



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

Πανεπιστήμιο Πατρών - Πολυτεχνική Σχολή
Τμήμα Μηχανικών Η/Υ και Πληροφορικής
Πρόγραμμα Μεταπτυχιακών Σπουδών στην Επιστήμη και
Τεχνολογία Υπολογιστών

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Σχεδιασμός, υλοποίηση και αξιολόγηση Web 2.0 εφαρμογών
μέσω χρήσης ευέλικτων μεθόδων πελατοκεντρικής προσέγγισης

ΑΘΗΝΑ ΝΙΚΟΥ

ΑΡΙΘΜΟΣ ΜΗΤΡΩΟΥ: 943

Επιβλέπων

Καθηγητής Ιωάννης Γαροφαλάκης

Συνεπιβλέπων

Δρ. Ιωάννης Χατζηγιαννάκης

Πάτρα, Σεπτέμβριος 2016

ΠΙΣΤΟΠΟΙΗΣΗ

Πιστοποιείται ότι η διπλωματική εργασία με θέμα

**Σχεδιασμός, υλοποίηση και αξιολόγηση Web 2.0 εφαρμογών
μέσω χρήσης ευέλικτων μεθόδων πελατοκεντρικής προσέγγισης**

της φοιτήτριας του Τμήματος Μηχανικών Η/Υ και Πληροφορικής

Αθηνά Νίκου

(Α.Μ.: 943)

παρουσιάστηκε δημόσια και εξετάστηκε στο τμήμα Μηχανικών Η/Υ
και Πληροφορικής στις

___/___/___

Ο Επιβλέπων

Ιωάννης Γαροφαλάκης
Καθηγητής

Στοιχεία διπλωματικής εργασίας

Θέμα:

Σχεδιασμός, υλοποίηση και αξιολόγηση Web 2.0 εφαρμογών
μέσω χρήσης ευέλικτων μεθόδων πελατοκεντρικής προσέγγισης

Φοιτήτρια:

Αθηνά Νίκου

Ομάδα επίβλεψης:

Καθηγητής Ιωάννης Γαροφαλάκης

Δρ. Ιωάννης Χατζηγιαννάκης

Εξεταστική επιτροπή:

Καθηγητής Ιωάννης Γαροφαλάκης, ΤΜΗΥΠ

Καθηγητής Αθανάσιος Τσακαλίδης, ΤΜΗΥΠ

Επίκ. Καθηγητής Χρήστος Μακρής, ΤΜΗΥΠ

Η εργασία αυτή γράφτηκε στο $\text{\LaTeX}$ και χρησιμοποιήθηκε η
γραμματοσειρά GFS Didot του Greek Font Society.

Περίληψη

Σύμφωνα με έρευνες που έχουν διεξαχθεί, σημαντική μερίδα των διαδικτυακών εφαρμογών που αναπτύσσονται με σκοπό να εξυπηρετήσουν τους τελικούς χρήστες αποτυγχάνουν στο να τους ικανοποιήσουν και να ενταχθούν στην καθημερινότητά τους, παρά το γεγονός ότι οι δημιουργοί τους έχουν επενδύσει χρόνο, κόπο και συχνά σημαντικό κεφάλαιο. Ένας από τους σημαντικότερους λόγους ύπαρξης αυτού του φαινομένου οφείλεται στο γεγονός ότι ο κύκλος ζωής των εφαρμογών αυτών χαρακτηρίζεται από έλλειψη κατανόησης των αναγκών των εν δυνάμει χρηστών και από αδυναμία έγκαιρης προσαρμογής στην ευμεταβλητότητα που χαρακτηρίζει τις ανάγκες και τις επιθυμίες τους.

Στην παρούσα μεταπτυχιακή διπλωματική εργασία παρουσιάζεται ο σχεδιασμός, η υλοποίηση και η αξιολόγηση Web 2.0 εφαρμογών με χρήση ευέλικτων μεθοδολογιών ανάπτυξης λογισμικού (Lean - Agile Software Development) οι οποίες έχουν σαν κύριο άξονά τους την πελατοκεντρική προσέγγιση. Οι μεθοδολογίες αυτές, συνοπτικά, περιλαμβάνουν στις αρχές τους τη συνεχή και επαναληπτική ενσωμάτωση του τελικού πελάτη καθ' όλη τη διάρκεια του κύκλου ζωής του λογισμικού, της διαχείρισης των διαθέσιμων πόρων με τρόπο που να αυξάνει την ικανοποίηση του τελικού πελάτη, περιορίζοντας ταυτόχρονα τις σπατάλες τόσο σε χρονικό όσο και σε οικονομικό επίπεδο χωρίς να γίνονται εκπτώσεις στην ποιότητα του τελικού προϊόντος. Η διαφορετικότητά τους σε σχέση με τις παραδοσιακές μεθόδους έγκειται στην ευμεταβλητότητα και την έγκαιρη και σταδιακή διανομή λειτουργικού λογισμικού, χαρακτηριστικά που ταιριάζουν στη φύση των web 2.0 εφαρμογών.

Στα πλαίσια της παρούσας εργασίας εστιάζουμε στην ανάπτυξη και την αξιολόγηση Web 2.0 εφαρμογών οι οποίες εξυπηρετούν την ηλεκτρονική παραγγελία φαγητού. Αρχικά παρουσιάζεται μια πλατφόρμα παραγγελίας φαγητού η οποία βασίζεται ως επί το πλείστον σε μεθόδους ανάπτυξης λογισμικού δομημένης σχεδίασης (π.χ. μοντέλο καταρράκτη – Waterfall), στις οποίες το προϊόν δημιουργείται εκ του μηδενός και ο τελικός πελάτης έρχεται σε επαφή με αυτό όταν έχει ολοκληρωθεί πλήρως. Στη συνέχεια, μέσω συλλογής απόψεων δυνητικών χρηστών της εφαρμογής, αξιολογούμε το χρονικό κυρίως κόστος που υπήρξε για την ανάπτυξη τμημάτων της εφαρμογής που θεωρήθηκαν μη επιθυμητά. Μέσω της αξιολόγησης αυτής γίνεται αντιληπτό γιατί, ιδίως σε εποχές οικονομικής κρίσης και αυξημένου ανταγωνισμού, οι μέθοδοι αυτές δεν είναι ιδιαιτέρως αποδοτικές στον τομέα ανάπτυξης Web 2.0 εφαρμογών. Στον αντίποδα παρουσιάζεται μια δεύτερη εφαρμογή ιδίου σκοπού η οποία σχεδιάστηκε με τον πιθανό τελικό χρήστη να έχει άμεση συμμετοχή. Στα πλαίσια αυτής της πελατοκεντρικής προσέγγισης διεξήχθη έρευνα μέσω της οποίας διερευνάται το επίπεδο εξοικείωσης του Έλληνα που έχει πρόσβαση στο

διαδίκτυο με την ηλεκτρονική παραγγελία φαγητού, ενώ ταυτόχρονα αποκαλύπτονται και τα χαρακτηριστικά εκείνα που μπορούν να κάνουν μια τέτοια εφαρμογή να είναι πιο προσφιλής σε αυτόν. Κάθε μια από τις εφαρμογές αναπτύχθηκε με χρήση διαφόρων τεχνολογιών όπως Spring Boot Framework, Hibernate, Thymeleaf, Bootstrap Framework, JQuery, HTML5, Javascript.

Abstract

According to surveys, a high percentage of web applications that are developed in order to serve the final users fail to satisfy them and be part of their everyday life despite the fact that their creators have spent a lot of time and money. This happens due to the fact that the delivered applications are characterized by lack of understanding the final customers needs and by the inability to adapt to their continuously changing needs.

This work aims to present the development of Web 2.0 applications by utilizing the principles of Lean - Agile software development methods. These methods articulate the importance of customer centricity. Lean - Agile consists of an iterative and incremental development approach, which means that during the software development more than one iterations of the development cycle may be in progress at the same time. The basic idea is to develop a software product through repeated cycles and in smaller portions, delivering working software frequently. The highest priority of course is to satisfy customers through this continuous delivery of valuable software. This way, lean agile methodology allows for customers' input to be incorporated more often, and working with customers continually, knowledge about their needs is being improved.

In this work we emphasize to the development of web applications that serve the online food ordering. Firstly we present a platform that has been developed based on the principals of traditional development methods like waterfall model. These methods are plan-driven and the applications are launched only by the time they are totally completed. Demonstrating the application to a number of potential users we managed to detect the features that were considered as unnecessary and evaluate the cost of them. Afterwards an online food ordering application that its development is based on the Lean - Agile software development methods is presented. A customer centric approach is applied by conducting a survey that aimed at getting feedback about Greek consumers' food ordering behavior, demonstrating their familiarization with electronic food ordering applications and checking how they evaluate a series of features of an electronic food ordering application. The aforementioned applications have been developed by using a variety of technologies such as Spring Boot Framework, Hibernate, Thymeleaf, Bootstrap Framework, JQuery, HTML5, Javascript.

Ευχαριστίες

Όσο κι αν φαίνεται σαν ατομική δουλειά η παρούσα εργασία, στην πραγματικότητα βοήθησαν αρκετοί άνθρωποι, ο καθένας με το δικό του τρόπο, για να ολοκληρωθεί. Θα ήθελα λοιπόν να ευχαριστήσω τον Καθηγητή του τμήματος Μηχανικών Ηλεκτρονικών Υπολογιστών και Πληροφορικής κ. Ιωάννη Γαροφαλάκη που ως επιβλέποντας καθηγητής μου, μου έδωσε την ευκαιρία να ασχοληθώ με το συγκεκριμένο αντικείμενο. Επίσης θα ήθελα να ευχαριστήσω ένθερμα τον Δρ. Ιωάννη Χατζηγιαννάκη για την καθοδήγηση και την υποστήριξή του καθ'όλη τη διάρκεια εκπόνησης της παρούσας εργασίας. Θα ήθελα να ευχαριστήσω ιδιαίτερα τον υποψήφιο διδάκτορα Δημήτριο Αμαξηλάτη για την ουσιαστική βοήθειά του και τις γνώσεις που μου προσέφερε σε όλη την πορεία της διπλωματικής μου. Τέλος ευχαριστώ τους φίλους μου Μαρία και Βασίλειο Κασιώλα που με στήριξαν αδιαμαρτύρητα καθ'όλη τη διάρκεια των μεταπτυχιακών μου σπουδών.

Δημοσιεύσεις

Στα πλαίσια εκπόνησης της παρούσας μεταπτυχιακής διπλωματικής εργασίας προέκυψαν επιστημονικά ευρήματα τα οποία οδήγησαν στην παρακάτω δημοσίευση:

Athina Nikou and Ioannis Chatzigiannakis. 2015. *Applying a customer centric development approach for web 2.0 applications*. In Proceedings of the 19th Panhellenic Conference on Informatics (PCI '15), Karanikolas Nikitas N., Akoumianakis Demosthenes, Nikolaidou Mara, Vergados Dimitris, and Xenos Michalis (Eds.). ACM, New York, NY, USA, 245-246.

ΠΕΡΙΕΧΟΜΕΝΑ

I	ΕΙΣΑΓΩΓΗ	1
1	Εισαγωγή	3
1.1	Κίνητρο εκπόνησης διπλωματικής εργασίας	3
1.2	Στόχοι διπλωματικής εργασίας	4
1.3	Συνεισφορά	5
1.4	Δομή διπλωματικής εργασίας	6
2	Πελατοκεντρικότητα	7
2.1	Εισαγωγή	7
2.2	Η σημασία της πελατοκεντρικότητας στην ανάπτυξη web-based λογισμικού.	9
2.2.1	Η εμπειρία χρήστη (user experience - UX)	9
2.2.2	Η εμπειρία πελάτη (customer experience - CX)	11
2.2.3	Σύγκριση εμπειρίας χρήστη (UX) και εμπειρίας πελάτη (CX)	11
2.2.4	Η πελατοκεντρικότητα και η ανάπτυξη web-based λογισμικού	12
2.3	Μοντέλο ανθρωποκεντρικού σχεδιασμού-ISO 9241-210:2010	14
2.4	Πλεονεκτήματα πελατοκεντρικής προσέγγισης έναντι άλλων	16
2.4.1	Τα πλεονεκτήματα της πελατοκεντρικότητας	17
2.4.2	Μύθοι σχετικά με την πελατοκεντρικότητα	18
II	ΠΑΡΑΔΟΣΙΑΚΕΣ ΜΕΘΟΔΟΙ	21
3	Παραδοσιακές Μέθοδοι Ανάπτυξης Λογισμικού	23

3.1	Εισαγωγή	23
3.1.1	Διαδικασία Παραγωγής Λογισμικού	24
3.1.2	Μοντέλο Διαδικασίας Παραγωγής Λογισμικού	24
3.1.3	Σημασία εφαρμογής μοντέλου διαδικασίας παραγωγής λογισμικού	26
3.2	Παραδοσιακές Μέθοδοι Ανάπτυξης Λογισμικού	26
3.2.1	Μοντέλο Καταρράκτη (Waterfall)	27
3.2.2	Γενικά Χαρακτηριστικά Παραδοσιακών Μεθόδων	30
3.3	Αξιολόγηση Παραδοσιακών μεθόδων	31
3.3.1	Δυνάμεις και αδυναμίες	31
3.3.2	Η σχέση τους με την πελατοκεντρικότητα	32
3.3.3	Αξιολόγηση καταλληλότητας χρήσης τους σε ανάπτυξη Web 2.0 λογισμικού	33
4	Περίπτωση Χρήσης Παραδοσιακών Μεθόδων	35
4.1	Πλατφόρμα <i>EAT-IN</i>	35
4.1.1	Εισαγωγή - Η ιδέα	35
4.1.2	Ανάλυση Απαιτήσεων	36
4.1.3	Υλοποίηση Πλατφόρμας	38
4.1.4	Αξιολόγηση Πλατφόρμας	52
4.2	Τεχνολογίες που χρησιμοποιήθηκαν	54
4.2.1	Spring Framework	55
4.2.2	Spring Boot	58
4.2.3	Hibernate	60
4.2.4	Λοιπές τεχνολογίες	61
4.3	Συμπεράσματα-Συζήτηση	62
III	ΕΥΕΛΙΚΤΕΣ ΜΕΘΟΔΟΙ	63
5	Ευέλικτες Μέθοδοι Ανάπτυξης Λογισμικού	65
5.1	Εισαγωγή	65
5.1.1	Ανάγκη για ταχεία Ανάπτυξη Λογισμικού	65
5.1.2	Αρχές - Αξίες - Ιδιότητες Ευέλικτων Μεθόδων	67
5.1.3	Τα χαρακτηριστικά των ευέλικτων μεθόδων	70
5.2	Μέθοδοι	72
5.2.1	Ακραίος προγραμματισμός	72
5.2.2	Lean Ανάπτυξη Λογισμικού	76
5.2.3	Ευέλικτες Μέθοδοι - Καινοτόμες Προσεγγίσεις	80
5.3	Ανάγκη Ευελιξίας στην Ανάπτυξη Web 2.0 Λογισμικού	83
5.4	Σύγκριση Ευέλικτων και Παραδοσιακών Μεθόδων	85
6	Περίπτωση Χρήσης Ευέλικτων Μεθόδων	89
6.1	Εισαγωγή	89
6.2	Εμπειρική Ανάλυση - Μεθοδολογία και Στόχοι	91
6.2.1	Στόχοι Έρευνας	91

6.2.2	Μεθοδολογία Έρευνας	92
6.2.3	Διερεύνηση προϋπάρχουσας κατάστασης - Ηλεκτρο- νική παραγγελία φαγητού στην Ελλάδα	93
6.3	Αξιολόγηση Εμπειρικής Ανάλυσης	97
6.3.1	Δεύτερη Φάση - Πρώτη Διάδραση	97
6.3.2	Αποτελέσματα Πρώτης Διάδρασης	104
6.4	Αξιοποίηση Εμπειρικής Ανάλυσης	104
6.4.1	Δεύτερη Φάση - Δεύτερη Διάδραση	104
6.4.2	Ελάχιστο Βιώσιμο Προϊόν (MVP) - Εφαρμογή 2EAT	106
6.4.3	B2B Πελάτες	107
6.4.4	B2C Πελάτες	108
6.4.5	Αποτελέσματα Δεύτερης Διάδρασης	112
6.5	Τεχνολογίες που χρησιμοποιήθηκαν	114
6.6	Συμπεράσματα - Συζήτηση	115
IV	ΣΥΜΠΕΡΑΣΜΑΤΑ	117
7	Συμπεράσματα	119
7.1	Συμπεράσματα	119
V	ΠΑΡΑΡΤΗΜΑ	123
8	Παράρτημα	125
8.1	Ερωτηματολόγια	125
8.1.1	Ερωτήσεις συνεντεύξεων (EAT-IN)	125
8.1.2	Ερωτηματολόγια (2EAT) - Πρώτη Διάδραση	126
8.1.3	Ερωτηματολόγια (2EAT) - Δεύτερη Διάδραση	133
	Κατάλογος σχημάτων	137
	Κατάλογος πινάκων	139
	Βιβλιογραφία	141

Μέρος Ι

ΕΙΣΑΓΩΓΗ

ΚΕΦΑΛΑΙΟ

1

ΕΙΣΑΓΩΓΗ

1.1 Κίνητρο εκπόνησης διπλωματικής εργασίας

Είναι κοινώς αποδεκτό ότι στην εποχή μας, ο τομέας ανάπτυξης λογισμικού διανύει μια περίοδο έξαρσης. Η τεχνική μετεξέλιξη του Παγκόσμιου Ιστού, δηλαδή το Web 2.0, έδωσε τη δυνατότητα της άμεσης συμμετοχής του τελικού χρήστη, καθιστώντας τον κυρίαρχο παράγοντα στην ανάπτυξη λογισμικού. Η συμβολή των χρηστών και καταναλωτών στο Web 2.0 δεν περιορίζεται μόνο στην δημιουργία και διακίνηση περιεχομένου. Ιδίως στο τμήμα της ανάπτυξης web-based λογισμικού, τόσο από τυπικές ή νεοφυείς επιχειρήσεις όσο και από μεμονωμένες μικρές ομάδες, ο χρήστης αποτελεί πλέον πάροχο ανατροφοδότησης προκειμένου να επιτευχθούν στόχοι, να αναπτυχθούν νέα προϊόντα αλλά και βελτιωθούν τα ήδη υπάρχοντα. Συνεπώς, η σημασία της ύπαρξης μιας πελατοκεντρικής προσέγγισης ανάπτυξης λογισμικού παύει πλέον να αποτελεί επιλογή και μετατρέπεται σε αναγκαιότητα.

Όλη αυτή η ανάπτυξη του παγκόσμιου ιστού έχει οδηγήσει, τα τελευταία χρόνια, όλο και περισσότερους να ασχοληθούν με την ανάπτυξη εφαρμογών οι οποίες έχουν σαν στόχο να εξυπηρετήσουν ανάγκες των τελικών χρηστών, μικρότερες ή μεγαλύτερες και ενίοτε ακόμα και να δημιουργήσουν καινούργιες. Ένας από τους τομείς που ενδιαφέρει ιδιαίτερα τους πιθανούς επιτυχημένους παρόχους είναι το ηλεκτρονικό εμπόριο το οποίο βάσει ερευνών τείνει σταδιακά να αντικαταστήσει το φυσικό εμπόριο. Συγκεκριμένα, ο τομέας ηλεκτρονικής παραγγελίας φαγητού στη χώρα μας παρουσιάζεται ως ανερχόμενος και πολλά υποσχόμενος. Στον τομέα

αυτό εστιάσαμε στην παρούσα εργασία, ο οποίος, όπως επίσης προκύπτει απο επίσημες έρευνες, γνωρίζει ραγδαία άνοδο καθώς την τελευταία πενταετία έχει καταφέρει να απασχολήσει σχεδόν το 50% των καταναλωτών που έχουν πρόσβαση στο διαδίκτυο. Δεν είναι τυχαίο συνεπώς που όλο και περισσότερες πλατφόρμες και εφαρμογές ηλεκτρονικής παραγγελίας φαγητού κάνουν την εμφάνισή τους.

Παρ' όλ' αυτά έχει παρατηρηθεί ότι πολλές προσπάθειες δεν είχαν την αναμενόμενη επιτυχημένη πορεία. Οι λόγοι στους οποίους οφείλεται αυτό ποικίλουν, ένας όμως απο αυτούς, πέραν της ελλιπούς αξιολόγησης του ανταγωνισμού, είναι η ανεπαρκής κατανόηση του κοινού στο οποίο απευθύνονται. Συνεπώς, πριν την όποια προσπάθεια, είναι σημαντική η διερεύνηση των επιθυμιών του τελικού χρήστη και είναι σφόδρον ο σχεδιασμός της εκάστοτε εφαρμογής να γίνεται βασιζόμενος στα αποτελέσματα της διερεύνησης αυτής. Οι σύγχρονες φιλοσοφίες-μεθολογίες ανάπτυξης web-based λογισμικού, σε αντίθεση με τις "βαριές" παραδοσιακές μεθόδους, έχουν κατανοήσει τη σημασία και την αξία αυτής της προσέγγισης, εξ' ου και τείνουν να είναι περισσότερο πελατοκεντρικές. Μία από τις βασικές και εξέχως αξιοποιούμενες μεθολογίες πελατοκεντρικής προσέγγισης είναι το ευέλικτο μοντέλο ανάπτυξης λογισμικού, το οποίο αποτελεί τον άξονα βάσει του οποίου κινηθήκαμε σε αυτή την εργασία. Ο συνδυασμός μεθόδων πελατοκεντρικής προσέγγισης και χρήσης ισχυρών τεχνολογιών που απλοποιούν τις διαδικασίες υλοποίησης λογισμικού μπορούν να διευκολύνουν την ανάπτυξη web-based λογισμικού πολύπλευρα. Στην εποχή μας, οι τεχνολογικές εξελίξεις αλλά και ο ευρύς ανταγωνισμός απαιτούν ταχύτητα και προσαρμοστικότητα, συνεπώς είναι απαραίτητη η σωστή αξιοποίησή της υπάρχουσας πληροφορίας προκειμένου η όποια προσπάθεια να απαιτεί λιγότερο χρόνο, κόπο και το λιγότερο δυνατό κόστος.

1.2 Στόχοι διπλωματικής εργασίας

Στόχος της παρούσας διπλωματικής εργασίας είναι να αναδείξει, διαμέσου των εμπειριών που αποκτήσαμε κατά τη διάρκεια εκπόνησής της, αποδοτικές μεθόδους σχεδιασμού, υλοποίησης και αξιολόγησης Web 2.0 λογισμικού που βασικό χαρακτηριστικό τους είναι η πελατοκεντρικότητα. Στην εργασία αυτή εστιάσαμε αρχικά στην υλοποίηση μιας πλατφόρμας ηλεκτρονικής παραγγελίας φαγητού, της οποίας ο σχεδιασμός βασίστηκε σε παραδοσιακές μεθόδους ανάπτυξης λογισμικού, πιο συγκεκριμένα στο μοντέλο του καταρράκτη. Η συμμετοχή των τελικών πελατών εντάχθηκε μόνο στην αρχική φάση ανάλυσης των απαιτήσεων και εν συνεχεία προχωρήσαμε χωρίς επιπλέον διάδραση με αυτούς, όπως ακριβώς ορίζει το χρησιμοποιηθέν μοντέλο. Όταν το προϊόν ολοκληρώθηκε τέθηκε προς αξιολόγηση από πιθανούς τελικούς χρήστες, τα αποτελέσματα της οποίας και παρουσιάζουμε σε σχετικό κεφάλαιο.

Εν συνεχεία, βασιζόμενοι στη φιλοσοφία των ευέλικτων μεθοδολογιών αναπτύξαμε μια δεύτερη εφαρμογή-πλατφόρμα, όπου ο τελικός χρήστης ήταν κύριος παράγοντας καθ' όλη τη διάρκεια ανάπτυξης της. Η διαδικασία σχεδιασμού, ανάπτυξης και αξιολόγησής της είναι επαναληπτική και επαυξητική. Μελετήσαμε εμπεριστατωμένα τις επιθυμίες του καταναλωτικού κοινού, μέσω έρευνας που διεξάγαμε, με στόχο να ανακαλύψουμε εκείνα τα χαρακτηριστικά που μπορούν να καταστήσουν μια εφαρμογή ηλεκτρονικής παραγγελίας φαγητού ελκυστική προς το μέσο χρήστη. Έτσι λοιπόν μελετήθηκαν σε βάθος η σχέση που έχει ο Έλληνας χρήστης με την ηλεκτρονική παραγγελία φαγητού, αν είναι θετικά προκείμενος και ποιά χαρακτηριστικά θα τον έκαναν να χρησιμοποιήσει μια εφαρμογή τέτοιου σκοπού. Η γνώση που αποκτήθηκε μέσω της μελέτης της συμπεριφοράς του μέσου χρήστη αξιοποιήθηκε στη δημιουργία του συγκεκριμένου Web 2.0 λογισμικού ηλεκτρονικής παραγγελίας, που κινείται με γνώμονα κυρίως τις επιθυμίες του.

Οι προαναφερθείσες προσεγγίσεις στοχεύουν στην αξιολόγηση των διαφόρων μεθόδων ανάπτυξης λογισμικού και στο να αναδειχθεί ποιά από αυτές τις προσεγγίσεις είναι καταλληλότερη για την ανάπτυξη Web 2.0 εφαρμογών από ολιγομελείς ή μονομελείς ομάδες. Ταυτόχρονα, στοχεύουν στο να παρουσιάσουν την απόδοση της κάθε μίας μεθόδου, όταν αυτές συνδυάζονται με σύγχρονες τεχνολογίες που διευκολύνουν το σχεδιασμό και την ανάπτυξη διαδικτυακών εφαρμογών. Τέλος, εστιάζοντας στην ηλεκτρονική παραγγελία φαγητού, επιχειρείται η παρουσίαση της υπάρχουσας κατάστασης στην ελληνική αγορά αλλά και η διερεύνηση των γενικότερων απαιτήσεων των Ελλήνων όσον αφορά τις εφαρμογές-πλατφόρμες που εξυπηρετούν το σκοπό αυτό.

1.3 Συνεισφορά

Στα πλαίσια της εργασίας αυτής διερευνήθηκε η θετική επιρροή των ευέλικτων μεθόδων πελατοκεντρικής προσέγγισης στο σχεδιασμό και την ανάπτυξη Web 2.0 εφαρμογών και παρουσιάστηκε ο λόγος που οι παραδοσιακές μέθοδοι κρίνονται πιο δύσκαμπτες για τη συγκεκριμένη χρήση. Ερευνήθηκε η συμπεριφορά και στάση του μέσου Έλληνα χρήστη που έχει πρόσβαση στο διαδίκτυο απέναντι στις εφαρμογές ηλεκτρονικής παραγγελίας φαγητού. Αξιολογήθηκε μια σειρά χαρακτηριστικών των προαναφερθέντων εφαρμογών, τα οποία πρέπει να αποτελούν βασικό γνώμονα πριν την υλοποίησή τους. Αξιοποιήθηκε η κεκτημένη αυτή γνώση για τη δημιουργία εφαρμογών που εξυπηρετούν την ηλεκτρονική παραγγελία φαγητού. Τέλος, μελετήθηκαν οι τεχνολογίες Spring Boot, Hibernate, Thymeleaf, Bootstrap καθώς και διάφορες διεπαφές προγραμματισμού εφαρμογών (APIs) όπως το Twitter Streaming API και το PayPal API και αξιοποιήθηκαν για την ανάπτυξη των διαδικτυακών εφαρμογών που υλοποιήθη-

καν. Η συνεισφορά της εργασίας αυτής έγκειται στο συνδυασμό αρχών των μεθόδων Agile, Lean και Customer Development και στην ανάδειξη της υπεροχής τους έναντι των παραδοσιακών μεθόδων όταν πρόκειται για ανάπτυξη web 2.0 λογισμικού. Ο συνδυασμός αυτός μπορεί να οδηγήσει στην ανάπτυξη λογισμικού βασισμένου σε αρχές (principle-driven) με αποτέλεσμα η ανάπτυξη του τελικού προϊόντος να απαιτεί λιγότερο χρόνο, μικρότερο κόστος, να είναι περισσότερο πελατοκεντρική και να εξασφαλίζει μεγαλύτερο βαθμό ικανοποίησης των τελικών χρηστών.

