του τι θα ελεγχτεί, καθώς στην πορεία ειδικεύτηκαν πάνω στην σχεδίαση των προγραμματιστών.

**Παράδοση στον Πελάτη**

Η πρώτη αλληλεπίδραση με τον πελάτη από άποψη αποτελεσμάτων είναι η επιλογή και αποδοχή των mockup screens η οποία έγινε επιτυχώς από την Ελένη που είναι προγραμματιστής. Δημιουργήθηκαν 2 είδη mockup, το ένα εκ των οποίων ήταν με σκούρο θέμα ενώ το δεύτερο ήταν με ανοιχτό θέμα. Οι πελάτες αποφάσισαν ότι προτιμούν το σκούρο θέμα με αποτέλεσμα οι υπόλοιπες οθόνες να σχεδιαστούν με σκούρο θέμα.