1.4 Δομή διπλωματικής εργασίας

Η παρούσα μεταπτυχιακή εργασία δομείται από επτά θεματικές ενότητες. Η πρώτη ενότητα αποτελεί την εισαγωγή της εργασίας, η οποία περιλαμβάνει το κίνητρο ενασχόλησης με το συγκεκριμένο θέμα, τους στόχους εκπόνησης, τη συνεισφορά της εργασίας και τέλος τη δομή αυτής. Στη δεύτερη ενότητα γίνεται παρουσίαση της γενικής έννοιας της πελατοκεντρικότητας στην ανάπτυξη λογισμικού και στη σημασία αυτής. Στο τρίτο κεφάλαιο δίνεται μια σύντομη περιγραφή του θεωρητικού πλαισίου των παραδοσιακών μεθόδων ανάπτυξης λογισμικού, ενώ στο τέταρτο κεφάλαιο παρουσιάζεται η πλατφόρμα ηλεκτρονικής παραγγελίας φαγητού, ο σχεδιασμός και η υλοποίηση της οποίας βασίστηκαν στο παραδοσιακό μοντέλο του καταρράκτη. Η πέμπτη ενότητα παρουσιάζει τη φιλοσοφία των ευέλικτων μεθόδων ανάπτυξης λογισμικού και γίνεται πιο λεπτομερής αναφορά στις σημαντικότερες εκ των μεθόδων αυτής της φιλοσοφίας. Στο έκτο κεφάλαιο τίθεται σε εφαρμογή αυτή η φιλοσοφία για τη σχεδίαση και υλοποίηση μιας δεύτερης εφαρμογής-πλατφόρμας ηλεκτρονικής παραγγελίας φαγητού, η οποία βασίζεται και ικανοποιεί σε μεγάλο βαθμό την πελατοκεντρικότητα που ορίζουν οι ευέλικτες μέθοδοι. Τέλος, παρατίθενται στην τελευταία ενότητα τα συμπεράσματα της μεταπτυχιακής αυτής εργασίας, ακολουθούμενη από τη βιβλιογραφία.

ΚΕΦΑΛΑΙΟ

2

ΠΕΛΑΤΟΚΕΝΤΡΙΚΟΤΗΤΑ

2.1 Εισαγωγή

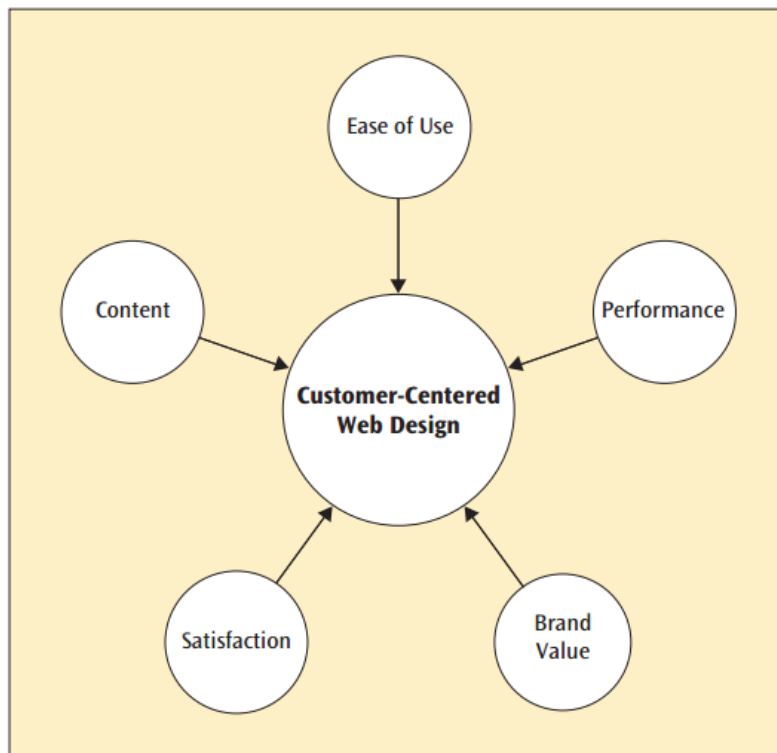
Ο παγκόσμιος ιστός δεν αποτελεί πλέον καινοτομία. Η ανάπτυξη του είναι ραγδαία, οι παροχές του πολλές και οι χρήστες του είναι πια υποψιασμένοι, καλύτεροι γνώστες σε σχέση με προηγούμενες δεκαετίες και συνεπώς πιο απαιτητικοί. Η δυνατότητα να διατηρηθεί το ενδιαφέρον ενός χρήστη - πελάτη μπορεί να συνδεθεί άμεσα με την έννοια του κέρδους και της επιτυχίας, πράγμα το οποίο πλέον κατανοεί η πλειοψηφία όσων δραστηριοποιούνται στην ανάπτυξη Web 2.0 λογισμικού. Η πορεία της ανάπτυξης web-based λογισμικού αναδεικνύει τη σημασία της πελατοκεντρικότητας όπως φαίνεται στις ενότητες που ακολουθούν.

Η Εξέλιξη της Ανάπτυξης Web-based λογισμικού

1. Πρώτη Γενιά: Η άποψη που υπερίσχυε ήταν "χτίσε, και θα έρθουν". Η λογική αυτή οδηγούσε ταλαντούχους ανθρώπους σε μια διαδικασία συνολικής διαχείρισης όλων των συνιστωσών ανάπτυξης λογισμικού, από τη θέσπιση του επιχειρηματικού μοντέλου έως την ανάπτυξη και διάθεση του προϊόντος λογισμικού. Στο τέλος της διαδικασίας υπήρχε ένα προϊόν το οποίο δεν μπορούσαν να γνωρίζουν πόσο αποδοτικό ήταν από την οπτική του πελάτη, ούτε να αξιολογήσουν την πολυπλοκότητα και την πιθανή δυσχρηστία του προϊόντος τους.
2. Δεύτερη Γενιά: Κατά τη διάρκεια αυτής της περιόδου, η άποψη που

επικρατούσε χαρακτηρίζεται από την έκφραση "διαφήμιση ότι το πουλάς ηλεκτρονικά και οι πελάτες θα έρθουν". Η άποψη αυτή οδήγησε πολλές νεοφυείς επιχειρήσεις να εισάγουν το ".com" στην προώθηση των προϊόντων τους και να επενδύσουν μεγάλο μέρος του κεφαλαίου τους στην προσέλκυση πελατών μέσω διαφήμισης. Επίσης, ήδη υπάρχουσες εταιρείες που είχαν εδραιωθεί στο χώρο διεξήγαν κοστοβόρες καμπάνιες προκειμένου να διαφημιστούν. Η στρατηγική αυτή όμως δεν λειτούργησε αποδοτικά γιατί η ανάπτυξη λογισμικού στο διαδίκτυο ήταν ακόμα σε πρώιμο στάδιο.

3. Τρίτη Γενιά: Η έννοια της πελατοκεντρικότητας αρχίζει να βρίσκει χρήση και να αποκτά ισχύ. Η πελατοκεντρική ανάπτυξη λογισμικού (customer centered design) οδηγεί σε προϊόντα λογισμικού που δημιουργούν θετική εμπειρία χρήστη (user experience). Ένα προϊόν λογισμικού λογίζεται ως πελατοκεντρικό όταν οι ίδιοι οι χρήστες υποστηρίζουν ότι παρέχει ευκολία στη χρήση, καλή απόδοση, αίσθημα εμπιστοσύνης και ικανοποίηση των αναγκών τους. Αυτά τα χαρακτηριστικά διαφοροποιούν την πελατοκεντρική προσέγγιση από τις υπόλοιπες [Σχήμα 2.1].



Σχήμα 2.1: Τα χαρακτηριστικά "κλειδιά" που κατευθύνουν την πελατοκεντρική ανάπτυξη λογισμικού

4. Τέταρτη Γενιά: Στις μέρες μας οι περισσότερες τεχνολογίες και επι-

χειρηματικά μοντέλα έχουν εξελιχθεί. Οι δυνατότητες που παρέχονται για την ανάπτυξη web-based λογισμικού έχουν αυξηθεί, ταυτόχρονα όμως η επίτευξη της πελατοκεντρικότητας αποτελεί μεγαλύτερη πρόκληση, καθώς οι χρήστες είναι πιο εξοικειωμένοι και πιο απαιτητικοί. Η χρήση μεθόδων τόσο για την προσέγγιση του πελάτη, όσο και γενικότερα στην όλη διαδικασία ανάπτυξης του λογισμικού, κρίνεται ιδιαίτερα σημαντική.

2.2 Η σημασία της πελατοκεντρικότητας στην ανάπτυξη web-based λογισμικού.

Όπως αναφέρθηκε προηγούμενα, με το πέρασμα των χρόνων κατέστη δυνατό να κατανοηθεί ότι για να είναι ένα προϊόν web-based λογισμικού πελατοκεντρικό, θα πρέπει να παρέχει μια θετική εμπειρία χρήσης στον πελάτη σε όλα τα πιθανά σενάρια χρήσης που θα αυτός πραγματοποιήσει. Είτε θέλει να πραγματοποιήσει μια ηλεκτρονική αγορά, είτε θέλει να γίνει μέλος κάποιας κοινότητας, είτε ακόμα και μια απλή αναζήτηση, θα πρέπει να του παρέχεται ευκολία χρήσης χωρίς ιδιαίτερα προβλήματα. Αν αναλογιστεί κανείς πόσο εύκολο είναι να χαθεί ένας πελάτης ο οποίος αντιμετωπίζει δυσάρεστες εκπλήξεις κατά τη διάρκεια χρήσης ενός web-based λογισμικού, γίνεται ακόμα πιο ξεκάθαρη η ανάγκη της παροχής θετικής εμπειρίας χρήσης.

2.2.1 Η εμπειρία χρήστη (user experience - UX)

Ένας πιο ευρύς ορισμός, ο οποίος όμως δίνεται από το διεθνή οργανισμό προτύπων (International Organization for Standardization), είναι "η απόκριση και η αντίληψη ενός χρήστη που προκύπτουν από τη χρήση ή και την αναμενόμενη χρήση ενός προϊόντος, συστήματος ή υπηρεσίας" [1]. Πιο απλά, η εμπειρία χρήστη έχει να κάνει με το πώς αυτός νοιώθει κατά την αλληλεπίδρασή του με ένα προϊόν, σύστημα ή υπηρεσία τη στιγμή που τα χρησιμοποιεί. Συχνά ο όρος UX συγχέεται με τη χρηστικότητα (usability), η οποία έχει να κάνει με το βαθμό ικανοποίησης κάποιων στόχων που έχει ο χρήστης από τη χρήση ενός προϊόντος ή μιας υπηρεσίας. Η επίτευξη θετικής εμπειρίας χρήστη προκύπτει από αν αν ικανοποιεί τα χαρακτηριστικά που ακολουθούν.

Εύχρηστο (Useful) Θα πρέπει ο χρήστης να μπορεί να χρησιμοποιήσει το προϊόν/υπηρεσία για το λόγο που επιθυμεί.

Εύκολη Ανεύρεση (Findable) Ο χρήστης θα πρέπει να μπορεί να βρίσκει άμεσα την πληροφορία/υπηρεσία που θέλει, όταν τη θέλει.

Χρήσιμο (Usable) Εάν δεν εξυπηρετείται η ανάγκη του χρήστη, τότε αυτός δεν θα ενδιαφερθεί για το προϊόν/υπηρεσία.



Σχήμα 2.2: Η κυψέλη του Peter Morville's για την εμπειρία χρήστη. Κάθε κυψελίδα αντιπροσωπεύει επιμέρους έννοιες που συνθέτουν την εμπειρία χρήση (UX)

Επιθυμητό (Desirable) Το προϊόν/υπηρεσία θα πρέπει να έχει σχεδιαστεί με τέτοιο τρόπο ώστε να είναι ελκυστικό για τον χρήστη.

Να έχει αξία (Valuable) Εάν το προϊόν/υπηρεσία δεν παρέχει αξία στο χρήστη, τότε αυτός θα σταματήσει να το χρησιμοποιεί σύντομα.

Αξιόπιστο (Credible) Ο χρήστης θα πρέπει να αισθάνεται ασφάλεια για να εμπιστευτεί πχ. τα προσωπικά δεδομένα του σε μία web εφαρμογή.

Προσβάσιμο (Accessible) Θα πρέπει να λαμβάνεται μέριμνα για την προσβασιμότητα χρηστών με ειδικές δεξιότητες.

Τα παραπάνω χαρακτηριστικά παρέχουν όχι μόνο ικανοποιητική εμπειρία χρήστη, αλλά αποτελούν και εργαλεία για τον έλεγχο, την περαιτέρω διερεύνηση και μέτρησή της.

Ένας άλλος όρος με τον οποίο συχνά συγχέεται η εμπειρία χρήστη είναι η διεπαφή χρήστη (user interface – UI). Η διεπαφή χρήστη επικεντρώνεται στον τρόπο με τον οποίο μία υπηρεσία, σύστημα ή προϊόν λειτουργεί, καθώς και στην εξωτερική εικόνα του. Περιλαμβάνει έννοιες όπως η σχεδίαση των ροών ενεργειών σε μία ιστοσελίδα και τη σχεδίαση της εικόνας της. Η εμπειρία χρήστη λαμβάνει υπόψη τη διεπαφή χρήστη, ελέγχοντας

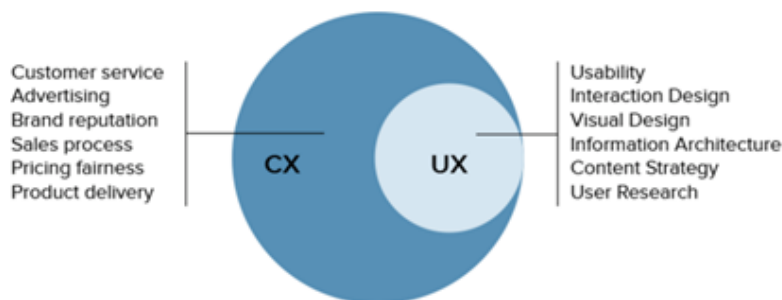
αν συμμορφώνεται με τις έννοιες που δόθηκαν νωρίτερα [Σχήμα 6.8].

2.2.2 Η εμπειρία πελάτη (customer experience - CX)

Η έννοια της εμπειρίας πελάτη περιλαμβάνει κάθε αλληλεπίδραση που έχει ένα άτομο με ένα όλο τα κανάλια (channels) ενός οργανισμού ή brand, συμπεριλαμβανομένου ενός προϊόντος/υπηρεσίας. Είναι μια ομπρέλα δηλαδή, κάτω από την οποία υπάρχουν όλα τα κανάλια και όλα τα προϊόντα/υπηρεσίες του οργανισμού/Brand και τα συναισθήματα που νοιώθουν οι χρήστες/πελάτες γι αυτά [3]. Τυπικά, η εμπειρία πελάτη αναφέρεται στην εξυπηρέτηση πελατών, τη διαφήμιση, τη φήμη του οργανισμού/Brand, τη διαδικασία πωλήσεων και το πόσο δίκαιη είναι αυτή, την παράδοση του προϊόντος και την εμπειρία χρήστη από τη χρήση του προϊόντος αυτού [Σχήμα 2.3].

2.2.3 Σύγκριση εμπειρίας χρήστη (UX) και εμπειρίας πελάτη (CX)

Είναι σύνηθες φαινόμενο οι σχεδιαστές υπηρεσιών/προϊόντων να εστιάζουν στην εμπειρία χρήστη, αλλά θα πρέπει να λαμβάνεται υπόψη και οι έννοιες που έχουν να κάνουν με την εμπειρία πελάτη. Η εμπειρία πελάτη αντιπροσωπεύει κάθε βήμα της διαδικασίας που εκκινά από τη στιγμή που οι χρήστες κάνουν έρευνα αγοράς μέχρι τη στιγμή που δοκιμάζουν το προϊόν/υπηρεσία ή/και προσφεύγουν στην εξυπηρέτηση πελατών εάν δεν ικανοποιούνται οι ανάγκες τους.



Σχήμα 2.3: Η συσχέτιση των εννοιών της εμπειρίας χρήστη με την εμπειρία πελάτη.¹

Οι παραπάνω έννοιες της εμπειρίας χρήστη και εμπειρίας πελάτη μπορούν, συνδυαζόμενες, να οδηγήσουν στη δημιουργία ενός προϊόντος με επίκεντρο τον πελάτη.

2.2.4 Η πελατοκεντρικότητα και η ανάπτυξη web-based λογισμικού

Όπως αναφέρθηκε προηγούμενα, με το πέρασμα των χρόνων κατέστη δυνατό να κατανοηθεί ότι για να είναι ένα προϊόν web-based λογισμικού πελατοκεντρικό, θα πρέπει να παρέχει μια θετική εμπειρία χρήσης στον πελάτη σε όλα τα πιθανά σενάρια χρήσης που θα αυτός πραγματοποιήσει. Είτε θέλει να πραγματοποιήσει μια ηλεκτρονική αγορά, είτε θέλει να γίνει μέλος κάποιας κοινότητας, είτε ακόμα και μια απλή αναζήτηση, θα πρέπει να του παρέχεται ευκολία χρήσης χωρίς ιδιαίτερα προβλήματα. Αν αναλογιστεί κανείς πόσο εύκολο είναι να χαθεί ένας πελάτης ο οποίος αντιμετωπίζει δυσάρεστες εκπλήξεις κατά τη διάρκεια χρήσης ενός web-based λογισμικού, γίνεται ακόμα πιο ξεκάθαρη η ανάγκη της παροχής θετικής εμπειρίας χρήσης.

Με τον όρο εμπειρία χρήστη, όπως ορίζεται στο **ISO 9241-210:2010** [7] είναι:

”η αντίληψη και οι αντιδράσεις που προκύπτουν από τη χρήση ή την προσδοκώμενη χρήση ενός προϊόντος, συστήματος ή υπηρεσίας”

Σύμφωνα με αυτό τον ορισμό, η εμπειρία χρήσης περιλαμβάνει όλα τα συναισθήματα, τις προτιμήσεις, την αντίληψη, τη αντιδράσεις (τόσο φυσικές όσο και ψυχολογικές), τις συμπεριφορές και τα επιτεύγματα που υπάρχουν πριν, κατά τη διάρκεια αλλά και μετά τη χρήση ενός προϊόντος. Σχετίζεται άμεσα με την ευχρηστία και τις διαστάσεις αυτής οι οποίες είναι οι παρακάτω:

1. **Αποδοτικότητα** (Efficiency): Για να ικανοποιείται η αποδοτικότητα, ο πελάτης-χρήστης θα πρέπει να ολοκληρώνει τις εκάστοτε διαδράσεις του με το προϊόν λογισμικού σε λογικά χρονικά πλαίσια. Δηλαδή δεν πρέπει να υπάρχουν καθυστερήσεις που προκαλούνται από πλευράς του προϊόντος κατά τη διάρκεια χρήσης του.
2. **Αποτελεσματικότητα** (Effectiveness): Ορίζει ότι οι ενέργειες του χρήστη ολοκληρώνονται χωρίς προβλήματα.
3. **Ικανοποίηση** (Satisfaction): Ο χρήστης είναι ευχαριστημένος από τη διάδρασή του με το προϊόν.

Η ευχρηστία ορίζει πως ο σχεδιασμός και η υλοποίηση γίνονται με στόχο τον πελάτη και πως η παροχή των απαιτήσεων γίνεται από αυτόν.

Η πελατοκεντρικότητα σχετίζεται με την ενσυναίσθηση προς τον πελάτη, δηλαδή την κατανόηση των αναγκών του και την ενσωμάτωσή τους στην ανάπτυξη του εκάστοτε λογισμικού με στόχο την ικανοποίησή τους. Αυτή η προσέγγιση ορίζει, κατά τη διάρκεια σχεδιασμού και ανάπτυξης του λογισμικού, τα χαρακτηριστικά που αναφέρονται ως σημαντικά από τους χρήστες. Αυτά θα πρέπει να συμπεριλαμβάνονται στο προϊόν και να είναι

έτσι φτιαγμένα, ώστε να είναι εύκολο στους χρήστες να τα κατανοήσουν και να τα χρησιμοποιήσουν. Με αυτόν τον τρόπο η ολοκλήρωση ανάπτυξης του προϊόντος γίνεται ευκολότερη, γρηγορότερη και λιγότερο κοστοβόρα, καθώς οι απαιτήσεις είναι πιο ξεκάθαρες και δεν χρειάζεται πειραματισμός προκειμένου να εντοπιστούν τα σημαντικά εκείνα στοιχεία που θα εξασφαλίσουν την γενική ικανοποίηση του τελικού πελάτη. Με απλά λόγια, η πελατοκεντρική προσέγγιση βοηθάει στο να αναπτύξει κανείς το σωστό προϊόν αλλά και να αναπτύξει το προϊόν σωστά.

Χαρακτηριστικό παράδειγμα, το οποίο αποδεικνύει τη σημασία του να βρίσκεται ο πελάτης στο κέντρο της διαδικασίας δημιουργίας ενός λογισμικού, είναι αυτό της IBM. Η ιστοσελίδα τους δεν λειτουργούσε σωστά και μια γρήγορη ανάλυση έδειξε ότι τα πιο ευρέως χρησιμοποιούμενα στοιχεία της σελίδας τους ήταν η "αναζήτηση" και η "βοήθεια", γεγονός που έκανε ξεκάθαρο ότι οι χρήστες δυσκολεύονταν να βρουν αυτό που χρειάζονταν [8]. Η εταιρεία εστίασε στις ανάγκες των πελατών της και φρόντισε να ανακατασκευάσει την ιστοσελίδα της με τρόπο τέτοιο ώστε να είναι πιο αποδοτική. Σε μια μόλις εβδομάδα η χρήση της "αναζήτησης" και της "βοήθειας" μειώθηκε δραματικά, ενώ οι πωλήσεις αυξήθηκαν κατά 400%.

Στην ερώτηση λοιπόν αν η πελατοκεντρικότητα βοηθάει ουσιαστικά εν' τέλει, η απάντηση είναι ναι. Ένα web-based προϊόν λογισμικού, το οποίο δεν περιέχει υπερβολές, είναι εύκολο στη χρήση και έχει σαν κέντρο τον πελάτη στοχεύοντας στη συνολική ικανοποίησή του, μπορεί να συγκεντρώσει καλύτερη αξιολόγηση η οποία μπορεί να οδηγήσει σε άμεση διαφήμισή του. Λαμπρό παράδειγμα αποτελεί η Google η οποία εδραιώθηκε ως η κυρίαρχη μηχανή αναζήτησης χωρίς ιδιαίτερη διαφήμιση. Από τη στιγμή που δούλευε σωστά και καλύτερα από άλλες μηχανές αναζήτησης, ήταν λογικό όσοι τη χρησιμοποιούσαν να την πρότειναν και στο περίγυρό τους.

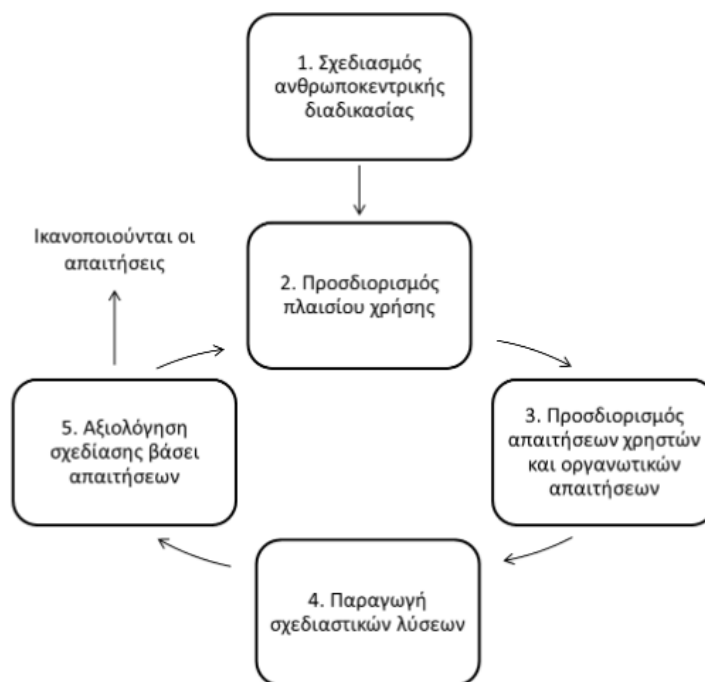
Σε γενικές γραμμές η πελατοκεντρικότητα ορίζοντας τον πελάτη ως τον κεντρικό άξονα σχεδιασμού και δόμησης ενός προϊόντος λογισμικού βοηθάει στην ανάπτυξη σχέσης εμπιστοσύνης [5] καθώς και στη δημιουργία πελατών πριν ακόμα παραδοθεί σε αυτούς το τελικό προϊόν [6],[31]. Τέλος, υπάρχει και άμεσος συσχετισμός μεταξύ της αύξησης της ικανοποίησης του πελάτη και της αύξησης του κέρδους. Σύμφωνα με έρευνες, αυξάνοντας την ικανοποίηση του πελάτη κατά 5% μπορεί να παρατηρηθεί αύξηση στα κέρδη τουλάχιστον κατά 75% [4]. Είναι επομένως επιτακτική ανάγκη η ενσωμάτωση του πελάτη καθ' όλη τη διάρκεια ανάπτυξης και συντήρησης ενός προϊόντος λογισμικού, καθώς στις μέρες μας, όπου ο ανταγωνισμός είναι μόλις ένα κλικ μακριά, δεν πρέπει να χάνεται ούτε ένας πελάτης.

2.3 Μοντέλο ανθρωποκεντρικού σχεδιασμού-ISO 9241-210:2010

Σύμφωνα με το πρότυπο **ISO 9241-210:2010** (human-centred design for interactive systems), μία αναθεωρημένη έκδοση του προτύπου ISO 13407:1999, η διαδικασία ανάπτυξης διαδραστικών συστημάτων περιλαμβάνει τις εξής φάσεις:

1. Σχεδιασμός της ανθρωποκεντρικής διαδικασίας
2. Προσδιορισμός του πλαισίου χρήσης
3. Προσδιορισμός απαιτήσεων χρηστών και οργανωτικών απαιτήσεων
4. Παραγωγή σχεδιαστικών λύσεων
5. Αξιολόγηση σχεδίασης βάσει απαιτήσεων

Η διαδικασία ανάπτυξης εκτελείται επαναληπτικά σύμφωνα με το Σχήμα 2.4



Σχήμα 2.4: Το μοντέλο ανθρωποκεντρικού σχεδιασμού κατά ISO 9241-210:2010

Το πρότυπο αυτό προτείνει τις παρακάτω έξι βασικές αρχές, προκειμένου να διασφαλίζεται ο ανθρωποκεντρικός σχεδιασμός λογισμικού:

1. Ο σχεδιασμός βασίζεται σε μια σαφή κατανόηση των χρηστών, των

εργασιών και του περιβάλλοντος χρήσης. Η καλή εμπειρία χρήσης για κάποιον μπορεί να μην είναι αποδεκτή για κάποιον άλλον, και αυτό πρέπει να το αποδέχεται η ομάδα σχεδιασμού.

2. Οι χρήστες εμπλέκονται σε όλη τη διάρκεια του σχεδιασμού και της ανάπτυξης. Η αρχή αυτή τονίζει ότι είναι σημαντικό να εμπλέκονται οι χρήστες σε όλες τις φάσεις της ανάπτυξης, όχι μόνο στην αρχή ή στο τέλος. Επιπλέον, το πρότυπο υπογραμμίζει ότι η ομάδα σχεδίασης πρέπει να εμπλέκει ενεργά τους χρήστες στο σχεδιασμό, δηλαδή να μην τους παρουσιάζει μία έτοιμη σχεδιαστική λύση αλλά να τους εμπλέκει στη διαδικασία δημιουργίας της. Η συνεχής και ενεργή εμπλοκή του χρήστη αποτελεί κεντρικό σημείο και του Συμμετοχικού Σχεδιασμού (participatory design ή cooperative design), μίας προσέγγισης στην ανάπτυξη συστημάτων με σκανδιναβική προέλευση.
3. Ο σχεδιασμός οδηγείται και βελτιώνεται βάσει ανθρωποκεντρικής αξιολόγησης. Η αξιολόγηση πρέπει να γίνεται σε όλα τα στάδια της ανάπτυξης και όχι μόνο στο τέλος της. Ως εκ τούτου, αξιολογούνται τόσο αρχικά πρωτότυπα και ιδέες, όσο και ολοκληρωμένες εκδόσεις του συστήματος.
4. Η διαδικασία είναι επαναληπτική. Σύμφωνα με το πρότυπο, ο πλέον κατάλληλος σχεδιασμός ενός διαδραστικού συστήματος δεν μπορεί να επιτευχθεί χωρίς να ακολουθηθεί μία επαναληπτική διαδικασία. Πίσω από αυτή την αρχή υπάρχει η άποψη ότι είναι εξαιρετικά δύσκολο, αν όχι αδύνατο, για τους χρήστες να εξηγήσουν τι θέλουν από ένα σύστημα. Έτσι, για να μπορέσει η ομάδα σχεδιασμού να ανακαλύψει τι θέλουν οι άνθρωποι, χρειάζεται να τους δείξει κάτι που μάλλον δεν θέλουν (τον αρχικό σχεδιασμό) και, στη συνέχεια, με βάση τα σχόλια και τη συμπεριφορά τους να προσδιορίσει πώς να τροποποιήσει ή να βελτιώσει τον αρχικό σχεδιασμό. Η αρχή αυτή δεν μπορεί να ικανοποιηθεί με το μοντέλο καταρράκτη, αλλά συμβαδίζει με την προσέγγιση της ευέλικτης ανάπτυξης λογισμικού (agile software development, ASD).
5. Ο σχεδιασμός αφορά το σύνολο της εμπειρίας του χρήστη. Η αρχή αυτή αποτελεί νέα προσθήκη ως προς το αντίστοιχο πρότυπο ISO 13407:1999. Με αυτόν τον τρόπο, το νέο πρότυπο υπογραμμίζει ότι η έννοια της ευχρηστίας είναι ευρύτερη από αυτήν της ευκολίας χρήσης ενός συστήματος και περιλαμβάνει όλες τις αντιληπτικές και συναισθηματικές πτυχές που τυπικά σχετίζονται με την εμπειρία του χρήστη (user experience - UX).

6. Η ομάδα σχεδιασμού περιλαμβάνει διεπιστημονικές δεξιότητες και οπτικές. Δεν αποτελεί καλή πρακτική η δημιουργία μίας ομάδας που αποτελείται από άτομα με ίδιους ή παρόμοιους ρόλους, γνωστικά αντικείμενα, ή δεξιότητες (π.χ. κυρίως γραφίστες, κυρίως προγραμματιστές κλπ.). Αντιθέτως, είναι χρήσιμο να υπάρχει ένα ευρύ φάσμα απόψεων, που συμπεριλαμβάνει, εκτός από τις απόψεις των ειδικών της επικοινωνίας ανθρώπου - υπολογιστή, των προγραμματιστών και των γραφιστών, τις απόψεις των εμπειρογνομόνων προσβασιμότητας, των τελικών χρηστών, των ειδικών του χώρου ή του αντικειμένου, των πωλητών, των ατόμων τεχνικής υποστήριξης, των συγγραφέων τεχνικών κειμένων, των αναλυτών επιχειρήσεων κλπ.

2.4 Πλεονεκτήματα πελατοκεντρικής προσέγγισης έναντι άλλων

Η πελατοκεντρική σχεδίαση είναι πιο στενά συνδεδεμένη με αυτό που είναι γνωστό ως σχεδιασμός με επίκεντρο τον χρήστη, μια προσπάθεια που πρωτοστάτησε στη δεκαετία του 1980 για τα μηχανικά χρήσιμα και χρησιμοποιούμενα συστήματα ηλεκτρονικών υπολογιστών. Η πελατοκεντρική σχεδίαση βασίζεται στο σχεδιασμό με επίκεντρο τον χρήστη, την αντιμετώπιση των ανησυχιών που πηγάζουν πέρα από την ευκολία χρήσης και την ικανοποίηση. Ειδικότερα, επικεντρώνεται στη συγχώνευση των θεμάτων μαρκετινγκ με τα ζητήματα ευχρηστίας. Στο διαδίκτυο είναι πολύ πιο εύκολο να αποκτήσει κανείς ακροατήριο απ' ότι στα παραδοσιακά μέσα, όμως ένας δυσκολότερος στόχος είναι να μετατρέψει κανείς τους web επισκέπτες σε πελάτες και να καταφέρει να τους κάνει να επιστρέφουν. Σε αντίθεση με κάποιον που πουλάει λογισμικό που πωλείται στα ράφια των supermarket (shrinkwrapped) σε έναν πελάτη που το αγοράζει προτού το χρησιμοποιήσει, θέλει κανείς να πείσει τους επισκέπτες της τοποθεσίας web να γίνουν πελάτες, κάνοντας την ίδια στιγμή την πρώτη τους χρήση απολαυστική. Ιδιαίτερη προσοχή πρέπει να δοθεί στους επιχειρηματικούς στόχους, στους στόχους μάρκετινγκ, στους στόχους ευχρηστίας και στους στόχους εμπειρίας του πελάτη. Οι στόχοι αυτοί συχνά αντιτίθενται μεταξύ τους και η ισορροπία ανάμεσά τους μπορεί να βρεθεί μόνο αν είναι κάποιος ενήμερος για όλους τους εξαρχές και ταυτόχρονα. Τα ζητήματα αυτά είναι πολύ πιο αλληλένδετα και πολύ πιο δύσκολα να σχεδιαστούν για το διαδικτυακό από ότι για το shrinkwrapped λογισμικό.

Σχεδιασμός με επίκεντρο την εταιρεία. Ένας τρόπος ανάπτυξης που συνήθιζε να είναι πολύ δημοφιλής ανάμεσα στις εταιρείες Fortune 500 είναι αυτό που αποκαλείται *σχεδιασμός με επίκεντρο την εταιρεία*. Σε αυτή την περίπτωση οι ανάγκες και τα συμφέροντα της εταιρείας κυριαρχούν της δομής και του περιεχομένου της υπηρεσίας. Το αδύναμο σημείο

ωστόσο είναι πως αυτό που οι εταιρείες νομίζουν πως πρέπει να υπάρχει στις web εφαρμογές τους δεν είναι απαραίτητα αυτό που οι πελάτες χρειάζονται ή θέλουν. Υπάρχουν πολλές εφαρμογές/ιστοσελίδες που οργανώνονται σύμφωνα την εσωτερική εταιρική δομή, με αραιή πληροφόρηση σχετικά με τα προϊόντα και τις υπηρεσίες που προσφέρουν. Αυτού του είδους οι εφαρμογές/ιστοσελίδες καλούνται *χλευαστικά φυλλάδια*. Περιέχουν λίγες χρήσιμες πληροφορίες και αγνοούν πλήρως τις μοναδικές δυνατότητες του web ως μέσο. Οι φυλλαδιακού τύπου εφαρμογές/ιστοσελίδες είναι αποδεκτές μόνο αν αποτελούν ένα βραχυπρόθεσμο βήμα προς πιο εξελιγμένες και πιο χρήσιμες ιστοσελίδες. Ένα άλλο παράδειγμα σχεδίασης με επίκεντρο την εταιρεία είναι η χρήση ενός φρασσεολογίου που είναι γνωστό μόνο στα πλαίσια της εταιρείας. Οι εταιρείες σε αυτή την περίπτωση συνήθως έχουν λανθασμένα συμπεράσματα (υπερεκτιμούν δηλαδή) σχετικά με τις γνώσεις των πελατών τους.

Σχεδίαση με επίκεντρο την τεχνολογία. Οι εφαρμογές/ιστοσελίδες που σχεδιάζονται με τις αρχές της σχεδίασης με επίκεντρο την τεχνολογία είναι συχνά κατασκευασμένες με μικρή εκ των προτέρων έρευνα σχετικά με τις ανάγκες της επιχείρησης και τις ανάγκες των πελατών. Οι εφαρμογές/ιστοσελίδες αυτές είναι συνήθως υπερφορτωμένες με animation στοιχεία, ήχο, βίντεο και κινούμενες διαφημίσεις. Το πρόβλημα με αυτή την προσέγγιση είναι πως συχνά καταλήγει στο να είναι η υπηρεσία ερασιτεχνική, δύσχρηστη και τελικά μη επιθυμητή. Οι εφαρμογές/ιστοσελίδες που σχεδιάστηκαν με βάση την τεχνολογία ήταν διάχυτες στις αρχές του web.

Σχεδιασμός με κέντρο τον σχεδιαστή. Ο σχεδιασμός με κέντρο τον σχεδιαστή (εγωκεντρικός σχεδιασμός) είναι ακόμη ένας διαδεδομένος τρόπος σχεδίασης σε ορισμένους κύκλους. Ορισμένες ομάδες σχεδιασμού έχουν βαθιές δημιουργικές παρορμήσεις, εμπνευσμένες από την τεχνική τους ικανότητα, ωστόσο η προσπάθειά τους συνήθως οδηγεί σε εφαρμογές/ιστοσελίδες που μπορεί να είναι αργές στο να φορτώσουν και δύσκολες ως προς τη χρήση, ενώ μπορεί και να μην λειτουργούν σε όλα τα προγράμματα περιήγησης. Ο σχεδιασμός επικεντρωμένος στο σχεδιαστή μπορεί να είναι ιδανικός για καλλιτεχνικού περιεχομένου εφαρμογές, αλλά όχι για το ηλεκτρονικό εμπόριο ή τους ενημερωτικούς ιστότοπους των οποίων η επιβίωση εξαρτάται από τον μεγάλο αριθμό των επαναλαμβανόμενων επισκεπτών.

2.4.1 Τα πλεονεκτήματα της πελατοκεντρικότητας

Στο σχεδιασμό με επίκεντρο την εταιρεία, οι σχεδιαστές δεν λαμβάνουν υπόψη γιατί οι χρήστες θα επισκέπτονται την εταιρική ιστοσελίδα και το τι θα επιθυμούν να κάνουν εκεί. Στον σχεδιασμό με επίκεντρο την τεχνολογία, η τεχνολογία είναι ένα αδιέξοδο και όχι ένα μέσο για την επίτευξη ενός σκοπού. Στο σχεδιασμό με επίκεντρο τον σχεδιαστή, δίνεται μικρότερη σημασία στις ανάγκες των χρηστών σε σχέση με τις δημιουρ-

γικές και εκφραστικές ανάγκες της ομάδας σχεδιασμού. Σε αντίθεση με αυτούς του τρόπους ανάπτυξης λογισμικού, η πελατοκεντρική σχεδίαση, η οποία επικεντρώνεται στους πελάτες και πρωταρχικά στις ανάγκες τους, βλέπει την τεχνολογία ως ένα εργαλείο που λαμβάνει υπόψη τη δύναμη των χρηστών. Οι σχεδιασμοί με επίκεντρο την τεχνολογία, την εταιρία, και τους σχεδιαστές είχαν λόγο ύπαρξης τις πρώτες μέρες του web, όταν ακόμα οι πρακτικές σχεδίασης λογισμικού ήταν σε αρχικό στάδιο. Πλέον, επιτυχημένα και εύκολα προς τη χρήση site όπως το amazon.com, το yahoo.com, το flickr.com και το ebay.com έχουν σχεδιαστεί εξ αρχής για να καλύπτουν τις ανάγκες των πελατών τους. Στις μέρες μας, ο προσεκτικός αφοφυγκρασμός των αναγκών των πελατών που αντανακλάται στην ιστοσελίδα, θα βοηθήσει στο να υπάρξει μια μακροχρόνια επιτυχία.

2.4.2 Μύθοι σχετικά με την πελατοκεντρικότητα

Στη συνέχεια παρατίθεται μία σειρά μύθων σχετικά με την εφαρμογή και χρησιμότητα της πελατοκεντρικότητας στην ανάπτυξη web-based λογισμικού. Οι μύθοι αυτοί καταρρίπτονται, αναδεικνύοντας ταυτόχρονα τη μεγάλη σημασία εφαρμογής της [8]:

Ο καλός σχεδιασμός παίρνει πολύ και κοστίζει υπερβολικά. Ο πελατοκεντρικός σχεδιασμός πράγματι προσθέτει κάποιο επιπλέον κόστος εκ των προτέρων, καθώς θα πρέπει να μιλά κανείς με τους πελάτες, να δημιουργεί πρότυπα, να λαμβάνει ανατροφοδότηση βάση των προτύπων αυτών κοκ. Ωστόσο, η πελατοκεντρική σχεδίαση μπορεί να μειώσει σημαντικά τα back-end κόστη, δηλαδή τις δαπάνες που προκύπτουν ως αποτέλεσμα της ανταπόκρισης στη δυσανεμία των πελατών μέσω της υπερφορτωμένης εξυπηρέτησης πελατών, επιστροφής χρημάτων, συνεχούς εκτεταμένης συντήρησης της εφαρμογής/υπηρεσίας κλπ. Ο πελατοκεντρικός σχεδιασμός μπορεί να μειώσει το συνολικό χρόνο και κόστος ανάπτυξης, καθώς επικεντρώνεται στο να εντοπίζει προβλήματα στα αρχικά στάδια του σχεδιασμού και όταν αυτά είναι ακόμα εύκολο να διορθωθούν, εμποδίζοντάς τα από το να προκαλέσουν σοβαρά, χρονοβόρα και δαπανηρά προβλήματα στο μέλλον.

Οι πελάτες μπορούν πάντοτε να βασίζονται στα εγχειρίδια και τη βοήθεια. Τα εγχειρίδια και η βοήθεια είναι σημαντικά, αλλά οι πελάτες είναι απίθανο να είναι τόσο υπομονετικοί ώστε να διαβάσουν ένα μεγάλο όγκο εγχειριδίων για να χρησιμοποιήσουν μια εφαρμογή/υπηρεσία. Τα εγχειρίδια και η βοήθεια είναι οι τελευταίες λύσεις ενός δυσαρεστημένου πελάτη. Ο πελατοκεντρικός σχεδιασμός παρέχει εργαλεία τέτοια που να μπορεί ο σχεδιαστής να "δει" μέσα από τα μάτια του πελάτη, να καταλάβει την κοσμοθεωρία του και έπειτα να σχεδιάσει μια εφαρμογή/υπηρεσία που να ανταποκρίνεται στις ανάγκες του.

Η έρευνα αγοράς αποκαλύπτει όλα αυτά που χρειάζεται ο πελάτης. Αν και η έρευνα αγοράς είναι ανεκτίμητο εργαλείο, δεν είναι επαρκής

όσον αναφορά την κατανόηση της συμπεριφοράς του πελάτη. Μια λίστα νέων χαρακτηριστικών μπορεί να ικανοποιήσει τους πελάτες που ζήτησαν τα συγκεκριμένα χαρακτηριστικά, αλλά τα ίδια χαρακτηριστικά μπορούν να σταθούν εμπόδιο στο να παραχθεί ικανοποιητική και επιτυχημένη εμπειρία πελάτη για ένα μεγαλύτερο υποσύνολο πελατών. Η έρευνα αγοράς θα πρέπει να είναι ισορροπημένη και να συνοδεύεται από άμεση παρατήρηση. Μια πελατοκεντρικής σχεδίασης ομάδα χρησιμοποιεί μια ποικιλία τεχνικών, από παρατήρηση μέχρι συνεντεύξεις, για να αποσπάσει τις πραγματικές ανάγκες των πελατών και να επικεντρωθεί στους τομείς που θα είναι πιο σημαντικοί για τους περισσότερους πελάτες.

Μέρος II

ΠΑΡΑΔΟΣΙΑΚΕΣ ΜΕΘΟΔΟΙ

ΠΑΡΑΔΟΣΙΑΚΕΣ ΜΕΘΟΔΟΙ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ

3.1 Εισαγωγή

Ηδη από τα τέλη της δεκαετίας του '50 είχε γίνει κατανοητή η ανάγκη ορισμού ενός συνόλου φάσεων βάσει των οποίων θα κινούνταν η ανάπτυξη λογισμικού, με την πρώτη προσπάθεια να γίνεται από τον Herbert D. Benington το 1956 [9]. Έκτοτε, πολλές διαφορετικές μέθοδοι (methods) και μοντέλα διαδικασιών (process models) έχουν αναπτυχθεί και χρησιμοποιηθεί για την παραγωγή ποιοτικού λογισμικού. Οι πιο πολλές από αυτές, που ονομάζονται "βαριές" ή "παραδοσιακές" μέθοδοι είναι προσανατολισμένες στην εκτεταμένη τεκμηρίωση και στις αυστηρά προκαθορισμένες διαδικασίες [10]. Το πλέον γνωστό και εφαρμοσμένο μοντέλο είναι το μοντέλο καταρράκτη, όπως αυτό ορίστηκε το 1970 από τον Winston W. Royce. Το μοντέλο αυτό χαρακτηρίζεται από μια γραμμική μορφή, όπου κάθε φάση του κύκλου ζωής του λογισμικού (Software Development Life Cycle) ολοκληρώνεται πλήρως πριν ξεκινήσει η επόμενη. Οι φάσεις ανάπτυξης του λογισμικού είναι πολλές και μεγάλης διάρκειας, επιβραδύνοντας τον συνολικό χρόνο ανάπτυξης. Ωστόσο, το κυριότερο από τα προβλήματα αυτών των μεθόδων είναι ότι προσχεδιάζουν το μεγαλύτερο κομμάτι της διαδικασίας ανάπτυξης του λογισμικού με μεγάλη λεπτομέρεια και σε βάθος χρόνου, προσπαθώντας να μαντέψουν και να προλάβουν κάθε πρόβλημα που μπορεί να προκύψει. Στο κεφάλαιο αυτό θα γίνει εκτενής αναφορά στο μοντέλο καταρράκτη, καθώς βάσει αυτού κινηθήκαμε

για τη δημιουργία της ηλεκτρονικής πλατφόρμας που θα παρουσιαστεί σε επόμενο κεφάλαιο.

3.1.1 Διαδικασία Παραγωγής Λογισμικού

Η διαδικασία παραγωγής λογισμικού [11] είναι ένα σύνολο δραστηριοτήτων και αντίστοιχων αποτελεσμάτων, που παράγουν τελικά ένα προϊόν λογισμικού. Υπάρχουν τέσσερις θεμελιώδεις δραστηριότητες, κοινές σε όλες τις διαδικασίες παραγωγής λογισμικού. Οι δραστηριότητες αυτές είναι οι παρακάτω:

1. **Η εξαγωγή προδιαγραφών του λογισμικού (software specification)**, κατά την οποία οι πελάτες μαζί με τους μηχανικούς ορίζουν το λογισμικό που θα παραχθεί και τους περιορισμούς της λειτουργίας του.
2. **Η ανάπτυξη του λογισμικού (software development)**, κατά την οποία σχεδιάζεται το λογισμικό και κατασκευάζονται τα σχετικά προγράμματα.
3. **Η επικύρωση του λογισμικού (software validation)**, κατά την οποία το λογισμικό ελέγχεται ώστε να εξασφαλιστεί ότι είναι αυτό που ζητάει ο πελάτης.
4. **Η εξέλιξη του λογισμικού (software evolution)**, κατά την οποία το λογισμικό τροποποιείται ώστε να προσαρμοστεί στις μεταβαλλόμενες απαιτήσεις του πελάτη και της αγοράς.

Κάθε τύπος συστημάτων λογισμικού χρειάζεται και διαφορετικές διαδικασίες ανάπτυξης. Το λογισμικό πραγματικού χρόνου ενός αεροσκάφους για παράδειγμα, πρέπει να έχει προδιαγραφεί πλήρως πριν αρχίσει η ανάπτυξη, ενώ στα συστήματα ηλεκτρονικού εμπορίου οι προδιαγραφές και τα προγράμματα συνήθως αναπτύσσονται μαζί. Κατά συνέπεια, οι παραπάνω γενικές δραστηριότητες μπορεί να οργανώνονται με διαφορετικούς τρόπους και να περιγράφονται σε διαφορετικά επίπεδα λεπτομέρειας, ανάλογα με τον τύπο του παραγόμενου λογισμικού. Από την άλλη πλευρά, η χρήση μιας ακατάλληλης διαδικασίας παραγωγής λογισμικού μπορεί να υποβαθμίσει την ποιότητα ή τη χρησιμότητα του παραγόμενου προϊόντος ή και να αυξήσει το κόστος ανάπτυξης.

3.1.2 Μοντέλο Διαδικασίας Παραγωγής Λογισμικού

Μοντέλο διαδικασίας παραγωγής λογισμικού ονομάζεται μια απλοποιημένη περιγραφή κάποιας διαδικασίας παραγωγής λογισμικού, η οποία παρουσιάζει μια συγκεκριμένη άποψη της διαδικασίας αυτής. Ένα μοντέλο διαδικασίας μπορεί να περιλαμβάνει δραστηριότητες που αποτελούν μέρος της διαδικασίας παραγωγής λογισμικού, προϊόντα λογισμικού, και τους ρόλους των ατόμων που συμμετέχουν στο έργο λογισμικού. Παραδείγματα

μερικών τύπων μοντέλων διαδικασιών παραγωγής λογισμικού που μπορεί να δημιουργηθούν είναι τα εξής [13]:

1. **Μοντέλο ροής εργασιών (workflow)**. Δείχνει την ακολουθία των δραστηριοτήτων της διαδικασίας, καθώς και τις εισόδους, τις εξόδους, και τις εξαρτήσεις αυτών των δραστηριοτήτων. Σε ένα τέτοιο μοντέλο οι δραστηριότητες αντιπροσωπεύουν ανθρώπινες πράξεις.
2. **Μοντέλο ροής δεδομένων (dataflow) ή δραστηριοτήτων (activity)**. Αναπαριστά τη διαδικασία ως σύνολο δραστηριοτήτων, κάθε μία από τις οποίες πραγματοποιεί κάποιο σχηματισμό δεδομένων. Ένα τέτοιο μοντέλο δείχνει με ποιόν τρόπο τα δεδομένα εισόδου της διαδικασίας, όπως ένα σύνολο προδιαγραφών, μετασχηματίζονται σε δεδομένα εξόδου, όπως ένα σχέδιο. Οι δραστηριότητες εδώ αντιπροσωπεύουν μετασχηματισμούς που πραγματοποιούνται από ανθρώπους ή από υπολογιστές.
3. **Μοντέλο ρόλων/ενεργειών (role/action)**. Αναπαριστά τους ρόλους των ανθρώπων που συμμετέχουν στη διαδικασία παραγωγής λογισμικού, καθώς και τις δραστηριότητες για τις οποίες είναι υπεύθυνοι.

Τα περισσότερα μοντέλα διαδικασιών παραγωγής λογισμικού βασίζονται στο ένα από τα τρία γενικά μοντέλα ή υποδείγματα ανάπτυξης λογισμικού:

1. **Προσέγγιση καταρράκτη (waterfall)**. Η προσέγγιση αυτή παίρνει τις παραπάνω δραστηριότητες και τις αναπαριστά ως ξεχωριστές φάσεις της διαδικασίας, όπως εξαγωγή προδιαγραφών απαιτήσεων, σχεδιασμός του λογισμικού, υλοποίησης, δοκιμές κοκ. Μόλις οριστεί ένα στάδιο, "κλείνει" και η ανάπτυξη προχωρά στο επόμενο στάδιο.
2. **Επαναληπτική ανάπτυξη (iterative)**. Η προσέγγιση αυτή διαπλέκει τις δραστηριότητες της εξαγωγής προδιαγραφών, της ανάπτυξης και της επικύρωσης. Αναπτύσσεται γρήγορα ένα αρχικό σύστημα από πολύ αφαιρετικές προδιαγραφές, το οποίο στη συνέχεια βελτιώνεται σύμφωνα με τα σχόλια του πελάτη, μέχρι να καταλήξει σε ένα σύστημα που ικανοποιεί τις ανάγκες του πελάτη. Τότε το σύστημα μπορεί να παραδοθεί ως έχει ή εναλλακτικά να υλοποιηθεί ξανά με μια πιο δομημένη προσέγγιση ώστε να παραχθεί ένα πιο συντηρήσιμο σύστημα.
3. **Τεχνολογία λογισμικού βάσει συστατικών στοιχείων (component - based software engineering CBSE)**. Η τεχνική αυτή προϋποθέτει ότι τα συστατικά στοιχεία του συστήματος υπάρχουν ήδη. Η διαδικασία ανάπτυξης του συστήματος εστιάζει στην ολοκλήρωση αυτών των υφιστάμενων μερών, αντί για την εξαρχής ανάπτυξή τους.

3.1.3 Σημασία εφαρμογής μοντέλου διαδικασίας παραγωγής λογισμικού

Η διαδικασία παραγωγής λογισμικού, όπως είδαμε και παραπάνω, αποτελείται από ένα σύνολο ενεργειών που θα πρέπει να γίνουν προκειμένου να παραχθεί το τελικό προϊόν. Όμως η σειρά και ο τρόπος με την οποία αυτές θα πραγματοποιηθούν δείχνει να έχει μεγάλη σημασία, καθώς χωρίς την ύπαρξη συγκεκριμένου πλάνου μπορεί εύκολα να δημιουργηθεί χάος. Συχνά πολλά έργα ακολουθούν απλά μια διαδικασία η οποία απαρτίζεται από αποφάσεις που έχουν ληφθεί γρήγορα, χωρίς ιδιαίτερη ανάλυση των κινδύνων και των απαιτήσεων. Αυτό σε βάθος χρόνου, και ιδίως στην περίπτωση που το εκάστοτε έργο απαιτεί προσαρμοστικότητα και υπάρχει ανάγκη για προσθήκη νέων χαρακτηριστικών, κάνει τη διαδικασία περίπλοκη με τα λάθη να κυριαρχούν και να είναι δύσκολο να αναιρεθούν.

Η μη εφαρμογή κάποιας μεθόδου, ή όπως επίσημα ορίζεται, η *ad-hoc* ανάπτυξη λογισμικού, μπορεί να λειτουργήσει μόνο υπό την προϋπόθεση ότι είναι εξαρχής ορισμένες και ξεκάθαρες οι απαιτήσεις τόσο του τελικού πελάτη/χρήστη, όσο και των εργαλείων που πρέπει να χρησιμοποιηθούν. Σε διαφορετική περίπτωση ο κίνδυνος να οδηγηθεί η όποια προσπάθεια σε ναυάγιο είναι μεγάλος [14].

Υπάρχει πληθώρα μεθόδων για ανάπτυξη λογισμικού που χρησιμοποιούνται τις τελευταίες δεκαετίες. Παρ' όλ' αυτά, είναι εξίσου σημαντικό να επιλεχθεί από αυτές εκείνη η οποία θα είναι καταλληλότερη για την κάθε περίπτωση, καθώς η λάθος επιλογή μπορεί να οδηγήσει στα ίδια αποτελέσματα με τη μη εφαρμογή κάποιας μεθόδου. Η επιλογή της μεθόδου θα πρέπει να γίνεται βάσει κριτηρίων όπως ο οικονομικός προϋπολογισμός, το μέγεθος της ομάδας σχεδίασης, το γνωσιακό υπόβαθρο των μελών, των τεχνολογιών και των εργαλείων που θα χρησιμοποιηθούν. Η επιλογή λάθος μεθόδου μπορεί να οδηγήσει σε καθυστέρηση της διανομής του τελικού λογισμικού, σε αποτυχία ίσως αυτού και πολύ πιθανόν σε δυσαρέσκεια του τελικού πελάτη/χρήστη.

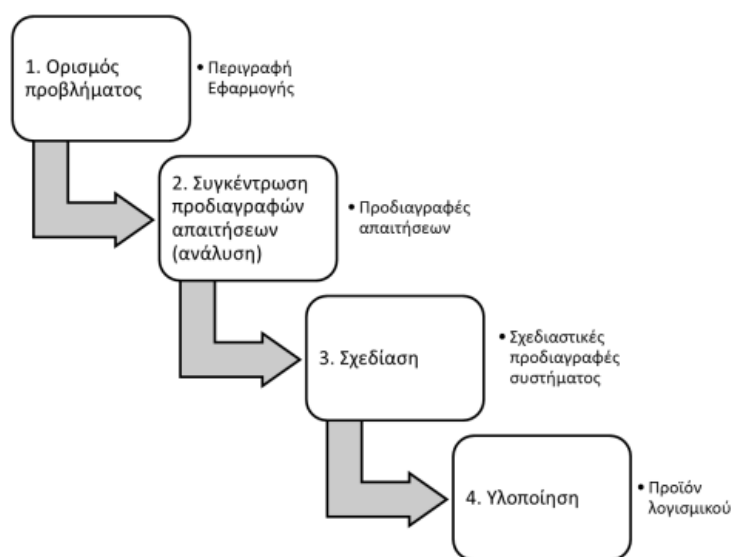
3.2 Παραδοσιακές Μέθοδοι Ανάπτυξης Λογισμικού

Οι παραδοσιακές μέθοδοι, οι οποίες έχουν χαρακτηριστεί και ως "βαριές μέθοδοι", χρησιμοποιήθηκαν για πολλά χρόνια για την ανάπτυξη λογισμικού. Βασίζονται σε μια ακολουθία σειριακών βημάτων όπως η ανάλυση των απαιτήσεων, ο σχεδιασμός της λύσης και η παράδοση του προϊόντος λογισμικού. Οι μέθοδοι αυτοί απαιτούν ορισμό, καταγραφή και έγγραφη τεκμηρίωση του συνόλου των απαιτήσεων από την αρχή του εκάστοτε έργου. Υπάρχει μια πλειάδα μεθόδων που χαρακτηρίζονται ως παραδοσιακές, με τις κυριότερες από αυτές να είναι το μοντέλο του καταρράκτη (waterfall), το σπειροειδές μοντέλο (spiral). Στο κεφάλαιο όμως αυτό θα

εστιάζουμε στην πιο γνωστή και ευρέως χρησιμοποιούμενη εξ' αυτών, το μοντέλο του καταρράκτη.

3.2.1 Μοντέλο Καταρράκτη (Waterfall)

Για πολλά χρόνια η διαδικασία ανάπτυξης λογισμικού είχε τη μορφή μιας αλληλουχίας διαδοχικών διακριτών βημάτων. Η σειριακή αυτή ακολουθία είναι γνωστή ως το **μοντέλο καταρράκτη (waterfall model)**. Η πρώτη επίσημη αναφορά στο μοντέλο του καταρράκτη έγινε το 1970 σε άρθρο του Winston W. Royce, όπου ωστόσο δεν αναφερόταν πουθενά ο όρος "καταρράκτης" [12]. Η κρίση λογισμικού το 1960 έφερε στην επιφάνεια την διαπίστωση πώς η διαδικασία παραγωγής λογισμικού μοιάζει με άλλες τεχνολογικές διαδικασίες. Για το λόγο αυτό, το μοντέλο του καταρράκτη μοιάζει με άλλες τεχνολογικές διαδικασίες [17]. Το μοντέλο καταρράκτη αποτελεί μια προσέγγιση που ακολουθεί διαδοχικά τα βήματα που την αποτελούν και κατά την οποία η ανάπτυξη γίνεται προς τα κάτω (όπως και σε ένα καταρράκτη, εξ' ου και η ονομασία). Όπως δίνεται στο Σχήμα 3.1, τα στάδια από τα οποία διέρχεται το μοντέλο αυτό είναι τα εξής:



Σχήμα 3.1: Διακριτά στάδια του μοντέλου καταρράκτη

- Στο αρχικό στάδιο γίνεται η γενική περιγραφή του προβλήματος και της λύσης που προτείνεται να δοθεί. Το αποτέλεσμα αυτής της φάσης είναι η αρχική "Περιγραφή της Εφαρμογής" (Application Description).
- Στο δεύτερο στάδιο, που είναι επίσης γνωστό ως ανάλυση του προβλήματος, συγκεντρώνονται και καταγράφονται οι απαιτήσεις της

εφαρμογής, οι οποίες παίρνουν τελικά τη μορφή ενός επίσημου εγγράφου των "Προδιαγραφών Απαιτήσεων" (Requirements Specifications). Βάσει του εγγράφου αυτού γίνεται η σύνταξη συμβολαίου υποχρεώσεων μεταξύ αυτών που ορίζουν το αρχικό πρόβλημα και αυτών που αναλαμβάνουν την ανάπτυξη της εφαρμογής.

- Με βάση το συμβόλαιο αυτό αρχίζει η επόμενη φάση του σχεδιασμού του προϊόντος, η οποία καταλήγει στις "Σχεδιαστικές Προδιαγραφές" (Design Specifications) που αποτελούν λεπτομερή περιγραφή του τελικού προϊόντος, επαρκή για την ανάπτυξή του.
- Τέλος, με βάση τις "Σχεδιαστικές Προδιαγραφές" γίνεται η ανάπτυξη του προϊόντος, που είναι και η τελευταία φάση της διαδικασίας. Ειδικότερα, η φάση αυτή αφορά την ανάπτυξη του κώδικα εφαρμογής, των βάσεων δεδομένων, συγγραφή εγχειριδίων κλπ., το αποτέλεσμα της δε, είναι το "Τελικό Προϊόν" (Final Product).

Κάθε ένα από τα στάδια αυτού του μοντέλου ανάπτυξης ολοκληρώνονται με έλεγχο που εστιάζει

1. στο κατά πόσον η αναπαράσταση είναι σύμφωνη με τις απαιτήσεις των πελατών, και
2. στο αν η αναπαράσταση είναι συνεπής με τις προδιαγραφές της προηγούμενης φάσης.

Τέλος, μετά την παράδοση του προϊόντος λογισμικού στο χρήστη το προϊόν μπαίνει σε φάση συντήρησης, η οποία περιλαμβάνει αλλαγές στο σύστημα που μπορεί να εντοπισθούν έπειτα από την χρήση του λογισμικού, εντοπισμό και διόρθωση λαθών, βελτίωση λειτουργιών ή ακόμα και προσθήκη νέων λειτουργιών εφόσον εντοπιστούν νέες απαιτήσεις.

Στο μοντέλο του καταρράκτη παρατηρούμε ότι κάθε φάση ανάπτυξης είναι διακριτή από τις υπόλοιπες και ότι η μετάβαση από τη μία φάση στην άλλη γίνεται μέσω εγγράφων, τα οποία συνιστούν διαδοχικές, αυξανόμενης λεπτομέρειας αναπαραστάσεις της εφαρμογής που καταλήγουν στο τελικό προϊόν. Το μοντέλο καταρράκτη έχει το πλεονέκτημα ότι περιγράφει με σαφήνεια τις φάσεις ανάπτυξης ενός προϊόντος, τις οποίες είναι αναπόφευκτο να συναντήσει κανείς σε οποιαδήποτε διαδικασία σχεδιασμού. Επίσης, αποτελεί μία εύκολα κατανοητή διαδικασία από όλους τους εμπλεκόμενους και φαινομενικά διευκολύνει τη διαχείριση του έργου (χρονοδιαγράμματα, παραδοτέα, ανάθεση εργασιών βάσει ειδικότητας). Ωστόσο, η αυστηρή τήρηση του μοντέλου παρουσιάζει προβλήματα. Το πιο σπουδαίο από τα προβλήματα αυτά είναι η αδυναμία λεπτομερούς προδιαγραφής ενός προϊόντος πριν από την υλοποίησή του. Στην πράξη, έχει αποδειχθεί ότι η ανάπτυξη λεπτομερών προδιαγραφών έχει συχνά ως συνέπεια την αναγκαστική τροποποίηση των απαιτήσεων που έχουν ήδη

συνταχθεί και η υλοποίηση του προϊόντος έχει ως συνέπεια να γίνονται αναγκαστικές εκ των υστέρων τροποποιήσεις των λεπτομερών προδιαγραφών. Το πρόβλημα αυτό μπορεί να αντιμετωπιστεί με την ανάπτυξη πρωτοτύπων κατά τη φάση ανάλυσης και σχεδιασμού, δηλαδή πρόχειρων, μικρών σε κλίμακα, εκδόσεων του προϊόντος που αποκαλύπτουν λεπτομέρειες του τελικού συστήματος. Η χρήση πρωτοτύπων συνεπάγεται όμως τον ασαφή διαχωρισμό των φάσεων του μοντέλου, αφού η φάση υλοποίησης εμπλέκεται με τις φάσεις ανάλυσης και σχεδιασμού. Συνέπεια αυτής της διάχυσης των φάσεων είναι να παρουσιάζονται προβλήματα στη διαχείριση της διαδικασίας και στον έλεγχο προόδου του έργου.

Δυνάμεις και αδυναμίες

Το μοντέλο αυτό είναι ιδανικό για να υποστηρίξει λιγότερο πεπειραμένες ομάδες και διευθυντές ή ομάδες των οποίων σύνθεση κυμαίνεται. Η τακτική σειρά των βημάτων ανάπτυξης, οι αυστηροί έλεγχοι για την εξασφάλιση της επαρκούς τεκμηρίωσης και οι αναθεωρήσεις του σχεδίου εξασφαλίζουν την ποιότητα, την αξιοπιστία και τη δυνατότητα συντήρησης του λογισμικού. Ένα επιπλέον πλεονέκτημα του μοντέλου είναι ότι η πρόοδος της ανάπτυξης των συστημάτων είναι μετρήσιμη. Τέλος, ο εκτενής προγραμματισμός του υπό ανάπτυξη λογισμικού βοηθά στην βέλτιστη χρήση των πόρων.

Αντίθετα, το μοντέλο του καταρράκτη χαρακτηρίζεται ως άκαμπτο, αργό, δαπανηρό και δυσκίνητο λόγω της γραμμικής δομής και των αυστηρών ελέγχων. Το πρόγραμμα προχωρά προς τα εμπρός, με πιθανή μόνο τη μικρή μετακίνηση προς τα πίσω. Στο εν λόγω πρότυπο υπάρχει μικρό περιθώριο για επαναληπτική χρήση και επαναχρησιμοποίηση του λογισμικού. Ακόμη, πολλά εξαρτώνται από τον πρόωρο προσδιορισμό και καταγραφή των απαιτήσεων, όμως οι χρήστες μπορεί να μην είναι σε θέση να καθορίσουν σαφώς τι χρειάζονται τόσο νωρίς στην εξέλιξη του έργου. Οι ασάφειες των προδιαγραφών, οι τυχόν ελλείψεις σε τμήματα των συστημάτων και οι απροσδόκητες ανάγκες σε υποδομές ανακαλύπτονται συχνά κατά τη διάρκεια της υλοποίησης κώδικα. Τα προβλήματα συχνά δεν ανακαλύπτονται μέχρι τη δοκιμή του τελικού προϊόντος. Η απόδοση του δεν μπορεί να εξεταστεί έως ότου υλοποιηθεί σχεδόν πλήρως, οπότε και είναι δύσκολο να διορθωθεί. Επιπλέον, το μοντέλο του καταρράκτη είναι δύσκολο να ανταποκριθεί στις αλλαγές. Οι αλλαγές που εμφανίζονται αργά στον κύκλο ζωής του προγράμματος είναι δαπανηρές και η ενσωμάτωσή τους είναι αποθαρρυντική. Το πρότυπο δε, παράγει υπερβολική τεκμηρίωση και η συνεχής ενημέρωσή της είναι επίπονη και χρονοβόρα. Τέλος, πολλές φορές οι γραπτές προδιαγραφές είναι δύσκολο να κατανοηθούν και να αξιολογηθούν από τους χρήστες του προγράμματος, γεγονός που δημιουργεί χάσμα μεταξύ των υπευθύνων ανάπτυξης και των τελικών χρηστών.

3.2.2 Γενικά Χαρακτηριστικά Παραδοσιακών Μεθόδων

Οι παραδοσιακές μέθοδοι, όπως είδαμε και προηγούμενα, ορίζουν μια πειθαρχημένη διαδικασία με σκοπό η διαδικασία ανάπτυξης λογισμικού να βασίζεται στην πρόβλεψη και προκειμένου να είναι αποδοτικό το τελικό προϊόν. Η συλλογή των απαιτήσεων και ο σχεδιασμός έχουν χαρακτηριστικά ολοκληρωτικό και λαμβάνουν χώρα σε πολύ αρχικό στάδιο, με αποτέλεσμα τη μη παροχή δυνατότητας αναπροσαρμογής σε μεταγενέστερες φάσεις της ανάπτυξης του προϊόντος. Έχουν μάλιστα χαρακτηριστεί ως γραφειοκρατικές, καθώς καθυστερούν ιδιαίτερα τη φάση υλοποίησης του λογισμικού εξαιτίας των προαπαιτούμενων που ορίζουν πριν αυτή ξεκινήσει [15]. Οι παραδοσιακές μέθοδοι σε γενικές γραμμές διέπονται από τα παρακάτω χαρακτηριστικά:

Πλانو-κεντρική (Plan oriented): Οι μέθοδοι αυτές στοχεύουν στην δημιουργία ενός πλάνου εκ των προτέρων το οποίο και ακολουθείται ευλαβικά μέχρι και την διάθεση του προϊόντος στον τελικό χρήστη.

Προβλεπτική Προσέγγιση (Predictive Approach): Το χαρακτηριστικό αυτό ορίζει την τάση που έχουν οι "βαριές" μέθοδοι να οργανώνουν και να φτιάχνουν ένα λεπτομερές πλάνο σχετικά με τη διαδικασία ανάπτυξης λογισμικού, το οποίο διαρκεί για μεγάλο χρονικό διάστημα. Η προσέγγιση αυτή γίνεται με τρόπο προβλεπτικό και επαναλαμβανόμενο μέχρι να καταλήξουν στη τελική δομή, η οποία αφορά τόσο το προϊόν όσο και την διαχείριση του έργου από πλευράς κόστους και ανθρώπινου δυναμικού. Δίνεται έμφαση στη σχεδίαση του προϊόντος με βάση τις ανάγκες και στο πώς αυτές οι ανάγκες θα καλυφθούν αποδοτικά. Η υλοποίηση θα αρχίσει να γίνεται όταν η διαδικασία αυτή ολοκληρωθεί πλήρως, και εννοείται πως η υλοποίηση θα ακολουθήσει ακριβώς τον σχεδιασμό που ορίστηκε.

Έγγραφή Τεκμηρίωση (Documentation): Η παραδοσιακή ανάπτυξη λογισμικού ορίζει τα έγγραφα των απαιτήσεων ως βασικό κομμάτι της τεκμηρίωσης. Κύριο στοιχείο στις μεθόδους αυτές είναι η πεποίθηση πως είναι δυνατό να συγκεντρωθούν όλες οι απαιτήσεις του πελάτη, εκ των προτέρων, πριν από τη σύνταξη του κώδικα. Και πάλι, αυτή η προσέγγιση αποτελεί επιτυχία στον κλάδο της μηχανικής, που την καθιστά ελκυστική για τη βιομηχανία λογισμικού. Το να συγκεντρωθούν όλες οι απαιτήσεις, να πάρουν ανατροφοδότηση από τον πελάτη και στη συνέχεια να διαταχτούν οι διαδικασίες (περισσότερη τεκμηρίωση) για τον περιορισμό και τον έλεγχο όλων των αλλαγών δίνουν όντως στο έργο ένα όριο προβλεψιμότητας. Η προβλεψιμότητα είναι πολύ σημαντική σε έργα λογισμικού που είναι ζωτικής κρισιμότητας.

Προσανατολισμένη στη Διαδικασία (Process Oriented): Ο στόχος

των βαριών μεθοδολογιών είναι να καθορίσουν μια διαδικασία που θα λειτουργήσει καλά για οποιονδήποτε τυχαίνει να τη χρησιμοποιήσει. Η διαδικασία θα αποτελείται από ορισμένα καθήκοντα που θα πρέπει να εκτελούνται από τους διαχειριστές, σχεδιαστές, δοκιμαστές κλπ.. Για κάθε μία από αυτές τις εργασίες υπάρχει πάντα μια καλά καθορισμένη διαδικασία.

Εργαλειο-κεντρική (Tool Oriented): Ορίζεται στις μεθόδους αυτές η χρήση ή ακόμα και η υλοποίηση εργαλείων, πριν τη διαδικασία υλοποίησης του προϊόντος λογισμικού, τα οποία θα διευκολύνουν την παραγωγή και θα πρέπει να χρησιμοποιούνται καθ' όλη τη διάρκεια του κύκλου ζωής του έργου.

3.3 Αξιολόγηση Παραδοσιακών μεθόδων

3.3.1 Δυνάμεις και αδυναμίες

Οι παραδοσιακές μέθοδοι δίνουν έμφαση στην τεκμηρίωση, τις τυποποιημένες διαδικασίες και την ακριβή καταγραφή των προδιαγραφών και απαιτήσεων από την αρχή του έργου. Προτιμούνται από εταιρείες που επιθυμούν ανάπτυξη με χαμηλό ρίσκο και που ο χρόνος παράδοσης του έργου δεν αποτελεί προτεραιότητα. Τέτοια έργα θα μπορούσαν να έχουν να κάνουν με ATMs, συστήματα επιτήρησης ενέργειας, συστήματα ERP. Μια άλλη κατηγορία συστημάτων όπου είναι πιο φρόνιμη η χρήση παραδοσιακών μεθόδων είναι αυτά που αναμένεται να έχουν πολυετή λειτουργία χωρίς αλλαγές στον τρόπο λειτουργίας τους καθ' όλη τη διάρκεια χρήσης τους. Ο Boehm [10] αναφέρει ότι τα έργα στα οποία είναι καλύτερο να χρησιμοποιηθούν παραδοσιακές μέθοδοι ανάπτυξης είναι αυτά που δίνουν έμφαση στην αποδοτικότητα, την προβλεψιμότητα και την επίτευξη ενός στόχου μέσα από μία σαφώς ορισμένη διαδικασία [16].

Η χρήση παραδοσιακών μεθόδων ανάπτυξης λογισμικού ενδείκνυται επομένως για έργα που είναι μεγάλης έκτασης, δαπανηρά και πολύπλοκα. Τα έργα αυτά έχουν εξ' αρχής σαφείς στόχους, σαφή λύση και οι απαιτήσεις είναι σαφώς προκαθορισμένες και αμετάβλητες καθ' όλη τη διάρκεια του κύκλου ζωής του συστήματος. Επιπλέον, έχουν να κάνουν με μεγάλο αριθμό χρηστών, με μεγάλο βαθμό αλληλοεξάρτησης, ενώ η σύνθεση της ομάδας σχεδιασμού και υλοποίησης παραμένει σταθερή καθ' όλη της διάρκεια του έργου και επιτηρείται από άκρως εξειδικευμένους και έμπειρους υπευθύνους. Συνήθως η ελαχιστοποίηση κατανάλωσης πόρων δεν αποτελεί προτεραιότητα στα έργα αυτά, ενώ δεν είναι απαραίτητη η συνεχής επικοινωνία και αλληλεπίδραση με τον τελικό χρήστη/πελάτη για την ανάπτυξη του λογισμικού [15].

Από την άλλη πλευρά, οι παραδοσιακές μέθοδοι συχνά περιγράφονται ως ακατάλληλες για γρήγορη ανάπτυξη λογισμικού. Αποκαλούνται

επίσης εκ φύσεως γραφειοκρατικές και δύσκαμπτες [19]. Η εμμονή τους στην λεπτομέρεια, μέσω τεκμηρίωσης σε όλες τις φάσεις ανάπτυξης, οδηγεί συχνά σε υπερβάσεις προϋπολογισμού και σε μεγαλύτερους χρόνους παράδοσης. Δεν είναι κατάλληλες για εφαρμογή σε μικρές ομάδες ανάπτυξης και δεν δίνουν έμφαση στην επικοινωνία και τη συνεργατικότητα μεταξύ ομάδων ανάπτυξης.

Βασικό αρνητικό τους σημείο είναι δυσκολία προσαρμογής τους στην αλλαγή. Η δυνατότητα προσαρμογής σε αλλαγές συχνά αποτελεί καθοριστικό παράγοντα στην επιτυχία ή αποτυχία ενός έργου ανάπτυξης λογισμικού. Η αλλαγή μπορεί να αφορά τις προδιαγραφές, τον τρόπο και το χρόνο ανάπτυξης και κάποιο ή κάποια χαρακτηριστικά της υπηρεσίας που είναι ήδη σε φάση ανάπτυξης [18], [19].

Ο τρόπος χειρισμού της πολυπλοκότητας καθιστά τις παραδοσιακές μεθόδους λιγότερο ελκυστικές. Η προσέγγιση του να σχεδιάζει κανείς όλη την πορεία ενός έργου και μετά να ακολουθεί κατά γράμμα το σχέδιο μπορεί να είναι αποδοτική σε σταθερά και λιγότερο πολύπλοκα περιβάλλοντα, αλλά όχι για περισσότερο πολύπλοκα περιβάλλοντα. Επιπλέον, η εφαρμογή δύσκαμπτων κανόνων και διαδικασιών στη σχεδίαση και ανάπτυξη κώδικα, η προσκόλληση στην ακριβή τήρησή τους και η αντιμετώπιση των σχεδιαστών σαν προβλέψιμης συμπεριφοράς τμήματα του έργου οδηγεί συχνά τις ομάδες σχεδιασμού σε αποτελέσματα χειρότερα των αναμενόμενων, καθώς επιβάλλει ένα συγκεκριμένο τρόπο σκέψης στους σχεδιαστές. Με αυτό τον τρόπο χάνεται, για παράδειγμα, η δυνατότητα υιοθέτησης μιας πιο έξυπνης, γρήγορης λύσης σε ένα επιμέρους πρόβλημα από την υποδειχθείσα.

Σε σχέση με τη γραφειοκρατική χροιά των παραδοσιακών μεθόδων, αποδεικνύεται από έρευνες [18] ότι ελάχιστοι χρήστες χρησιμοποιούν τελικά τις λεπτομερείς αναφορές, τεκμηρίωση και εγχειρίδια που συνοδεύουν το προϊόν. Επιπλέον, οι μέθοδοι αυτές λαμβάνουν επανατροφοδότηση από τους χρήστες μόνο κατά το τελικό στάδιο του κύκλου ζωής ενός έργου, με ό,τι κινδύνους αυτό συνεπάγεται σε σχέση την εμπειρία χρήστη ή/και πελάτη.

Κλείνοντας, οι παραδοσιακοί μέθοδοι θεωρούν βασικά συστατικά ενός επιτυχημένου έργου την παράδοσή του εντός χρονικών ορίων και προϋπολογισμού. Η θεώρηση αυτή δεν λαμβάνει υπόψη το καθολικά, σήμερα πια, ζητούμενο της ικανοποίησης των χρηστών [18].

3.3.2 Η σχέση τους με την πελατοκεντρικότητα

Σε σχέση με την έννοια της πελατοκεντρικότητας οι βαριές μέθοδοι τείνουν να υστερούν στον τρόπο που αυτή εφαρμόζεται καθώς, όπως είδαμε και παραπάνω, οι μέθοδοι αυτές εστιάζουν κατά τη διάρκεια ανάπτυξης πιο πολύ στις διαδικασίες παρά στους τελικούς πελάτες (plan oriented,

process oriented). Η συμμετοχή των τελικών πελατών είναι σημαντική αλλά περιλαμβάνεται κυρίως στα αρχικά στάδια, όπου και γίνεται η συλλογή των απαιτήσεων, και στο τέλος της διαδικασίας ανάπτυξης του λογισμικού, όπου και το τελικό προϊόν τους έχει παραδοθεί [20]. Εκεί πλέον δίνεται στον τελικό πελάτη η δυνατότητα να αξιολογήσει και να κρίνει εάν το τελικό προϊόν είναι αυτό που ζήτησε και αν φυσικά καλύπτει τις ανάγκες του και ικανοποιεί της απαιτήσεις του. Καθώς δεν υπάρχει διάδραση με τον πελάτη κατά τη διάρκεια του σταδίου της υλοποίησης, είναι δύσκολο για εκείνον να κατανοήσει μόνο μέσα από μια σχεδιαστική ή έγγραφη παρουσίαση τη λειτουργικότητα του τελικού προϊόντος. Επίσης, εφόσον το προϊόν του παραδίδεται ολοκληρωμένο, οι όποιες αλλαγές των απαιτήσεων του έχουν προκύψει κατά την υλοποίηση του λογισμικού δεν μπορούν να ενσωματωθούν, παρά μόνο μετά την ολοκλήρωσή του. Ταυτόχρονα, η συμμετοχή του χρήστη μόνο στα αρχικά και τελικά στάδια ανάπτυξης δεν βοηθά στη δημιουργία πελατών, ούτε στη δόμηση μιας σχέσης εμπιστοσύνης.

3.3.3 Αξιολόγηση καταλληλότητας χρήσης τους σε ανάπτυξη Web 2.0 λογισμικού

Όλα τα παραπάνω μας δείχνουν πως οι παραδοσιακές μέθοδοι τείνουν να μην είναι ιδιαίτερα κατάλληλες για την ανάπτυξη Web 2.0 λογισμικού. Οι web 2.0 εφαρμογές ξεφεύγουν από τα απλά όρια μια στατικής ιστοσελίδας και παρέχουν τη δυνατότητα στον τελικό χρήστη να χειρίζεται λογισμικό το οποίο αποτελεί συνδυασμό τέχνης και τεχνολογίας [21]. Χαρακτηρίζονται από την ανάγκη για συνεχή πρόοδο και ανάπτυξη (Continuous Evolution), για χρονικά σύντομο πλάνο ανάπτυξης λογισμικού (Short Development Schedule), για χρήση έτοιμων μικρότερων τμημάτων κώδικα (Modularity) και για συντηρησιμότητα (Maintainability) [22]. Στόχος είναι η άμεση προσαρμογή στις αλλαγές, προκειμένου να επιτυγχάνεται όσο το δυνατόν πιο θετική εμπειρία χρήστη. Από τη στιγμή που η φύση τέτοιου είδους λογισμικού χαρακτηρίζεται από συνεχώς μεταβαλλόμενες απαιτήσεις, με γρήγορο ρυθμό, η χρήση τέτοιων μεθόδων φαντάζει ασύμβατη. Εφόσον η προσέγγιση αυτών των μεθόδων ορίζει παράδοση του προϊόντος μετά την πλήρη ολοκλήρωσή του και χωρίς ενδιάμεση συμβολή του τελικού πελάτη, υπάρχει πάντα ο κίνδυνος οι εξελίξεις να καταστήσουν το προϊόν ξεπερασμένο. Ταυτόχρονα, η έξαρση που υπάρχει τα τελευταία χρόνια στην ανάπτυξη Web 2.0 εφαρμογών συνεπάγεται ισχυρό ανταγωνισμό. Η υιοθέτηση μιας παραδοσιακής μεθόδου δεν βοηθά στο να προβλεφθεί και να προληφθεί ο ανταγωνισμός, αφού δεν υπάρχει η λογική μιας σταδιακής παράδοσης λειτουργικού λογισμικού. Η έλλειψη προσαρμοστικότητας και γρήγορης ανταπόκρισης φαίνεται να μην ταιριάζει σαν λογική στην ανάπτυξη web-based λογισμικού, το οποίο έχει κατά κύριο λόγο ανάγκη από τα δύο αυτά χαρακτηριστικά.

ΚΕΦΑΛΑΙΟ

4

ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ ΠΑΡΑΔΟΣΙΑΚΩΝ ΜΕΘΟΔΩΝ

4.1 Πλατφόρμα *EAT-IN*

4.1.1 Εισαγωγή - Η ιδέα

Στο παρόν κεφάλαιο παρουσιάζεται μια πλατφόρμας ηλεκτρονικής παραγγελίας φαγητού της οποίας η ο σχεδιασμός και η ανάπτυξη έγινε βασιζόμενη στις αρχές των παραδοσιακών μεθόδων ανάπτυξης λογισμικού, όπως παρουσιάστηκαν προηγουμένως. Ακολουθώντας τη λογική των παραδοσιακών μεθόδων αρχικά έπρεπε να οριστεί ποιό ήταν το πρόβλημα που επιχειρούσαμε να λύσουμε μέσω της πρότασης μας. Η βασική ιδέα προς υλοποίηση ήταν η δημιουργία μιας πλατφόρμας η οποία θα απευθύνεται κυρίως σε εργαζόμενους οι οποίοι κατά τη διάρκεια του μεσημεριανού φαγητού βρίσκονται ακόμα στο χώρο εργασίας τους. Κύριος στόχος ήταν η ανάπτυξη της εφαρμογής με τέτοιο τρόπο προκειμένου να διευκολύνει τη διαδικασία απόφασης και παραγγελίας του φαγητού προωθώντας έναν πιο ισορροπημένο τρόπο διατροφής ακόμα και τις ώρες απουσίας από το σπίτι. Με δεδομένο ότι πολλές εταιρείες στη χώρα μας δεν διαθέτουν υποδομές σίτισης για τους υπαλλήλους τους στις εγκαταστάσεις τους, αποφασίσαμε να δημιουργήσουμε μια εφαρμογή για όσους εργάζονται σε τέτοιες εταιρείες ώστε η διαδικασία σίτισης να πάψει να είναι καθημερινό μέλημα για αυτούς.

Μέσω της εφαρμογής ο εκάστοτε χρήστης θα μπορούσε να επιλέξει

μέσα από μια σειρά γευμάτων τα οποία θα διατίθενται από ένα σύνολο εστιατορίων. Η ιδέα δεν εμπεριέχει την παρουσίαση κατάλογου συγκεκριμένου εστιατορίου αλλά στοχεύει στο να δώσει μια ποικιλία επιλογών στο χρήστη μέσα από γεύματα που τα ίδια τα συνεργαζόμενα με την εφαρμογή εστιατόρια έχουν διαθέσει. Έτσι υιοθετήθηκε η ιδέα δημιουργίας ενός δυναμικού είτε ημερήσιου είτε εβδομαδιαίου μενού για τον κάθε χρήστη βάσει των προτιμήσεων του, όπως αυτές έχουν καταγραφεί σε προηγούμενες παραγγελίες του. Ταυτόχρονα η εφαρμογή θα εξυπηρετούσε και τους ιδιοκτήτες εστιατορίων εξίσου, δίνοντάς τους τη δυνατότητα να έχουν άμεσο έλεγχο των παραγγελιών αλλά και των γευμάτων που επιθυμούν να διαθέσουν μέσω ενός ειδικά διαμορφωμένου πάνελ.

Για την υλοποίηση αυτής της ιδέας ακολουθήθηκαν αρχές και αξίες που διέπουν τις παραδοσιακές μεθόδους ανάπτυξης λογισμικού. Αρχικά προχωρήσαμε στη διαδικασία ανάλυσης των απαιτήσεων. Σε αυτή τη φάση ερχόμενοι σε επαφή με εργαζομένους σε κάποιες εταιρείες θελήσαμε μέσα από τη διαδικασία συνεντεύξεων να δούμε αν υπάρχει όντως το πρόβλημα του οποίου τη λύση επιδιώκαμε και σε τι βαθμό ενώ ταυτόχρονα έγινε και μια προσπάθεια συλλογής των απαιτήσεων των πιθανών χρηστών. Εν συνεχεία, έχοντας αποκτήσει μια ικανοποιητική εικόνα και λειτουργώντας με τη λογική της πρόβλεψης πιθανών μελλοντικών απαιτήσεων, προχωρήσαμε στην ανάπτυξη της πλατφόρμας EAT-IN όπως αυτή θα παρουσιαστεί παρακάτω. Τέλος παρουσιάζοντας την υλοποιημένη εφαρμογή σε μερίδα των ατόμων που αρχικά είχαμε προσεγγίσει αξιολογήσαμε την αξία της και τη λειτουργικότητα της για τους πιθανούς τελικούς χρήστες.

Μέσα από τις εμπειρίες που συλλέξαμε από την προσπάθεια αυτή έγινε πιο ξεκάθαρο για ποιούς λόγους η υιοθέτηση των παραδοσιακών μεθόδων δεν είναι η κατάλληλη για την ανάπτυξη τέτοιων εφαρμογών καθώς τα αποτελέσματα έδειξαν πως υπήρξε αρκετή σπατάλη χρόνου και επιπλέον κόπος για τη δημιουργία χαρακτηριστικών που εν τέλει οι πιθανοί χρήστες δεν φάνηκε να ενδιαφέρονται τόσο ή γιατί δεν τα είχαν φανταστεί όπως τελικά υλοποιήθηκαν. Η συμμετοχή του τελικού πελάτη μόνο στα αρχικά στάδια της ανάπτυξης του προϊόντος φάνηκε να δημιουργεί προβλήματα που δύσκολα θα μπορούσαν εξ'αρχής να γίνουν αντιληπτά όπως θα δούμε και παρακάτω.

4.1.2 Ανάλυση Απαιτήσεων

Στη φάση της ανάλυσης των απαιτήσεων αυτό που είχε σημασία να διερευνήσουμε ήταν αν αρχικά η αρχική ιδέα επρόκειτο να έχει κάποια ανταπόκριση. Κοινώς, ζητούμενο ήταν να δούμε αν όντως το πρόβλημα που θα έλυνε η εφαρμογή υπήρχε ή αν οι ήδη υπάρχουσες εφαρμογές το κάλυπταν. Επίσης στόχος ήταν να εντοπιστεί αν ο τρόπος που επιχειρούσε η ιδέα μας να επιλύσει το πρόβλημα ήταν επαρκώς αποδεκτή. Τέλος η συλλογή προτάσεων από τους πιθανούς τελικούς χρήστες θα οδηγούσε σε

μια τελική εικόνα του προϊόντος πριν ξεκινήσει η υλοποίησή του.

Για τους λόγους αυτούς πραγματοποιήσαμε ένα σύνολο συνεντεύξεων - συζητήσεων με 40 εργαζομένους σε εταιρείες της πόλης. Οι ερωτήσεις που τέθηκαν παρατίθενται στο παράρτημα αυτής της εργασίας. Μέσα από τις συνεντεύξεις αυτές προέκυψε πως όντως οι πιο πολλές εταιρείες δεν διαθέτουν εγκαταστάσεις σίτισης για τους εργαζομένους τους. Η πλειοψηφία αυτών περιορίζεται στη διάθεση αυτόματου πωλητή και κάποιων ηλεκτρικών συσκευών όπως καφετιέρα, τοστιέρα και φούρνο μικροκυμάτων. Κάποιες από αυτές διαθέτουν κυλικείο με λίγες επιλογές ενώ ελάχιστες διαθέτουν εγκαταστάσεις σίτισης. Η συντριπτική πλειοψηφία των συνεντευξιζόμενων δήλωσε πως έχει περίπου μισή ώρα με 45 λεπτά διάλειμμα για γεύμα ημερησίως και πως αρκετές φορές επιλέγουν να παραγγείλουν το φαγητό τους καθώς δεν το έχουν προμηθευτεί από το σπίτι τους. Ιδίως άτομα μικρότερης ηλικίας, νέοι στο χώρο εργασίας και άτομα που δεν έχουν δημιουργήσει οικογένεια τείνουν να παραγγέλλουν πιο συχνά το μεσημεριανό φαγητό τους.

Μέρος αυτών δήλωσε πως πραγματοποιεί της παραγγελίες τηλεφωνικά ως επί το πλείστον. Καταφεύγουν όμως και στη λύση της ηλεκτρονικής παραγγελίας κάποιες φορές. Μικρότερο ποσοστό επιλέγει να πραγματοποιήσει τις παραγγελίες ηλεκτρονικά. Όσοι επέλεγαν την ηλεκτρονική παραγγελία δήλωσαν πως συχνά ο χρόνος μέχρι την τελική απόφαση ήταν περισσότερος από όσο θα έπρεπε. Συχνά ήταν αναποφάσιστοι και κατέληγαν να χάνονται σε μια πλειάδα καταλόγων διαφόρων εστιατορίων με αποτέλεσμα να καθυστερεί το διάλειμμά τους. Σε κάποιες περιπτώσεις μάλιστα, σπάνιες σχετικά, όταν έφταναν στην τελική τους επιλογή υπήρχε πιθανότητα να μην είναι διαθέσιμο το συγκεκριμένο προϊόν παρά το γεγονός ότι αναγραφόταν στους ηλεκτρονικούς καταλόγους. Το γεγονός αυτός δήλωσαν πως επηρέαζε αρνητικά τη διάθεσή τους.

Η ιδέα να υπάρχει συγκεντρωμένη η πληροφορία σχετικά με τη διάθεση των γευμάτων για την εκάστοτε ημέρα τους φάνηκε ελκυστική και θεώρησαν πως θα τους διευκόλυνε και θα εξοικονομούσαν χρόνο. Στην πιθανότητα να υπάρχει ένα εβδομαδιαίο μενού βάσει των προτιμήσεών τους επίσης έδειξαν θετική προδιάθεση και το βρήκαν ενδιαφέρον. Οι γενικότερες απαιτήσεις τους ήταν να μπορούν να βρίσκουν εύκολα την πληροφορία που τους ενδιαφέρει, η πληροφορία αυτή να είναι έγκυρη προκειμένου η διαδικασία παραγγελίας να είναι γρήγορη και εύκολη. Επιπροσθέτως δήλωσαν ότι θα επιθυμούσαν να του παρέχεται ενημέρωση σχετικά με το χρόνο παράδοσης της παραγγελίας τους προκειμένου να γνωρίζουν τη ώρα τελικά θα κανονίσουν το διάλειμμά τους.

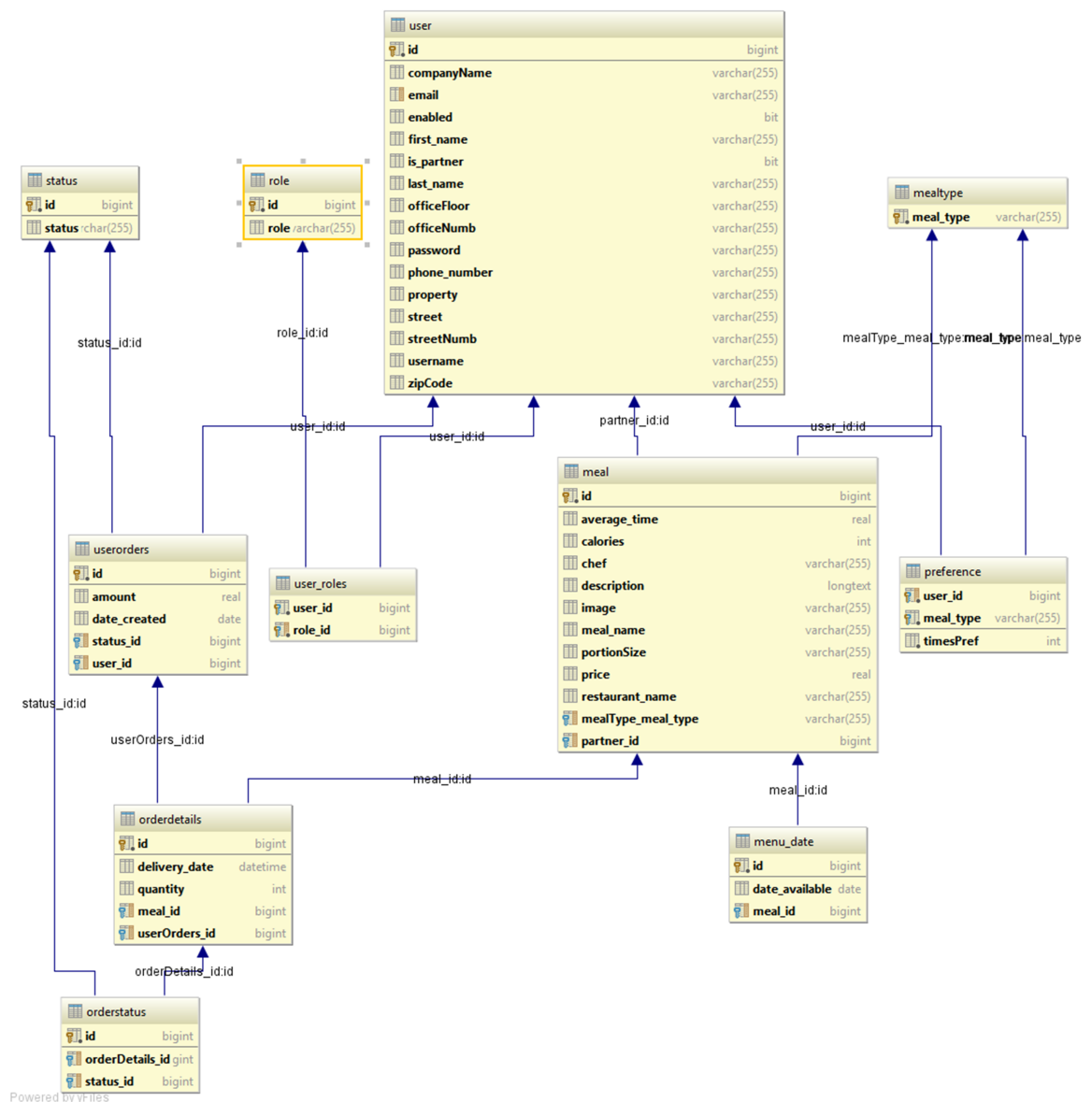
Τέλος, οι ερωτηθέντες έδειξαν ιδιαίτερα ένθερμοι στην ιδέα να τους παρέχεται πλήρης πληροφόρηση σχετικά με το γεύμα που θα παραγγείλουν. Ήθελαν να υπάρχει επαρκής πληροφορία για τη διατροφική αξία του, τα συστατικά που το αποτελούν, τις θερμίδες που έχει και ούτω καθ'εξής. Η πιθανότητα να έχουν πραγματική εικόνα του φαγητού που θα κατανα-

λώσουν επίσης τους ενδιέφερε πολύ και θα επιθυμούσαν να παρέχεται σαν επιλογή από μια εφαρμογή ηλεκτρονικής παραγγελίας φαγητού.

Μετά το πέρας των συνεντεύξεων απο τις οποίες και προέκυψαν οι ανάγκες των πιθανών χρηστών έγινε ένα αρχικό προσχέδιο της εφαρμογής αναγράφοντας ποια στοιχεία είναι εκείνα που σίγουρα θα θέλαμε να υλοποιηθούν. Στην πορεία της υλοποίησης η πρόβλεψη πιθανών αναγκών θα έδινε μια τελική μορφή η οποία και θα παραδινόταν ολοκληρωμένη στον πιθανό τελικό πελάτη. Στην παρούσα φάση επιχειρήθηκε και η προσέγγιση ιδιοκτητών εστιατορίων προκειμένου να δούμε και τις δικές του απόψεις σχετικά. Η συμμετοχή όμως δεν ήταν επαρκής προκειμένου να συμπεριληφθούν τα αποτελέσματα στην εργασία αυτή. Για το λόγο αυτό τα χαρακτηριστικά που υλοποιήθηκαν στην πλατφόρμα και αφορούν τους πιθανούς συνεργάτες έχουν χαρακτήρα πιο πολύ πρόβλεψης.

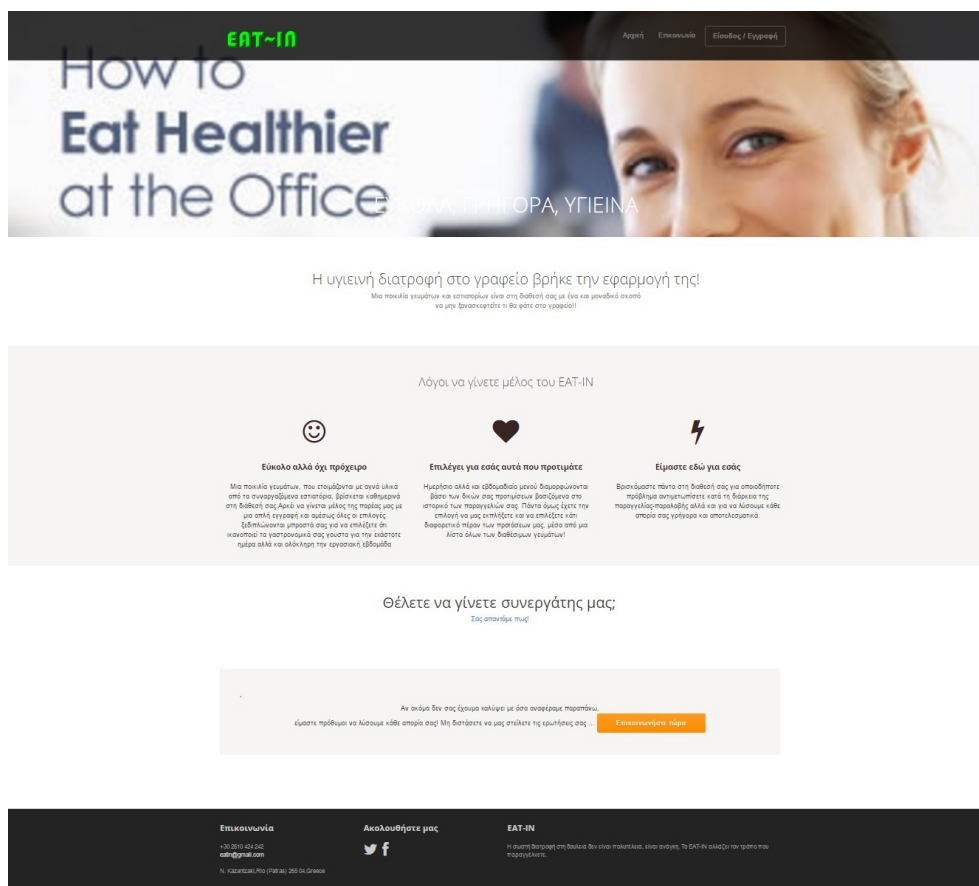
4.1.3 Υλοποίηση Πλατφόρμας

Στο παρόν τμήμα της εργασίας θα γίνει παρουσίαση της εφαρμογής EAT-IN όπως αυτή υλοποιήθηκε βασιζόμενοι στην πληροφορία που ορίστηκε στην φάση ανάλυσης απαιτήσεων. Σύμφωνα με τον εκπρόσωπο των παραδοσιακών μεθόδων, το μοντέλο του καταρράκτη, η διαδικασία της υλοποίησης έγινε χωρίς περεταίρω ανάμειξη των πιθανών τελικών χρηστών. Η πλατφόρμα που προέκυψε παρουσιάζεται παρακάτω δίνοντας μια πλήρη εικόνα της διαδικασίας παραγγελίας καθώς και των δυνατοτήτων που γενικά που παρέχει τόσο στον χρήστη που ενδιαφέρεται να αγοράσει το γεύμα του όσο και στον πάροχο του γεύματος, δηλαδή το εστιατόριο. Για την υλοποίησή της έγινε χρήση των τεχνολογιών Spring Framework (Spring Boot), Hibernate, Thymeleaf μηχανής προτύπων, Twitter Bootstrap, τις οποίες και θα δούμε αναλυτικότερα παρακάτω, καθώς και των τεχνολογιών HTML, Javascript, MySQL βάσης δεδομένων και του PayPal API. . Στην εικόνα 4.1 δίνεται το σχήμα της βάσης δεδομένων της πλατφόρμας.



Σχήμα 4.1: Σχήμα βάσης δεδομένων EAT-IN.

Η πρώτη επαφή που έχουν οι πιθανοί χρήστες με την πλατφόρμα γίνεται μέσω της αρχικής σελίδας όπου παρουσιάζονται γενικές πληροφορίες καθώς και τρόποι για να ακολουθήσουν το EAT-IN στα μέσα κοινωνικής δικτύωσης [Σχήμα 4.2]. Από εκεί επιλέγοντας την "Είσοδο/Εγγραφή" ανακατευθύνονται στη φόρμα εισόδου. Αν δεν διαθέτουν ήδη κάποιο λογαριασμό τους προτείνεται να κάνουν εγγραφή.



Σχήμα 4.2: Αρχική σελίδα EAT-IN.

Και οι δύο πιθανοί πελάτες πρέπει αρχικά να εγγραφούν. Ο καθένας πρέπει να εγγραφεί βάσει της ιδιότητας του και αναλόγως της επιλογής οδηγείται σε μια δεύτερη φόρμα όπου πρέπει να συμπληρώσει κάποια επιπλέον στοιχεία ώστε να ολοκληρώσει την εγγραφή του. Οι επιλογές που δίνονται είναι εγγραφή ως χρήστης (user) ή ως συνεργάτης (partner) [Σχήμα 4.3]. Σε περίπτωση που έχουμε εγγραφή νέου user, αφού συμπληρώσει την αρχική καρτέλα μεταφέρεται σε μια δεύτερη όπου συμπληρώνει στοιχεία σχετικά με το χώρο εργασίας του, όπως το όνομα της εταιρείας που εργάζεται, διεύθυνση αυτής και λοιπά. Τα στοιχεία αυτά στην παρούσα φάση είναι προαιρετικά και μπορεί να τα συμπληρώσει αργότερα. Στην περίπτωση που είναι νέος partner θα πρέπει να συμπληρώσει πλήρως τα στοιχεία της επιχείρησης που του ζητούνται στη δεύτερη καρτέλα. Αφού ολοκληρωθεί η εγγραφή εισέρχονται στην εφαρμογή ανακατευθυνόμενος ο καθένας αναλόγως της ιδιότητας του σε διαφορετική σελίδα [Σχήματα 4.4, 4.5].

Νέα εγγραφή

Δημιουργία νέου λογαριασμού
Είσοδος εδώ σε περίπτωση που διαθέτε ήδη λογαριασμό

Εγγραφείτε ως νέος : USER
USER
PARTNER

Όνομα
Όνομα

Επώνυμο
Επώνυμο

Ηλεκτρονική διεύθυνση *
Ηλεκτρονική διεύθυνση
Παράδειγμα: ονομα@domain.com

Όνομα χρήστη
Όνομα χρήστη

Κωδικός *
Κωδικός

Επιβεβαίωση Κωδικού *
Επανάλβετε κωδικό

Ελάχιστο: 6 αλφαριθμητικά

☐ Διάβασα τους Όρους Χρήσης

Εγγραφή

Σχήμα 4.3: Διαδικασία αρχικής εγγραφής κοινή και για τα δύο είδη πελάτων.

Ολοκλήρωση εγγραφής
Συμπληρώστε τα στοιχεία της εταιρίας που εργάζεστε.

Επωνυμία Εταιρίας
Επωνυμία Εταιρίας

Διευθυνση-Οδός
Διευθυνση-Οδό

Διευθυνση-Αριθμός *
Διευθυνση-Αριθμός

Τ.Κ.: *
Τ.Κ.

Όροφος γραφείου σας
Όροφος

Αριθμός γραφείου *
Αριθμός γραφείου

Τηλ. Επικοινωνίας *
Τηλ. Επικοινωνίας

Ολοκλήρωση Εγγραφής

Αργότερα

Σχήμα 4.4: Εγγραφή χρήστη.

Ολοκληρώστε την εγγραφή σας
Συμπληρώστε τα στοιχεία της επιχείρησής σας.

Όνομα Εστατορίου: *
Collazione

Διευθυνση-Οδός
Διευθυνση-Οδός

Διευθυνση-Αριθμός *
Διευθυνση-Αριθμός

Τ.Κ.: *
Τ.Κ.

Τηλ.Επικοινωνίας: *
2610454545

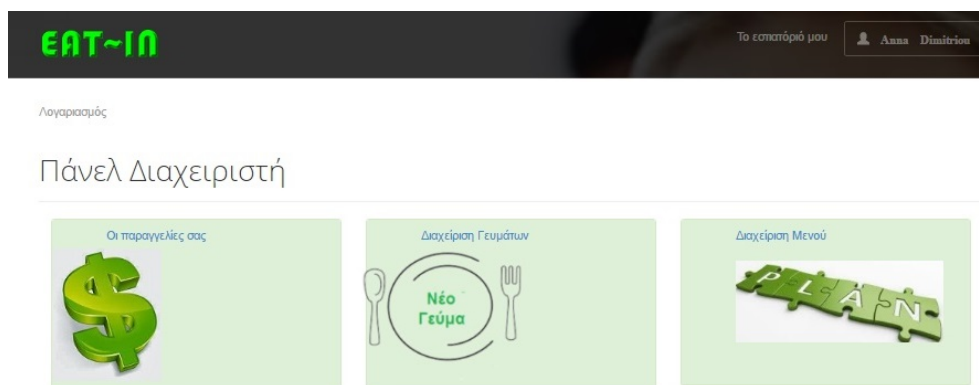
Ολοκλήρωση Εγγραφής

Σχήμα 4.5: Έγγραφή συνεργάτη.

B2B πελάτες

Ζητούμενο της προσπάθειας αυτής ήταν να διαφοροποιηθεί από τα ήδη υπάρχοντα προϊόντα λογισμικού που εξυπηρετούν τον τομέα ηλεκτρονικής παραγγελίας φαγητού. Για το λόγο αυτό δίνεται μια ελευθερία διάθεσης και διαχείρισης των γευμάτων από τον εκάστοτε συνεργάτη. Εκείνος θα αποφασίσει τι μπορεί να διαθέσει ανά πάσα στιγμή και δεν θα γίνεται παρουσίαση του συνολικού μενού. Με αυτό τον τρόπο αποφεύγονται κατάλογοι που δεν είναι ενημερωμένοι σωστά καθώς και η ανάμειξη ενδιάμεσου, γνώστη των απαραίτητων τεχνολογιών, προκειμένου να γίνουν οι όποιες αλλαγές.

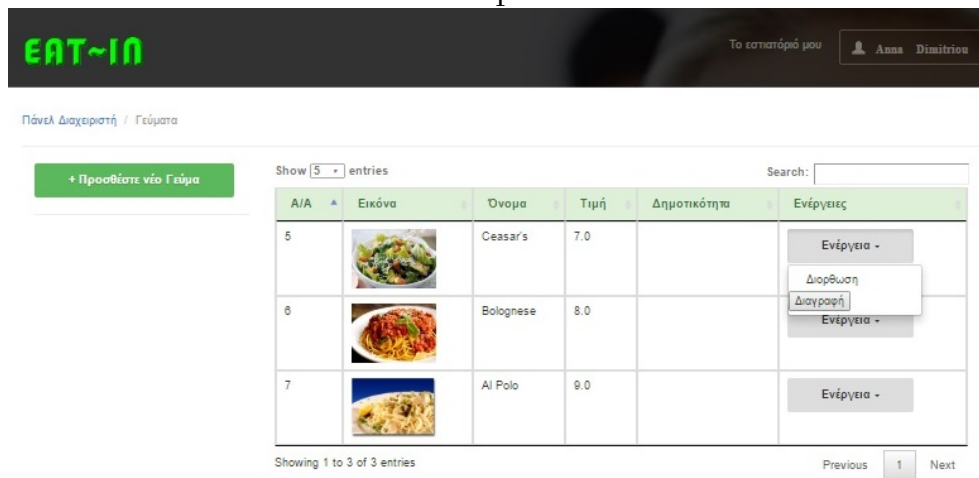
Όταν ο συνεργάτης εισαχθεί η εφαρμογή τον κατευθύνει σε ένα πάνελ μέσα από το οποίο μπορεί να κάνει τα εξής [Σχήμα 4.6]:



Σχήμα 4.6: Πάνελ συνεργάτη.

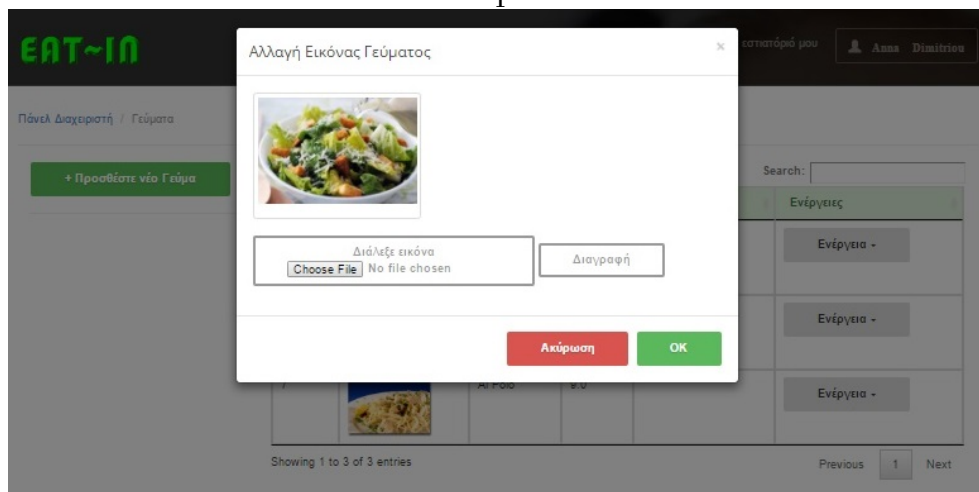
- **Να προσθέσει να αλλάξει πληροφορίες για τα γεύματα (Διαχείριση Γευμάτων):** Τα προς διάθεση από τον συνεργάτη γεύματα ορίζονται στην εφαρμογή από τον ίδιο. Έχει πλήρη εικόνα των γευμάτων που ήδη έχει διαθέσει ενώ όποτε επιθυμεί μπορεί να προσθέσει ένα νέο να διαφοροποιήσει ένα ήδη υπάρχον ή και να το διαγράψει [Σχήμα 4.7, Σχήμα 4.8]. Όταν καλείται να προσθέσει νέο γεύμα πρέπει να συμπληρώσει τις απαραίτητες πληροφορίες που το αφορούν και να το συνοδεύει με ανάλογη εικόνα [Σχήμα 4.9. Η διάθεση των γευμάτων είναι αποκλειστικά υπόθεση του ιδίου και δεν απαιτεί την ανάμειξη κάποιου που θα πρέπει να χειριστεί τον κώδικα της εφαρμογής.

1



Σχήμα 4.7: Διαγραφή ή διόρθωση διαθέσιμων γευμάτων.

1



Σχήμα 4.8: Αλλαγή εικόνας γεύματος.

Πάνελ Διαχείριση / Νέο Γεύμα

Προσθέστε νέο γεύμα!

Όνομα εστιατορίου

Όνομα Γεύματος

Τιμή

Μέσος χρόνος διανομής

Όνομα chef

Μέγεθος μερίδας

Θερμίδες

Τύπος

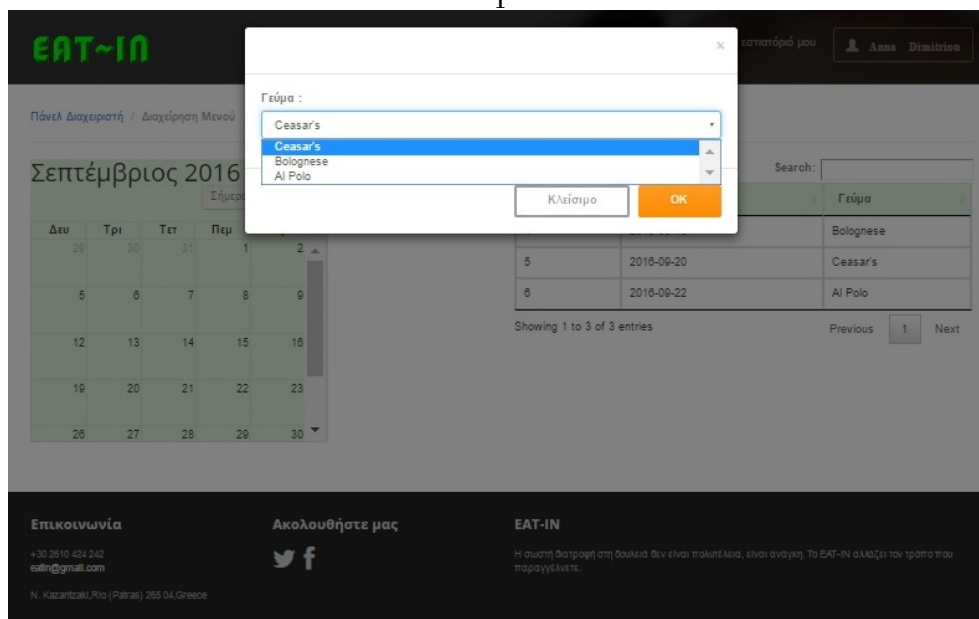
Περιγραφή Γεύματος
 Δώστε συνταγή ή συστατικά του πιάτου...

Εικόνα Γεύματος
 No file chosen

Σχήμα 4.9: Προσθήκη νέου γεύματος.

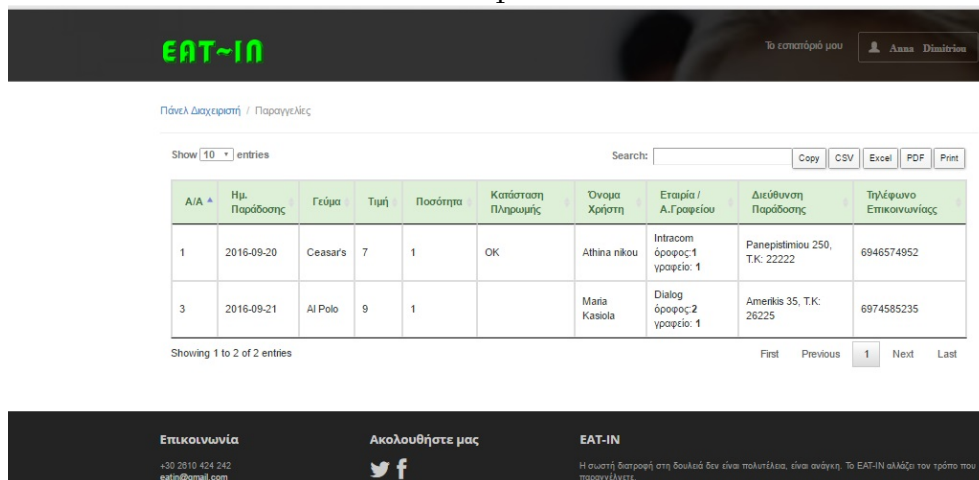
- **Να ορίσει ημερομηνία διάθεσης του κάθε γεύματος (Διαχείριση Μενού)** :Ο εκάστοτε συνεργάτης μπορεί να τοποθετήσει το εκάστοτε γεύμα στην ημερομηνία που το έχει ή θα το έχει διαθέσιμο. Το παρέχεται ένα ημερολόγιο το και πατώντας στην ημέρα που επιθυμεί εμφανίζεται ένα παράθυρο από το οποίο μπορεί να επιλέξει από τα γεύματα που ήδη έχει ανεβάσει στην εφαρμογή. [Σχήμα 4.10]
- **Να παρακολουθεί τις παραγγελίες που τον αφορούν (Οι παραγγελίες μου)**: Όλες οι παραγγελίες που αφορούν το εστιατόριο του συνεργάτη εμφανίζονται στη σελίδα των παραγγελιών σε ειδικό πίνακα. Τα στοιχεία που παρέχονται είναι η ημερομηνία παράδοσης, στοιχεία διανομής, τηλέφωνο επικοινωνίας, καθώς και η κατάσταση της παραγγελίας αν έχει αποπληρωθεί η όχι [Σχήμα 4.11]. Με τον τρόπο αυτό το κάθε εστιατόριο έχει εικόνα των γευμάτων που καλείται να παραδώσει ακόμα και για επόμενες ημερομηνίες από την τρέχουσα.

1



Σχήμα 4.10: Διαχείριση μενού.

1



Σχήμα 4.11: Πίνακας παραγγελιών συνεργάτη

Όλες οι παραπάνω ενέργειες παρέχουν αυτονομία στον πιθανό συνεργάτη καθώς είναι σε θέση να αποφασίσει ο ίδιος τι επιθυμεί να διαθέσει την εκάστοτε μέρα και να διαχειριστεί γενικότερα τη παροχή γευμάτων ανάλογα με τη διαθεσιμότητά του. Όποια αλλαγή απαιτείται, όπως για παράδειγμα αν κάποιο πιάτο του τελειώσει, είναι σε θέση άμεσα να ενημερώσει την εφαρμογή χωρίς επιπλέον βοήθεια και με τον τρόπο αυτό αποφεύγονται διαδικασίες επικοινωνίας με το χρήστη που έκανε την παραγγελία προκειμένου να προβεί σε τροποποίηση αυτής. Το EAT-IN στοχεύει στην συγκέντρωση πολλών εστιατορίων που θα κάνουν τη διαδικασία

που αναφέρεται παραπάνω. Με τον τρόπο αυτό καθημερινά θα υπάρχει μια ποικιλία γευμάτων και εστιατορίων δίνοντας έτσι πολλές επιλογές στο χρήστη αλλά με τρόπο τέτοιο που να κάνει τη διαδικασία της αναζήτησης και της εύρεσης του γευμάτων ευκολότερη όπως θα δούμε αμέσως μετά.

B2C πελάτες

Όταν ο χρήστης επιθυμεί να πραγματοποιήσει την παραγγελία του μέσω της εφαρμογής καλείται όπως είδαμε και παραπάνω να εγγραφεί σε περίπτωση που δεν διαθέτει ήδη λογαριασμό, διαφορετικά προχωράει σε είσοδο και είναι έτοιμος πλέον να παραγγείλει.

EAT-IN Αρχική Μενου Καλάθι 0 Athens nikou

Αρχική / Εβδομαδιαίο Μενού

Εβδομαδιαίο Μενού σύμφωνα με τις προτιμήσεις σου *Athina*

Θυμάσαι πως μπορείς να δεις όλα τα διαθέσιμα ανά ημέρα γεύματα πατώντας στην ανάλυση ημερομηνιών!

17 September 2016	18 September 2016	19 September 2016	20 September 2016	21 September 2016
 4.00€ Vegan Soup and Noodle Bar To Θέλω	 8.00€ Bolognese and Collazione To Θέλω	Δεν υπάρχουν ακόμα διαθέσιμα γεύματα για την ημέρα αυτή. Δοκιμάστε αργότερα! Λυπούμαστε!	 7.00€ Caesar's and Collazione To Θέλω	 4.00€ Vegan Soup and Noodle Bar To Θέλω
 4.00€ Liam and Fainon To Θέλω	 6.00€ Noodles and Noodle Bar To Θέλω		 6.00€ Kanelonia and Fainon To Θέλω	

Επικοινωνία
+30 2610 424 242
eatin@gmail.com
N. Kazantzaki Rio (Patras) 265 04, Greece

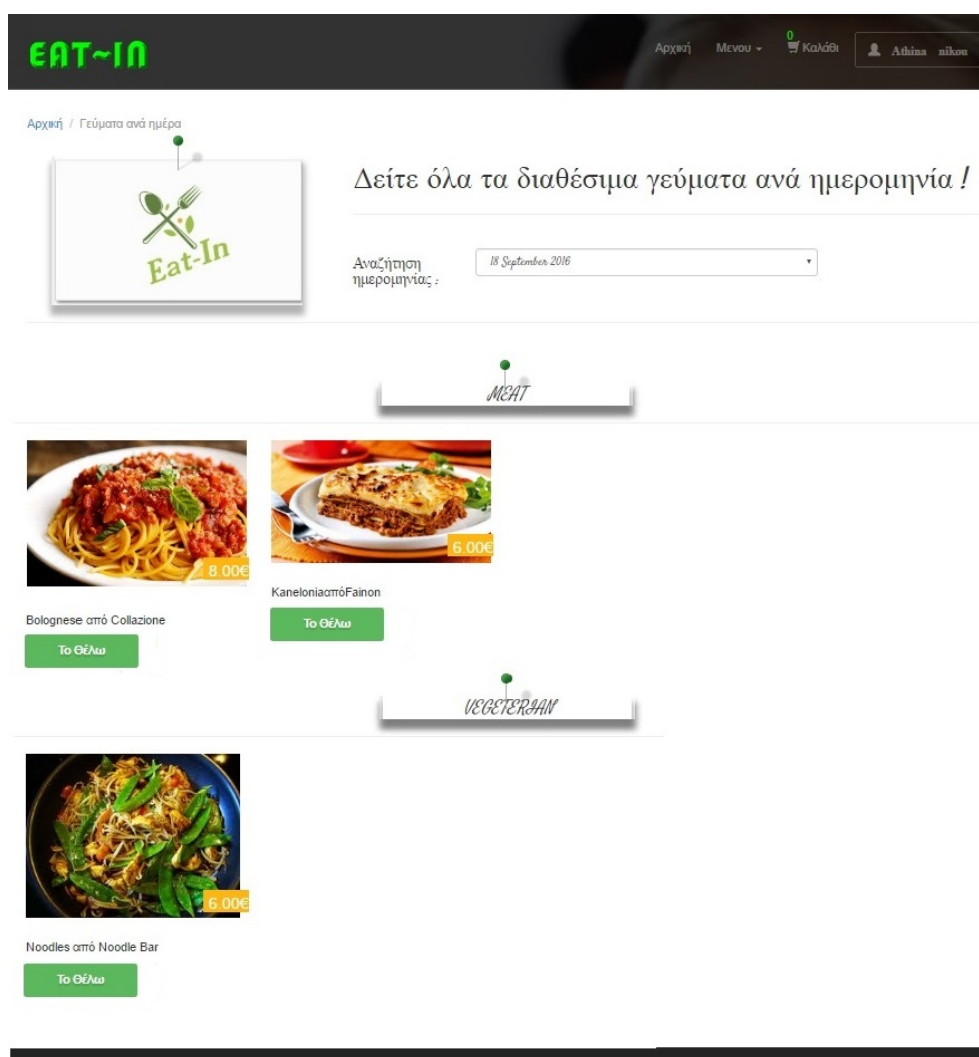
Ακολουθήστε μας
Twitter Facebook

EAT-IN
Η σωστή διατροφή στη δουλειά δεν είναι πολυτέλεια, είναι ανάγκη. Το EAT-IN αλλάζει τον τρόπο που παραγγέλνεται.

Copyright © 2016, ITYE Λιόφοντος.

Σχήμα 4.12: Εβδομαδιαίο μενού βάσει προτιμήσεων χρήστη.

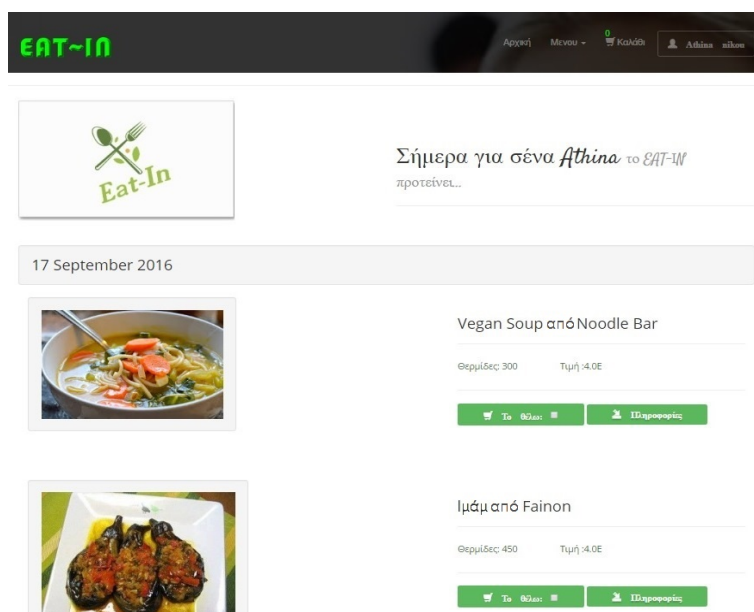
Με την είσοδό του ανακατευθύνεται σε σελίδα η οποία εμφανίζει ένα εβδομαδιαίο μενού [Σχήμα 4.12]. Το μενού αυτό περιλαμβάνει γεύματα για την τρέχουσα ημερομηνία και για της επόμενες 4 εργάσιμες. Κάθε ημέρα προτείνονται στο χρήστη δύο γεύματα. Σε περίπτωση που είναι νέος χρήστης η επιλογή αυτών των γευμάτων γίνεται τυχαία από όλα τα διαθέσιμα για εκείνη την ημέρα γεύματα. Αν έχει ήδη πραγματοποιήσει παραγγελίες τότε το μενού αυτό διαμορφώνεται αναλόγως των προτιμήσεων του. Ένα από τα δύο γεύματα επιλέγεται βάσει του υπερισχύοντος προτιμώμενου τύπου γεύματος (θαλασσινά, κρεατικά, χορτοφάγων), ενώ το δεύτερο επιλέγεται τυχαία από μία από τις υπόλοιπες κατηγορίες.



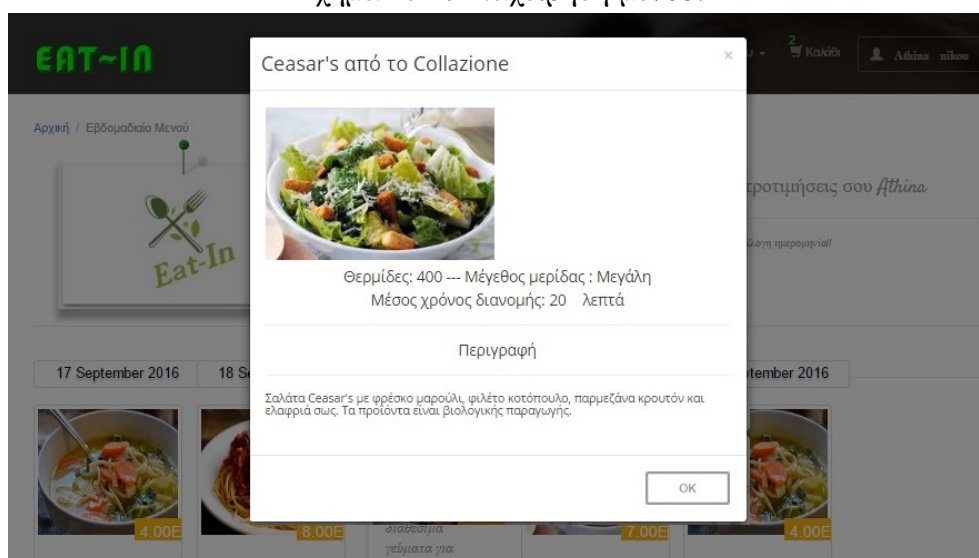
Σχήμα 4.13: Συγκεντρωτική εικόνα διαθέσιμων γευμάτων κατηγοριοποιημένα βάσει του τύπου τους.

Κάνοντας κλίκ στην αναγραφόμενη ημερομηνία που υπάρχει πάνω

από κάθε ζεύγος γευμάτων ο χρήστης ανακατευθύνεται σε σελίδα που του παρουσιάζει όλα τα διαθέσιμα γεύματα καταναμεημένα βάσει του τύπου τους. Στη σελίδα αυτή έχει τη δυνατότητα να αναζητήσει τα γεύματα και για τις υπόλοιπες ημερομηνίες της εβδομάδας [Σχήμα 4.13].



Σχήμα 4.14: Διαχείριση μενού.

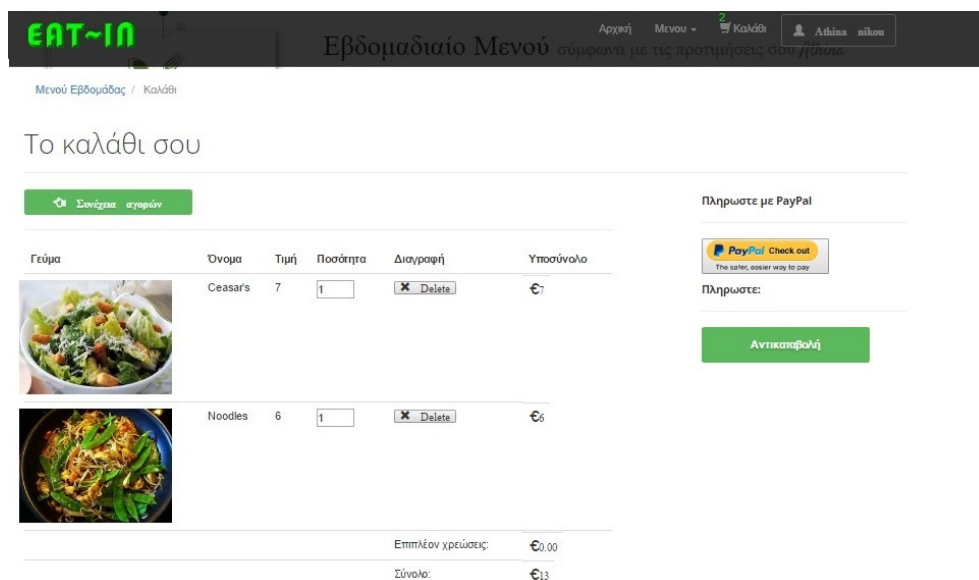


Σχήμα 4.15: οι διαθέσιμες πληροφορίες για το γεύμα

Εκτός από το εβδομαδιαίο μενού και όλα τα διαθέσιμα βάσει τύπου γεύματα ο χρήστης έχει τη δυνατότητα να δει ξεχωριστά στο ημερήσιο μενού τα δύο γεύματα που το προτείνονται από την εφαρμογή [Σχήμα

4.14]. Σε κάθε μία από τις προαναφερθείσες σελίδες δίνεται εικόνα του εκάστοτε γεύματος και ο πάροχος αυτού ενώ ο χρήστης μπορεί να έχει περισσότερες πληροφορίες για κάθε ένα ξεχωριστά απλά πατώντας πάνω στην αντίστοιχη εικόνα. Οι πληροφορίες που παρέχονται σχετίζονται με τη διατροφική αξία του γεύματος, την περιγραφή αυτού, το μέγεθός της μερίδας καθώς και το χρόνο που απαιτείται για την παράδοσή του [Σχήμα 4.15].

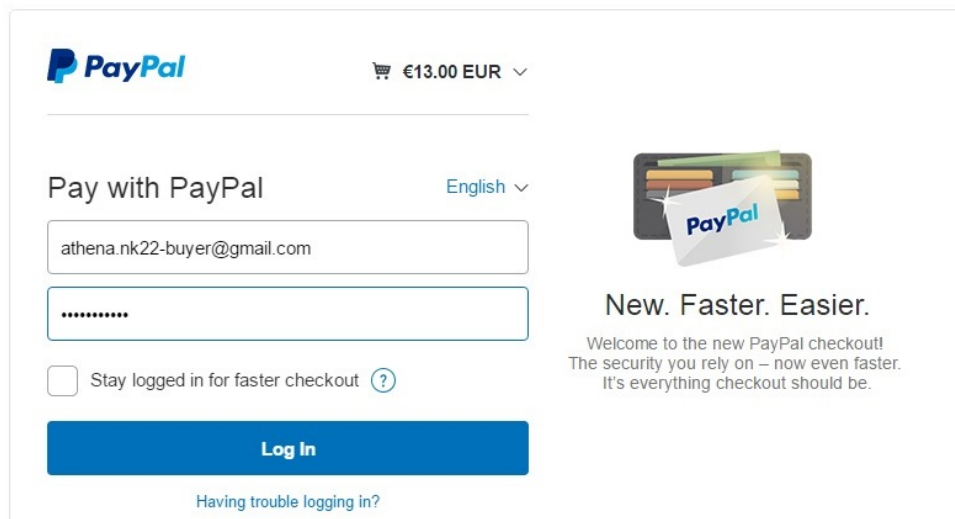
Ο χρήστης μπορεί να επιλέξει να παραγγείλει για όποια μέρα επιθυμεί, για μια αλλά και για περισσότερες μέρες στην εβδομάδα. Μόλις αποφασίζει προσθέτει τα πιάτα που επιθυμεί στο καλάθι αγορών του το οποίο και ενημερώνεται με τον ανάλογο αριθμό των προϊόντων που ζητήθηκαν [Σχήμα 4.16].



Σχήμα 4.16: Καλάθι αγορών χρήστη.

Αν αλλάξει γνώμη μπορεί να διαγράψει κάποιο από αυτά η και να αυξήσει τον αριθμό των μερίδων αλλά και να συνεχίσει τις αγορές του επιστρέφοντας στο εβδομαδιαίο μενού. Μόλις έχει ολοκληρώσει τη διαδικασία επιλογής επιλέγει τον τρόπο που επιθυμεί να αποπληρώσει το συνολικό κόστος της παραγγελίας του. Οι επιλογές που του δίνονται είναι είτε μέσω PayPal είτε με αντικαταβολή. Αν επιλέξει το πρώτο τρόπο ανακατευθύνεται στη σελίδα του PayPal και αφού πραγματοποιηθεί η πληρωμή οδηγείται σε επόμενη σελίδα όπου υπάρχει συγκεντρωτικά η παραγγελία του ενώ ταυτόχρονα του ζητείται να συμπληρώσει όποια στοιχεία δεν συμπλήρωσε κατά την εγγραφή του [Σχήματα 4.17, 4.18].

sand test's Test Store



PayPal €13.00 EUR

Pay with PayPal English

athena.nk22-buyer@gmail.com

.....

☐ Stay logged in for faster checkout ?

Log In

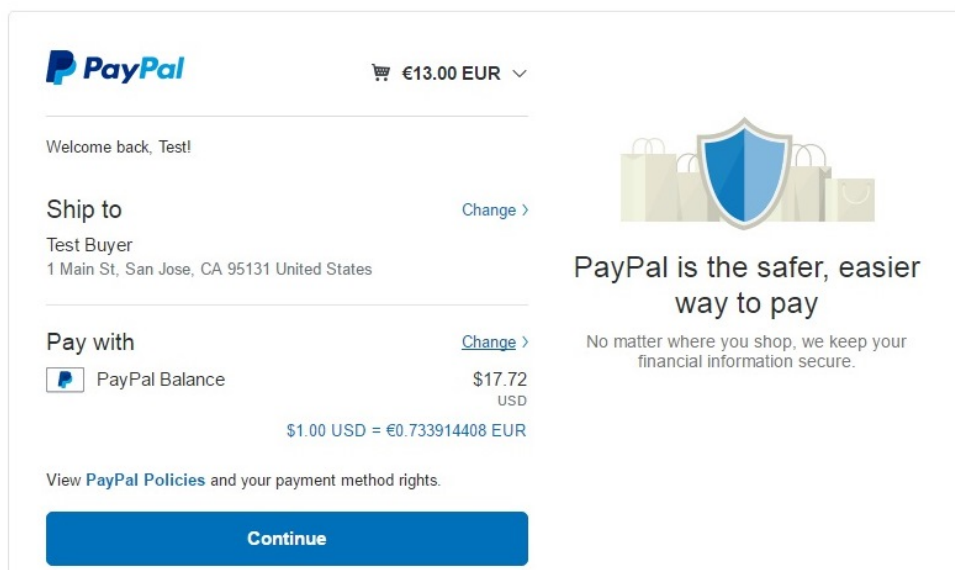
Having trouble logging in?

New. Faster. Easier.

Welcome to the new PayPal checkout!
The security you rely on – now even faster.
It's everything checkout should be.

Σχήμα 4.17: Είσοδος στον λογαριασμό PayPal του χρήστη.

sand test's Test Store



PayPal €13.00 EUR

Welcome back, Test!

Ship to Change

Test Buyer
1 Main St, San Jose, CA 95131 United States

Pay with Change

PayPal Balance \$17.72 USD

\$1.00 USD = €0.733914408 EUR

View PayPal Policies and your payment method rights.

Continue

PayPal is the safer, easier way to pay

No matter where you shop, we keep your financial information secure.

Σχήμα 4.18: Ολοκλήρωση πληρωμής μέσω PayPal.

Ο λογαριασμός PayPal που απεικονίζεται στις παραπάνω εικόνες προκύπτει με χρήση του Sandbox test account. Αφού ολοκληρώσει την πληρωμή οδηγείται στη συγκεντρωτική εικόνα της παραγγελίας του όπου και καλείται να συμπληρώσει τυχόν στοιχεία που δεν είχε συμπληρώσει στην αρχική εγγραφή. Αν τα στοιχεία του είναι όλα συμπληρωμένα επιλέγει απλώς ολοκλήρωση της παραγγελίας του. Στο ίδιο ακριβώς σημείο κατευ-

θύνεται και στην περίπτωση επιλογής πληρωμής με αντικαταβολή [Σχήμα 4.19].

Ολοκλήρωση Παραγγελίας

Η παραγγελία σας (Κωδικός παραγγελίας : 4)

Γεύμα	Ποσότητα	Ημέρα Παράδοσης
Cesar's	1	20 September 2018
Noodles	1	18 September 2018

Στοιχεία Παράδοσης

Όνομα:

Επώνυμο:

Εταιρεία:

Οδός:

Αριθμός: *

T.K. *

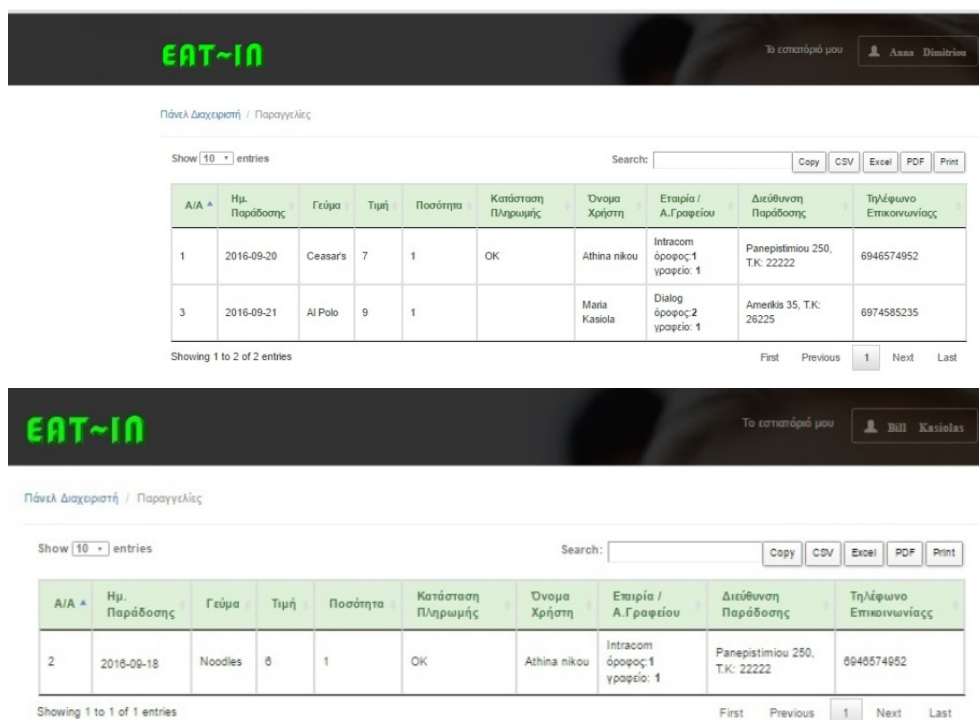
Όροφος γραφείου:

Αριθμός γραφείου *

Τηλέφωνο *

Σχήμα 4.19: Ολοκλήρωση παραγγελίας χρήστη.

Στο σημείο αυτό ολοκληρώνεται η διαδικασία παραγγελίας από το χρήστη και οι παραγγελίες διανέμονται στον ανάλογο συνεργάτη. Μια παραγγελία μπορεί να απαρτίζεται από γεύματα που ανήκουν σε παραπάνω από ένα εστιατόρια. Μέσω της εφαρμογής το εκάστοτε εστιατόριο λαμβάνει τις παραγγελίες που το αφορούν και αυτές καλείται να εξυπηρετήσει και να κάνει τη διανομή στην ανάλογη ημερομηνία [Σχήμα 4.20].



EAT-IN Το εστιατόριό μου [Ανα](#) [Διατήρηση](#)

Πάνελ Διαχειριστή / Παραγγελίες

Show **10** entries Search: [Copy](#) [CSV](#) [Excel](#) [PDF](#) [Print](#)

A/A *	Ημ. Παράδοσης	Γεύμα	Τιμή	Ποσότητα	Κατάσταση Πληρωμής	Όνομα Χρήστη	Εταιρία / Α.Γραφείου	Διεύθυνση Παράδοσης	Τηλέφωνο Επικοινωνίας
1	2016-09-20	Ceasar's	7	1	OK	Athina nikou	Intracom όροφος 1 γραφείο 1	Panepistimiu 250, T.K. 22222	6946574952
3	2016-09-21	Al Polo	9	1		Maria Kasiola	Dialog όροφος 2 γραφείο 1	Amerikis 35, T.K. 26225	6974585235

Showing 1 to 2 of 2 entries First Previous **1** Next Last

EAT-IN Το εστιατόριό μου [Bill](#) [Kasiolas](#)

Πάνελ Διαχειριστή / Παραγγελίες

Show **10** entries Search: [Copy](#) [CSV](#) [Excel](#) [PDF](#) [Print](#)

A/A *	Ημ. Παράδοσης	Γεύμα	Τιμή	Ποσότητα	Κατάσταση Πληρωμής	Όνομα Χρήστη	Εταιρία / Α.Γραφείου	Διεύθυνση Παράδοσης	Τηλέφωνο Επικοινωνίας
2	2016-09-18	Noodles	8	1	OK	Athina nikou	Intracom όροφος 1 γραφείο 1	Panepistimiu 250, T.K. 22222	6946574952

Showing 1 to 1 of 1 entries First Previous **1** Next Last

Σχήμα 4.20: Διανομή παραγγελιών στους ανάλογους συνεργάτες.

4.1.4 Αξιολόγηση Πλατφόρμας

Μετά το πέρας της υλοποίησης της πλατφόρμας ηλεκτρονικής παραγγελίας φαγητού που παρουσιάστηκε παραπάνω, ακολούθησε η αξιολόγηση αυτής παρουσιάζοντας την σε πιθανούς τελικούς χρήστες. Συγκεκριμένα παρουσιάστηκε σε ένα σύνολο εικοσιπέντε ατόμων το οποίο αποτελούνταν πλειοψηφικά από άτομα που πήραν μέρος στις αρχικές συνεντεύξεις για τον ορισμό των απαιτήσεων. Προστέθηκαν επί τούτου και άτομα τα οποία δεν είχαν συμπεριληφθεί στη φάση αυτή προκειμένου να υπάρχει ένα πιο αξιόπιστο αποτέλεσμα. Το σύνολο αυτών των ατόμων κλήθηκε να παρακολουθήσει τη διαδικασία πραγματοποίησης παραγγελίας μέσω της πλατφόρμας EAT-IN και να δηλώσει ποια στοιχεία τους ήταν αρεστά, ποια τους δυσάρεστους, ποιά τους ήταν αδιάφορα και γενικά να δηλώσουν τα επίπεδα ικανοποίησης τους από την εμπειρία τους με την εφαρμογή. Οι απαντήσεις που συλλέχθηκαν δεν ήταν όσο ευχάριστες αναμένονταν.

Οι πιθανοί χρήστες λοιπόν ανέφεραν πως η εικόνα της πλατφόρμας τους δημιουργούσε ευχάριστη αίσθηση και τους προδιέθετε πολύ θετικά. Τόσο τα χρώματα όσο και οι εικόνες τους δημιουργούσαν μια θετική προδιάθεση να χρησιμοποιήσουν το EAT-IN και να παραγγείλουν κάποιο από

τα διαθέσιμα γεύματα. θεώρησαν πως μέσω της πλατφόρμας είχαν όλη τη διαθέσιμη πληροφορία που αναζητούσαν και αρχικά φάνηκε να εστιάζουν πιο πολύ σε αυτό. Παρ'όλ'αυτά όσο τους παρουσιάζονταν τα βήματα της διαδικασίας παραγγελίας άρχισαν να αναφέρουν τα στοιχεία εκείνα που κατά τα λεγόμενά τους είτε δεν τους ικανοποιούσαν επαρκώς είτε δεν ανταποκρίνονταν σε πραγματικές περιπτώσεις.

Αρχικά το πρώτο στοιχείο που τους απώθησε βάσει των λεγομένων τους ήταν οι φόρμες εγγραφής. Τις χαρακτήρισαν μακροσκελείς και υποστήριξαν ότι αυτό πιθανότατα θα τους οδηγούσε στο να εγκαταλείψουν την εφαρμογή πριν ακόμα της δώσουν μια πρώτη ευκαιρία. Επιπροσθέτως ανέφεραν πως σύμφωνα με τα δεδομένα του χώρου εργασίας τους η φόρμα εγγραφής περιελάμβανε στοιχεία που δεν ήταν καν χρήσιμα, όπως τον όροφο του γραφείου και τον αριθμό αυτού καθώς οι εταιρείες τους δεν επιτρέπουν είσοδο μη εργαζομένων εκεί παρά οι ίδιοι βγαίνουν από το χώρο εργασίας στον προαύλιο χώρο να παραλάβουν την παραγγελία τους. Όλοι σχεδόν επέλεγαν την ολοκλήρωση της εγγραφής σε μεταγενέστερο στάδιο με την ελπίδα πως θα ξέφευγαν από τη βαρετή κατά τα λεγόμενά τους διαδικασία. Αυτό που προτιμούσαν είναι, εφόσον οι εγγραφές είναι ένα αναγκαίο κακό, να υπήρχε μια και μοναδική φόρμα η οποία θα περιλάμβανε τα απολύτως απαραίτητα, όπως το ονοματεπώνυμο κατά προτίμηση σε μια γραμμή, το όνομα και τη διεύθυνση της εταιρείας και ένα τηλέφωνο επικοινωνίας ώστε να τους ενημερώνουν όταν ο διανομέας θα έχει φτάσει. Όπως μας είπαν αυτή είναι η συνήθης διαδικασία που ακολουθούν όταν παραγγέλνουν από το τηλέφωνο. Τέλος, όσοι είχαν επιλέξει να παραλείψουν την πλήρη εγγραφή τους δυσανεστήθηκαν με το γεγονός ότι τους ζητούνταν στη σύνοψη της παραγγελίας τους να τα προσθέσουν. Δήλωσαν πως αυτός θα ήταν ένας λόγος να εγκαταλείψουν το καλάθι αγορών εκείνη τη στιγμή. Συνεπώς προέκυψαν ζητήματα που δεν είχαν υπολογιστεί από την αρχή καθώς οι ερωτηθέντες στην αρχική φάση δεν είχαν εστιάσει σε τόσες λεπτομέρειες και εφόσον η πληροφορία αυτή δεν ήταν ξεκάθαρη κατέληξε στο να υποτιμηθεί η επιρροή που έχει μια φόρμα εγγραφής στην διάθεση του πιθανού πελάτη. Προκειμένου να αλλάξουν έπρεπε να τροποποιήσουμε σε μεγάλο βαθμό τη διαδικασία εγγραφής γεγονός που κοστολογείται με τροποποίηση διακοσίων γραμμών κώδικα και αφαίρεση περίπου τετρακοσίων.

Δυσάρεστη έκπληξη επίσης αποτέλεσε και η αντίδραση ποσοστού των συμμετεχόντων στη διαδικασία αξιολόγησης της EAT-IN πλατφόρμας στην εικόνα του μενού με τις φωτογραφίες. Ανέφεραν πως τους θύμιζε πολύ μια διαδικασία που ακολουθεί ήδη ένα γνωστό εστιατόριο της πόλης το οποίο προβάλλει μέσω μέσου κοινωνικής δικτύωσης τα γεύματα με εικόνες την τιμή και το όνομά τους. Μάλιστα ανέφεραν ότι το συγκεκριμένο εστιατόριο δέχεται πολλές παραγγελίες καθημερινά από αυτούς αλλά και από πολλούς άλλου συνεργάτες τους. Ο χρόνος που είχε παρέλθει από τη στιγμή που έγιναν οι πρώτες συζητήσεις με τους πιθανούς χρήστες μέχρι

τη στιγμή που τους παρουσιάστηκε το τελικό προϊόν είχε λειτουργήσει αρνητικά καθώς ήδη κάποιος άλλος είχε κάνει κάτι παρόμοιο αλλά με τρόπο εύκολο και ανέξοδο και είχε ήδη καθιερωθεί. Άρα διαπιστώθηκε πως δεν είχε σημασία μόνο το να γίνεται όσο το δυνατόν καλύτερη η εφαρμογή και να παρουσιαστεί στο τέλος ολοκληρωμένη όσο το να παρουσιαστεί γρήγορα ώστε να χαραχθεί σαν πρώτη και πρωτότυπη στο μυαλό των χρηστών

Παρατηρήθηκε επίσης πως η πλειοψηφία έδινε περισσότερη σημασία στη σελίδα όπου εμφανίζονται συγκεντρωτικά τα γεύματα της ημέρας και όχι τόσο το εβδομαδιαίο μενού. Παρά το γεγονός ότι στην αρχή είχαν υποστηρίξει ότι θα τους άρεσε σαν ιδέα όταν ήρθαν σε επαφή με αυτό φάνηκε να μην τους ενδιαφέρει τόσο όσο πίστευαν αρχικά. Μάλιστα κάποιοι είπαν πως τους ήταν υπεραρκετό το να βλέπουν συγκεντρωτικά τα γεύματα και ότι θα ήθελαν ίσως και μια επιπλέον επιλογή αναζήτησης όπως για παράδειγμα βάσει εστιατορίου. Επίσης θεώρησαν το ημερήσιο μενού τελείως περιττό, ενώ η αρχική μας πρόθεση ήταν να βοηθήσουμε στην αναζήτηση. Συνεπώς καταλήξαμε πως ακόμα και να μην είχαν υλοποιηθεί καθόλου το εβδομαδιαίο και το ημερήσιο μενού δεν θα ήταν κάτι που θα έλειπε καθώς φάνηκε πως θα αποτελέσουν χαρακτηριστικά που δεν θα χρησιμοποιούνται ιδιαίτερος συχνά. Αν και για την τροποποίηση της φόρμας εγγραφής το κόστος δεν ήταν ιδιαίτερα μεγάλο, η αφαίρεση του εβδομαδιαίου και του ημερήσιου μενού απαιτεί αφαίρεση πάνω από χίλιες γραμμές κώδικα όμως το πιο σημαντικό είναι ότι μεγάλο μέρος της πλατφόρμας βρίσκεται σε άμεση σύνδεση με το εβδομαδιαίο μενού και συνεπώς θα πρέπει να αλλάξει σημαντικό μέρος της λογικής της.

Σε γενικές γραμμές λοιπόν αυτό που έγινε μάθημα μέσω της συγκεκριμένης εφαρμογής είναι ότι η απλότητα είναι πολύ πιο σημαντική για τους χρήστες από τα όποια επιπλέον χαρακτηριστικά. Επίσης κατανοήθηκε πως εν τέλει είχε σπαταληθεί πολύτιμος χρόνος στο να υλοποιηθεί η εφαρμογή, ενώ στο ενδιάμεσο οι χρήστες άλλαζαν γνώμη και κάποιοι άλλοι με πιο απλοϊκούς τρόπους κέρδιζαν έδαφος στη συνείδηση των πιθανών τελικών μας χρηστών.

4.2 Τεχνολογίες που χρησιμοποιήθηκαν

Η υλοποίηση της πλατφόρμας EAT-IN έγινε ως επί τω πλείστον με χρήση διαφόρων projects της "οικογένειας" της Spring. Για την ακρίβεια έγινε χρήση του Spring Boot Framework το οποίο χρησιμοποιεί όλη τη λογική του Spring Framework αφαιρώντας το μεγαλύτερο μέρος των XML ρυθμίσεων που αυτό απαιτεί κάνοντας έτσι την ανάπτυξη και αυτόνομων εφαρμογών γρήγορη και εύκολη. Επιπροσθέτως χρησιμοποιήθηκαν το Spring Security Framework, η Hibernate ORM για την υλοποίηση του Java Persistence API (JPA) ενώ η βάση που χρησιμοποιήθηκε ήταν MySQL βάση δεδομένων. Τέλος έγινε χρήση της μηχανής προτύπων Thymeleaf, του

Bootstrap Framework, και των τεχνολογιών HTML5, Javascript και JQuery. Ακολουθεί μια σύντομη παρουσίαση του Spring Framework προκειμένου να κατανοηθεί που βασίζεται η λειτουργία του Spring Boot καθώς και κάποιον εκ των υπολοίπων τεχνολογιών.

4.2.1 Spring Framework

Το Spring Framework είναι ένα διαδεδομένο περιβάλλον εργασίας ανοιχτού κώδικα (open source) το οποίο αναπτύχθηκε ώστε να διευκολύνει την ανάπτυξη Java EE εφαρμογών. Πρόκειται για ένα "ελαφρύ" πλαίσιο ανάπτυξης λογισμικού το οποίο εμπεριλαμβάνει στον πυρήνα του (core) την αρχή του Inversion of Control (IoC).

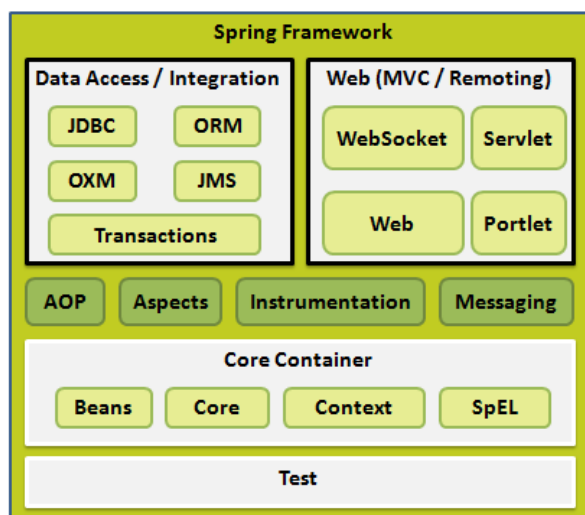
Το IoC, είναι με τεχνική η οποία βοηθάει στην εξαγωγή των εξαρτήσεων που έχουν οι κλάσεις μεταξύ τους. Όταν, δηλαδή, μια κλάση η οποία για να εκτελέσει κάποιες διεργασίες απαιτεί ένα αντικείμενο κάποια άλλης κλάσης, παραδοσιακά, χωρίς τη λογική του IoC, θα έπρεπε μέσα στην πρώτη κλάση να δημιουργήσουμε ένα αντικείμενο της δεύτερης. Με τη χρήση του IoC παρέχεται μέσω κάποιου εξωτερικού μηχανισμού ένα αντικείμενο της δεύτερης κλάσης στην πρώτη κατά τη στιγμή της εκτέλεσης. Στην ουσία η εξάρτηση εισάγεται (inject) εξ' ου και ο όρος IoC είναι συνυφασμένος με την έννοια του Dependency Injection.

Μια Java EE εφαρμογή κατά κανόνα, αποτελείται από αρκετά αντικείμενα τα οποία συνεργάζονται - έχουν εξαρτήσεις το ένα από το άλλο. Αυτά τα αντικείμενα στα πλαίσια του Spring framework, ονομάζονται Spring beans. Το Dependency Injection (DI), είναι μια διαδικασία, μέσω της οποίας τα αντικείμενα, δηλώνουν ποιες εξαρτήσεις έχουν, δηλαδή ποια άλλα αντικείμενα χρειάζονται για να εκτελέσουν επιτυχώς τις μεθόδους τους. Το DI (ή IoC) έχει κεντρικό ρόλο στη λειτουργικότητα του Spring Framework το οποίο διαθέτει τον IoC Container. Ο container αυτός είναι που εκχωρεί (inject) τις εξαρτήσεις (dependencies) κατά την δημιουργία των beans. Είναι εκείνος που διαχειρίζεται ολοκληρωτικά τον κύκλο ζωής των beans από τη δημιουργία τους μέχρι και τη στιγμή της καταστροφής τους και εκείνος που τα συνδέει μεταξύ τους όταν απαιτείται. Τις οδηγίες για το πως και πότε πρέπει να εκτελέσει αυτές τις ενέργειες τις λαμβάνει είτε μέσω XML αρχείου στο οποίο υπάρχει το configuration των beans είτε με χρήση Java Annotations (π.χ @Autowired). Στη Spring το DI μπορεί να γίνει είτε με πέρασμα παραμέτρων στον constructor είτε μέσω μεθόδων setter/getter.

Η λογική του βοηθάει στο να συνδέονται οι κλάσεις μεταξύ τους διατηρώντας ταυτόχρονα την ανεξαρτησία τους δίνοντας τη δυνατότητα να αυξάνει τη δυνατότητα επαναχρησιμοποίησης τους και ευκολότερου τεσταρίσματος.

Το Spring Framework αποτελεί το πρώτο και βασικότερο κομμάτι

της οικογένειας της Spring η οποία μετρά έως τώρα εικοσιτέσσερα projects ανάμεσά τους τα Spring Boot, Spring Security, Spring Data, Spring Social και άλλα με δυνατότητα συνδιασμού αυτών. Όλα όμως αυτά τα projects έχουν ένα κοινό στοιχείο, το ότι εκμεταλλεύονται το Spring Framework και κατά συνέπεια τα τμήματα (modules) που το αποτελούν. Συγκεκριμένα το Spring Framework αποτελείται από είκοσι modules κατανεμημένα στις κατηγορίες που φαίνεται στο Σχήμα 4.21 και στοχεύουν στο να παράσχουν όλα όσα χρειάζεται ένας προγραμματιστής προκειμένου να αναπτύξει enterprise εφαρμογές. Τα modules αυτά μπορούν να συνδιαστούν μεταξύ τους αναλόγως των απαιτήσεων χωρίς φυσικά να χρειάζεται να χρησιμοποιηθούν όλα μαζί.



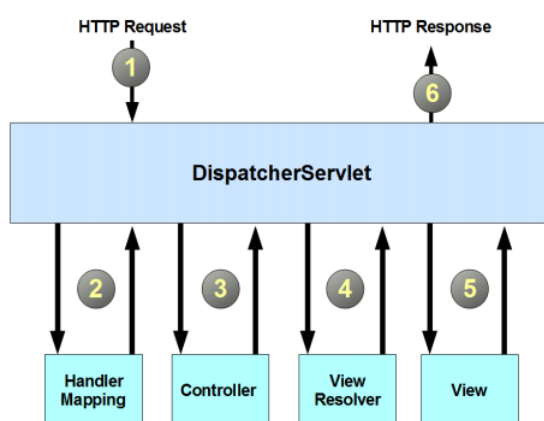
Σχήμα 4.21: Τμήματα της Spring (Spring modules).

Core Container: Τα modules της Spring είναι χτισμένα πάνω από το βασικό της πυρήνα (core container) το οποίο είναι ο θεμελιώδης container και η βάση με την οποία η Spring υλοποιεί το DI. Το βασικότερο “συστατικό” είναι ο BeanFactory και επιτρέπει την διαμόρφωση ενός xml αρχείου το οποίο καθορίζει ποια beans δημιουργούνται και ποια είναι τα χαρακτηριστικά τους. Το πλεονέκτημα του BeanFactory είναι ότι διαχωρίζει το configuration της εφαρμογής και τις προδιαγραφές σχετικές με τις εξαρτήσεις από τον κώδικα της εφαρμογής. Έτσι ο προγραμματιστής μπορεί να ορίσει πως θα διαμορφώνονται τα αντικείμενα και οι μεταξύ τους σχέσεις.

Data Access/Integration: Στην κατηγορία αυτή εμπεριέχονται τεχνολογίες που σχετίζονται με την πρόσβαση και την ενσωμάτωση δεδομένων. Σε αυτό περιλαμβάνονται modules όπως το JDBC τμήμα που παρέχει ένα JDBC στρώμα αφαίρεσης (abstraction) που αφαιρεί την ανάγκη για χρονόβόρο κώδικα σχετικό με το JDBC και το ORM τμήμα παρέχει στρώματα ενσωμάτωσης για δημοφιλή APIs αντικείμενοσχεσιακής χαρτογράφησης, όπως Hibernate, JPA.

Web: Στην κατηγορία αυτή περιέχονται modules που εξυπηρετούν τη δημιουργία εφαρμογών που στηρίζονται στο διαδίκτυο (web - based applications). Το Web-Socket module παρέχει υποστήριξη για αμφίδρομη επικοινωνία μεταξύ πελάτη και εξυπηρετητή στις διαδικτυακές εφαρμογές. Το Web τμήμα παρέχει βασικά χαρακτηριστικά web-oriented ενσωμάτωσης, όπως λειτουργικότητα ανεβάσματος αρχείων (file upload) και αρχικοποίηση του IoC container. Ιδιαίτερο ενδιαφέρον παρουσιάζει το Spring web MVC το οποίο είναι η εφαρμογή του model - view - controller προτύπου σχεδίασης διαδικτυακών εφαρμογών από τη Spring. Το Spring Web MVC έχει σχεδιαστεί γύρω από έναν DispatcherServlet. Ο DispatcherServlet αναλαμβάνει να διεκπεραιώσει όλα τα HTTP αιτήματα που δέχεται. Τα βήματα που γίνονται από τη στιγμή αποστολής ενός HTTP αιτήματος μέχρι την διεκπεραίωση του [Σχήμα 4.22] είναι τα εξής [23]:

- Ο DispatcherServlet δέχεται ένα HTTP αίτημα.
- Συμβουλευτείται το HandlerMapping για να επιλέξει το κατάλληλο Controller
- Ο Controller, ελέγχει την μορφή του αιτήματος (εάν είναι GET ή POST) και αναλόγως καλεί το κατάλληλο Service. Το Service θα δημιουργήσει το Model με όλα τα απαραίτητα δεδομένα που χρειάζονται και στην συνέχεια, θα επιστρέψει το όνομα του View στον DispatcherServlet
- Ο DispatcherServlet θα συμβουλευτεί αυτή την φορά τον ViewResolver, για να επιλέξει το κατάλληλο View που ικανοποιεί το αίτημα που δέχτηκε.
- Ο DispatcherServlet στέλνει τα δεδομένα στο View το οποίο εμφανίζεται στον περιηγητή και εκεί ολοκληρώνεται το αίτημα του χρήστη.



Σχήμα 4.22: Βήματα ολοκλήρωσης HTTP αιτήματος με Spring MVC.

Το Spring MVC είναι συμβατό με πολλές διαφορετικές τεχνολογίες

όπως οι Thymeleaf, Velocity, FreeMarker, JSTL. Επιπροσθέτως διαθέτει υπηρεσία validation η οποία αποτρέπει το χρήστη από το να δώσει λανθασμένα δεδομένα σε μια φόρμα, γλιτώνοντας τον προγραμματιστή από την υλοποίηση ενός συνόλου ελέγχων.

Το Spring Framework λοιπόν είναι ένα ελαφρύ πλαίσιο εργασίας, το ποίο διευκολύνει τους προγραμματιστές να αναπτύξουν υψηλού επιπέδου εφαρμογές με τρόπο που ο κώδικας είναι πιο απλουστευμένος και επαναχρησιμοποιήσιμος και εύκολα ελέγξιμος (test). Τα διάφορα τμήματά του μπορούν να αξιοποιηθούν κατά βούληση και παρέχουν ένα σύνολο δυνατοτήτων διαχείρισης δεδομένων, ανάπτυξης διαδικτυακών εφαρμογών, πιστοποίησης και εξουσιοδότησης που εξασφαλίζουν την ασφάλεια μιας εφαρμογής (security) ενώ ταυτόχρονα μπορεί να συνδυαστούν με πολλές υπάρχουσες τεχνολογίες.

4.2.2 Spring Boot

Το Spring Boot Framework ³ είναι η λύση που δίνει η Spring για να επιτύχει τη λογική του παραδείγματος σχεδιασμού " σύμβαση αντί παραμετροποίησης" (convention-over-configuration). Ουσιαστικά με αυτό επιτυγχάνεται η μείωση των αποφάσεων που πρέπει να πάρει ο προγραμματιστής που χρησιμοποιεί το framework διατηρώντας όμως και την απαιτούμενη ευελιξία. Βασίζεται και αυτό στην ισχυρή λογική του Spring Framework μειώνοντας σημαντικά το απαιτούμενο Spring configuration και απλοποιώντας τις διαδικασίες [Σχήμα 4.23]. Έτσι δίνει τη δυνατότητα στο προγραμματιστή να δημιουργήσει άμεσα και εύκολα εφαρμογές που τρέχουν με τη λιγότερη δυνατή προσπάθεια. Παρέχει μάλιστα και Spring Boot Initializr μέσω του οποίου μπορεί να δημιουργηθεί μια εφαρμογή επιλέγοντας τα Spring τμήματα που απαιτεί η εκάστοτε εφαρμογή και παράγει κώδικα έτοιμο προς χρήση και περαιτέρω επεξεργασία.

³<http://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>



Σχήμα 4.23: Η Spring Boot σε σχέση με το χρήστη και τη Spring.

Όπως φαίνεται και στην παραπάνω εικόνα η Spring Boot βρίσκεται ανάμεσα στο χρήστη και τη Spring. Η θέση αυτή εξηγεί την έννοια της απλούστερης διαδικασίας για τον τελικό χρήστη που συνδυάζεται με τις δυνατότητες της Spring. Επι της ουσίας δεν αποτελεί κάτι εντελώς καινούργιο και δεν υπάρχει προκειμένου να λύσει διαφορετικά προβλήματα. Μπορεί κανείς να πει ότι είναι η Spring αν της αφαιρεθεί η XML παραμετροποίηση(configuration) και περάσει σαν επιπρόσθετο χαρακτηριστικό κάποιους ενσωματωμένους HTTP servers.

Όντας λοιπόν βασισμένη στη Spring κληρονομεί και χαρακτηριστικά της. Αξιοποιεί το Spring web MVC (Model-View-Controller)

Το Spring Boot είναι ένα ισχυρό εργαλείο το οποίο συναδυάζει μια σειρά πλεονεκτημάτων κάποια από τα οποία είναι τα παρακάτω :

- Καθιστά εύκολη την ανάπτυξη Spring based εφαρμογών μειώνοντας το χρόνο ανάπτυξης αυτών και αυξάνοντας την παραγωγικότητα.
- Απαλλάσσει από την ανάγκη για όλο αυτό το XML ή Annotation configuration που συνοδεύει μια κλασική Spring εφαρμογή. Έτσι περιορίζει τα επαναλαμβανόμενα κομμάτια κώδικα που μπορεί να υπήρχαν στην αμιγώς Spring εφαρμογή.
- Είναι εύκολο να τη συνδυάσεις με άλλα projects της Spring οικογένειας όπως Spring ORM, Spring Data, Spring Security και άλλα.
- Διαθέτει ενσωματωμένους HTTP servers όπως ο Apache Tomcat ο οποίος ρυθμίζεται αυτόματα χωρίς απαίτηση για περαιτέρω παραμετροποίηση.
- Διαθέτει ισχυρά starter POMs που διευκολύνουν το Maven configuration

ενώ ταυτόχρονα απαλλάσσει από την ανάγκη για ορισμό έκδοσης για την εκάστοτε εξέλιξη. Ένα σύνολο βιβλιοθηκών είναι γρήγορα προσβάσιμες.

- Παρέχει αυτόματη ρύθμιση της Spring όταν απαιτείται και γενικότερα αυτοματοποιεί τις διαδικασίες παραμετροποίησης (configuration) για μια σειρά web λειτουργιών (JPA, Security basic, Spring MVC).
- Παρέχει πολλά plugins προκειμένου να γίνει εύκολα χρήση ενσωματωμένων και in-memory βάσεων δεδομένων.

4.2.3 Hibernate

Για την διατήρηση των δεδομένων (persistence) στην Java, στα πλαίσια των Enterprise Java Beans 3, έχει οριστεί μία τεχνολογία που ονομάζεται Java Persistence API. Η τεχνολογία αυτή ορίζει ένα περιβάλλον εργασίας (framework) για την αντιστοίχιση οντοκεντρικών μοντέλων (object-oriented models) σε σχήματα παραδοσιακών σχεσιακών βάσεων δεδομένων. Κεντρικός στόχος για την ανάπτυξη της υπήρξε ο ορισμός ενός ενιαίου τρόπου για την διατήρηση δεδομένων σε όλες τις Java εφαρμογές. Η Hibernate αποτελεί μία βιβλιοθήκη ελεύθερου λογισμικού για τη γλώσσα προγραμματισμού Java, η οποία υλοποιεί την διεπαφή που ορίζεται στα πλαίσια του Java Persistence API. Είναι μία ολοκληρωμένη λύση στο πρόβλημα της διαχείρισης δεδομένων εκτελώντας όλες τις ενέργειες μεταξύ των εφαρμογών και των σχεσιακών βάσεων δεδομένων απαλλάσσοντας τον προγραμματιστή από επιπρόσθετο κόπο και τον κίνδυνο αστοχιών στην αποτύπωση σχέσεων μεταξύ κλάσεων και σχήματος βάσης. Ο κεντρικός στόχος αυτής είναι η δημιουργία μίας διεπαφής (interface) μεταξύ των διαδεδομένων σχεσιακών βάσεων δεδομένων και του αντικειμενοστραφούς προγραμματισμού. Με άλλα λόγια, επιτρέπει τη χρήση μίας σχεσιακής βάσης δεδομένων ως αντικειμενοστραφή. Για την επίτευξη αυτού, δημιουργούνται αντιστοιχίες μεταξύ των εννοιών του αντικειμενοστραφούς προγραμματισμού, όπως οι συσχετίσεις, η κληρονομικότητα και ο πολυμορφισμός (τα οποία δεν υπάρχουν σε μία σχεσιακή βάση δεδομένων), και των πινάκων και σχέσεων μεταξύ αυτών μίας σχεσιακής βάσης. Με αυτόν τον τρόπο ο προγραμματιστής βλέπει τελικά μία αντικειμενοστραφή βάση δεδομένων, παρόλο που στην ουσία χρησιμοποιεί μία σχεσιακή. Έτσι ο προγραμματιστής χρησιμοποιεί τα αντικείμενα της συγκεκριμένης εφαρμογής, τα τροποποιεί σχετικά με τη λογική της εφαρμογής που αναπτύσσει και τα αποθηκεύει (τροποποιεί, διαγράφει και αναζητά) στη βάση ως αντικείμενα. Σκέπτεται, δηλαδή, με αντικειμενοστραφείς έννοιες και όχι με βάση το σχήμα της σχεσιακής βάσης δεδομένων. Έτσι η Hibernate, γνωρίζοντας την αντιστοιχία μεταξύ βάσης και λογικής της εφαρμογής, αναλαμβάνει να κατασκευάσει τον κατάλληλο κώδικα τον οποίο στέλνει τελικά στη βάση δεδομένων. Έπειτα τα αποτελέσματα που επιστρέφονται από τη βάση στην Hibernate δίνονται στην εφαρμογή ως αντικείμενα. Πρόκειται, δηλαδή, για

ένα ενδιάμεσο επίπεδο μεταξύ εφαρμογής και βάσης δεδομένων.

Η χρήση της Hibernate μπορεί να οφελήσει προσφέροντας τα εξής:
Παραγωγικότητα: Στην ανάπτυξη λογισμικού ένα μεγάλο μέρος της προγραμματιστικής προσπάθειας αφιερώνεται στην διεπαφή της εφαρμογής με τη βάση δεδομένων. Το Hibernate αυτοματοποιώντας τις βασικές λειτουργίες Δημιουργία/Ανάγνωση/Τροποποίηση/Διαγραφή (CRUD – Create Read Update Delete) επιτρέπει αρχικά στον προγραμματιστή να επικεντρώνει την προσπάθειά του στη λογική της εφαρμογής (business logic).

Συντηρησιμότητα: Με τη χρήση του Hibernate γράφονται σημαντικά λιγότερες γραμμές κώδικα και ο κώδικας είναι πιο κατανοητός και καλογραμμένος. Αυτό κάνει την συντήρηση της εφαρμογής ευκολότερη.

Ανεξαρτησία από τη βάση δεδομένων: Με τη συμβατότητα του Hibernate με διαφορετικές βάσεις δεδομένων και τη δυνατότητα σύνδεσής του με τη βάση μέσω δηλώσεων οριζόμενων σε ειδικό αρχείο η αναπτυσσόμενη εφαρμογή μπορεί με ελάχιστες τροποποιήσεις να χρησιμοποιηθεί με βάσεις δεδομένων διαφορετικών κατασκευαστών. Το γεγονός αυτό στερεί μεν από το Hibernate την εκμετάλλευση των ιδιαίτερων χαρακτηριστικών της χρησιμοποιούμενης βάσης, όμως, και σε αυτή την περίπτωση, δίνεται η δυνατότητα χρήσης πηγαίας SQL μέσα στο Hibernate που εκμεταλλεύεται τα ιδιαίτερα αυτά χαρακτηριστικά.

4.2.4 Λοιπές τεχνολογίες

Bootstrap: Το Bootstrap ⁴ είναι μια συλλογή εργαλείων (Framework) ανοιχτού κώδικα (open source) για τη δημιουργία ιστοσελίδων και διαδικτυακών εφαρμογών. Περιέχει HTML και CSS για τις μορφές τυπογραφίας, κουμπιά πλοήγησης και άλλων στοιχείων του περιβάλλοντος, καθώς και προαιρετικές επεκτάσεις JavaScript. Είναι το πιο δημοφιλές πρόγραμμα στο GitHub ⁵. Αναπτύχθηκε από τον Mark Otto και τον Jacob Thornton στο Twitter ως ένα πλαίσιο για την ενθάρρυνση της συνέπειας στα εσωτερικά εργαλεία. Πολύ σημαντικό είναι ότι υποστηρίζει- από την έκδοση 2.0 - ανταποκρισιμο σχεδιασμό (responsive design). Αυτό σημαίνει ότι η διάταξη των ιστοσελίδων προσαρμόζεται δυναμικά, λαμβάνοντας υπόψη τα χαρακτηριστικά της συσκευής που χρησιμοποιείται (PC, tablet, κινητό τηλέφωνο). Έχει σχετικά ελλιπή υποστήριξη για HTML5 και CSS, αλλά είναι συμβατό με όλους τους φυλλομετρητές (browsers). Διαθέτει πολλά στατικά JavaScript σε μια μορφή jQuery plugin. Παρέχουν πρόσθετη διεπαφή χρήστη με στοιχεία όπως παράθυρα διαλόγου, επεξηγήσεις, και καρουσέλ.

⁴<https://el.wikipedia.org/wiki/Bootstrap>

⁵ «GitHub: Search Stars>1»

Thymeleaf: Η Thymeleaf ⁶ είναι μια Java XML/XHTML/HTML5 μηχανή προτύπων (template engine) η οποία λειτουργεί στο view layer μια εφαρμογής βασιζόμενης στο MVC μοντέλο. Είναι λογισμικό ανοιχτού κώδικα (open source software) το οποίο παρέχει πλήρη ενσωμάτωση με το Spring Framework και στοχεύει στην πλήρη αντικατάσταση του JSP.

4.3 Συμπεράσματα-Συζήτηση

Μεσα από τη διαδικασία του σχεδιασμού, της υλοποίησης και της αξιολόγησης της web 2.0 πλατφόρμας EAT-IN αποκτήσαμε εμπειρίες οι οποίες υποδεικνύουν πως η χρήση μεθόδων διαδικασιών οι οποίες έχουν έναν πιο ολοκληρωτικό χαρακτήρα στην ανάπτυξη λογισμικού δεν είναι ίσως οι κατάλληλες όταν το προς υλοποίηση προϊόν απευθύνεται σε μεγάλο κοινό και όχι σε κάποιον πελάτη ο οποίος δεσμεύεται μέσω κάποιου συμβολαίου. Ιδίως στην περίπτωση όπου στόχος είναι γενικότερα οι χρήστες του διαδικτύου ή έστω μια μικρότερη ομάδα αυτών εκεί η παραδοσιακές μέθοδοι δείχνουν να είναι ανεπαρκείς.

Στις μέρες μας όπου η χρήση του διαδικτύου και των web 2.0 εφαρμογών είναι καθημερινή συνήθεια, οι τελικοί χρήστες είναι πλέον υποψιασμένοι και πιο απαιτητικοί σε σχέση με παλαιότερα χρόνια. Επίσης όλο και πιο πολλοί έχουν τη δυνατότητα να δραστηριοποιηθούν στην ανάπτυξη εφαρμογών. Συνεπώς το καταναλωτικό κοινό παραμένει περίπου ίδιο αριθμητικά αλλά αυξάνουν οι γνώσεις τους, ενώ ταυτόχρονα οι πάροχοι αυξάνουν άρα και ο ανταγωνισμός. Η υπερίσχυση λοιπόν βασιζέται σημαντικά στην κατανόηση των αναγκών του τελικού πελάτη και στη συνεχή ενσωμάτωσή του κατά τη διαδικασία ανάπτυξης του όποιου προϊόντος λογισμικού. Επίσης καθώς οι απαιτήσεις αλλάζουν πολύ γρήγορα απαιτείται και γρήγορη και εύκολη προσαρμογή.

Όπως είδαμε και στην περίπτωση ανάπτυξης της πλατφόρμας που παρουσιάστηκε στο κεφάλαιο αυτό ακόμα και η χρήση ισχυρών τεχνολογιών που διευκολύνουν τη διαδικασία της υλοποίησης αν δεν συνδυαστεί σωστά με τις απαιτήσεις και τις ανάγκες του τελικού πελάτη μπορεί εύκολα να οδηγήσει σε αποτυχία. Οι αρχές όμως των παραδοσιακών μεθόδων περιορίζουν τρόπον τινά τη συμμετοχή του τελικού πελάτη με αποτέλεσμα να μην βοηθούν πολύ στην ανάπτυξη μια πελατοκεντρικής προσέγγισης η οποία φαντάζει η μόνη λύση προκειμένου να αποφεύγονται δυσάρεστες εκπλήξεις, χάσιμο χρόνου και μη έγκαιρο εντοπισμό της πιθανής αποτυχίας. Στη περίπτωση της EAT-IN πλατφόρμας τα προβλήματα που παρουσιάστηκαν κατά τη διάρκεια αξιολόγησης της από τους πιθανούς χρήστες θα μπορούσαν να είχαν αποφευχθεί εάν υπήρχε το στοιχείο της πελατοκεντρικότητας και της συχνής διάδρασης με τους τελικούς χρήστες.

⁶<https://en.wikipedia.org/wiki/Thymeleaf>

Μέρος III

ΕΥΕΛΙΚΤΕΣ ΜΕΘΟΔΟΙ

ΚΕΦΑΛΑΙΟ

5

ΕΥΕΛΙΚΤΕΣ ΜΕΘΟΔΟΙ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ

5.1 Εισαγωγή

5.1.1 Ανάγκη για ταχεία Ανάπτυξη Λογισμικού

Όπως μελετήθηκε στο προηγούμενο κεφάλαιο μέχρι και τη δεκαετία του 90' οι "βαριές" ή "παραδοσιακές" μέθοδοι ανάπτυξης λογισμικού κατείχαν εξέχουσα θέση. Όμως ήδη ήταν εμφανή τα προβλήματα που η χρήση τέτοιων μεθοδολογιών δημιουργούσε. Η βασική τους αδυναμία εντοπιζόταν στη ανικανότητά τους για αποδοχή και ενσωμάτωση της αλλαγής. Έτσι αυτή η προσέγγιση άρχισε να χαρακτηρίζεται ως ανελαστική, μη παραγωγική, και γραφειοκρατική λόγω του μεγάλου όγκου εγγράφων και εγχειριδίων που τη συνοδεύουν. Θεωρούνται μη παραγωγικές διότι οι φάσεις ανάπτυξης του λογισμικού είναι πολλές και μεγάλης διάρκειας επιβραδύνοντας τον συνολικό χρόνο ανάπτυξης. Αυτή είναι μία σημαντική ενδογενής αδυναμία καθιστούσε της παραδοσιακές μεθόδους ανελαστικές και ανίκανες να αντιδράσουν στις αλλαγές που προκύπτουν, όπως για παράδειγμα στις αλλαγές των απαιτήσεων των πελατών που είναι ένα συνηθισμένο γεγονός. Συνεπώς η ανάγκη για μεθόδους οι οποίες θα ήταν πιο ευέλικτες και ελαστικές ώστε να επιτυγχάνεται ταχεία ανάπτυξη λογισμικού είχε πλέον έρθει στο προσκήνιο.

Έχει παρατηρηθεί πως οι επαγγελματίες ανάπτυξης λογισμικού προσπαθούν συνεχώς να αυξήσουν την παραγωγικότητα και παράλληλα να

διατηρήσουν την ποιότητα σε ψηλά επίπεδα [28]. Η παραπάνω διαπίστωση είναι ένας από τους σημαντικότερους λόγους που οδήγησε τις επιχειρήσεις ανάπτυξης λογισμικού στη συνεχή αναζήτηση νέων μεθόδων που θα τις βοηθήσουν στην υλοποίηση των έργων τους. Μια ακόμη ανάγκη που θεωρείται ισχυρό κίνητρο, είναι τα προβλήματα που προκύπτουν σχετικά με ανικανοποίητες απαιτήσεις των χρηστών και τα ελλειπή λειτουργικά συστήματα.

Το χρονοδιάγραμμα ενός έργου ορίζει πότε θα παραδοθούν συγκεκριμένα μέρη του λογισμικού, στο κάθε ένα από τα οποία θα ικανοποιούνται συγκεκριμένες απαιτήσεις. Παρατηρείται στην πράξη πως οι απαιτήσεις δεν μπορούν να προδιαγραφούν πλήρως από την αρχή του έργου και για αυτό τον λόγο οι ομάδες ανάπτυξης λογισμικού έπρεπε να βρουν τρόπους να διαχειριστούν τα τμήματα του λογισμικού για τα οποία οι απαιτήσεις δεν έχουν καθοριστεί πλήρως ή καθόλου. Ένας ακόμη σημαντικός κίνδυνος προκύπτει όταν οι απαιτήσεις ενδέχεται να τροποποιηθούν ή να αλλάξουν ριζικά, γεγονός που οδηγεί στον επαναπρογραμματισμό του έργου. Αυτό συνεπάγεται, βέβαια, περισσότερο κόστος και χρόνο. Ο κίνδυνος αυτός καθιστά επιτακτική την ανάγκη για εύρεση και θεμελίωση μεθόδων ώστε να μπορούν να εντοπιστούν οι πραγματικές ανάγκες του πελάτη εξ' αρχής. Επίσης οι γρήγοροι ρυθμοί με τους οποίους αλλάζουν οι απαιτήσεις δημιουργούν την ανάγκη οι επιχειρήσεις να μπορούν να προσαρμόζονται εύκολα στις συνεχώς μεταβαλλόμενες τεχνολογίες. Έτσι, οι ομάδες ανάπτυξης λογισμικού αναζητούν μια πιο ευέλικτη μέθοδο η οποία θα προβλέπει τις αστάθειες στις απαιτήσεις.

Με την αλλαγή της χιλιετίας εξελίχθηκε μία νέα ομάδα μεθόδων η οποία με την ιδρυτική της διακήρυξη το 2001 (Agile Manifesto ¹), (Agile Alliance, 2001) και με την ονομασία ευέλικτες μέθοδοι (Agile Methods) υπόσχεται προσαρμοστικότητα και ανταπόκριση στις αλλαγές, παραγωγικότερες πρακτικές και λιγότερη γραφειοκρατία. Δημιουργήθηκαν σαν απάντηση σε μια οικονομία που συνεχώς εξελισσόταν και οι απαιτήσεις αυξάνονταν [44]. Οι μέθοδοι αυτές επέτρεπαν στην ομάδα ανάπτυξης να εστιάζει στο ίδιο το λογισμικό, και όχι στο σχεδιασμό ή την τεκμηρίωσή του. Οι ευέλικτες μέθοδοι βασίζονται σε μια επαναληπτική προσέγγιση για την εξαγωγή των προδιαγραφών, την ανάπτυξη και την παράδοση του λογισμικού, και σχεδιάστηκαν κυρίως με στόχο την υποστήριξη της ανάπτυξης επιχειρηματικών εφαρμογών όπου οι απαιτήσεις συστήματος συνήθως μεταβάλλονται γρήγορα κατά τη διαδικασία ανάπτυξης. Στοχεύουν στη γρήγορη παράδοση λειτουργικού λογισμικού στους πελάτες, οι οποίοι μπορούν στη συνέχεια να προτείνουν νέες και τροποποιημένες απαιτήσεις που θα συμπεριληφθούν στις επόμενες επαναλήψεις του συστήματος. Το όνομα ευέλικτες αναφέρεται κυρίως στην συνολική ικανότητα να προσαρμόζονται κατάλληλα τη διαδικασία ανάπτυξης, όταν προκύπτουν αλλαγές στην πορεία του έργου. Οι ευέλικτες μέθοδοι ενσωμάτωσαν μια ευρεία

¹<http://agilemanifesto.org/>

συλλογή από καλές και δοκιμασμένες αξίες και πρακτικές που βοηθούν στην ανάπτυξη ποιοτικού λογισμικού.

Η ανάπτυξη του λογισμικού γίνεται σε σύντομους επαναληπτικούς (Iterative) και αυξητικούς (Incremental) κύκλους, παρέχοντας την δυνατότητα της προσαρμογής και αντίδρασης στις αλλαγές που τίθενται από το διαρκώς μεταβαλλόμενο επιχειρησιακό περιβάλλον. Με σαφή προτίμηση στην επικοινωνία πρόσωπο με πρόσωπο για τον περιορισμό των παραγόμενων εγγράφων και σε στενή συνεργασία με τον πελάτη οι κατασκευαστές και οι διαχειριστές του έργου επικεντρώνονται στην ανάπτυξη λογισμικού, που θεωρείται το πρωταρχικό μέτρο προόδου. Σαν προσαρμοσμένες (adaptive) μέθοδοι και όχι προφητικές, όπως οι παραδοσιακές μέθοδοι, καλωσορίζουν τις αλλαγές ιδιαίτερα στις απαιτήσεις των πελατών, που είναι ευμετάβλητες και αβέβαιες. Οι ευέλικτες μέθοδοι είναι προσανατολισμένες στους ανθρώπους και όχι στις διαδικασίες, όπως οι παραδοσιακές μέθοδοι και βασίζονται πολύ στην ομαδική εργασία. Υπόσχονται ποιοτικότερο λογισμικό και ταχύτερη ανάπτυξη. Οι σημαντικότερες ευέλικτες μέθοδοι είναι οι εξής: Extreme Programming (XP) (Beck 2000), Dynamic Systems Development Method (DSDM) (Stapleton and MacDonald, 1997), SCRUM (Schwaber 2001), Feature Driven Design (FDD) (Coad et al. 1999), CrystalClear (Cockburn 2001; Cockburn 2004), και Agile Modelling (Ambler 2002).

Η πιο διαδεδομένη από αυτές τις μεθόδους είναι ο ακραίος προγραμματισμός (Extreme Programming - XP) ο οποίος δημιουργήθηκε για να αντιμετωπίσει τόσο τα προβλήματα, που προκαλούνται από τις μεταβολές των απαιτήσεων, όσο και εκείνα που προέρχονται από τους κινδύνους στα έργα λογισμικού. Ο ακραίος προγραμματισμός βασίζεται σε ένα σύνολο από αρχές και πρακτικές, που υποστηρίζουν την ολοκλήρωση του έργου μέσα στα προβλεπόμενα χρονικά όρια, ενώ παράλληλα επιτρέπουν στους υπεύθυνους ανάπτυξης του λογισμικού να ανταποκρίνονται στις μεταβαλλόμενες απαιτήσεις του πελάτη καθ' όλη τη διάρκεια του κύκλου παραγωγής. Δίνοντας ιδιαίτερη έμφαση στην ομαδική εργασία οι πελάτες, διαχειριστές και κατασκευαστές του λογισμικού συμμετέχουν όλοι σε μία ομάδα με στόχο την παραγωγή λογισμικού υψηλής ποιότητας και την ικανοποίηση του πελάτη. Ένας από τους κυριότερους λόγους της επιτυχίας της μεθόδου και της ευρείας της αποδοχής στην βιομηχανία λογισμικού είναι ακριβώς ο στόχος της ικανοποίησης του πελάτη.

5.1.2 Αρχές - Αξίες - Ιδιότητες Ευέλικτων Μεθόδων

Eρελιξία (agility) είναι η ικανότητα της προσαρμογής και επαναπροσδιορισμού ενός αναπτυσσόμενου και συνεχώς εξελισσόμενου συστήματος, στην περίπτωση που εμφανίζονται αλλαγές στις αρχικές θεωρήσεις και παραδοχές. Οι οργανισμοί, που χρησιμοποιούν ευέλικτες μεθόδους, βλέπουν την αλλαγή σαν ευκαιρία βελτίωσης και προόδου και όχι σαν

απειλή. Οι ευέλικτες μέθοδοι είναι:

- **Επαναληπτικές (iterative):** Αρχικά παραδίδεται ένα πλήρες σύστημα και στη συνέχεια γίνονται αλλαγές στη λειτουργία κάθε υποσυστήματος σε κάθε νέα έκδοση.
- **Αυξητικές (incremental):** Το σύστημα, όπως ορίζεται στις απαιτήσεις, χωρίζεται σε υποσυστήματα με βάση τη λειτουργία τους. Νέες λειτουργίες προστίθενται σε κάθε νέα έκδοση.
- **Αυτο-διοργανούμενες (self-organizing):** Η ομάδα έχει την αυτονομία να οργανώνεται έτσι, ώστε να επιτύχει την βέλτιστη ολοκλήρωση των υποσυστημάτων με τον καλύτερο δυνατό τρόπο.
- **Προκύπτουσες (emergent):** Οι απαιτήσεις και η τεχνολογία, που θα χρησιμοποιηθεί, προκύπτουν κατά τη διάρκεια του κύκλου ανάπτυξης.

Παρόλο που αυτές οι μέθοδοι βασίζονται στην ιδέα της αυξανόμενης ανάπτυξης και παράδοσης λογισμικού, προτείνουν διαφορετικές διεργασίες για να το πετύχουν. Ωστόσο, μοιράζονται ένα κοινό σύνολο αρχών και για αυτό έχουν πολλά κοινά σημεία. Οι τέσσερις βασικές αρχές που καθορίζουν τις ευέλικτες μεθόδους είναι ¹:

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

Πρώτον, τονίζεται η ανάγκη για συνεργασία μεταξύ των συμμετεχόντων σε ένα έργο λογισμικού, κάτι που αποτελεί τον σημαντικότερο παράγοντα κατά την ανάπτυξη λογισμικού. Οι διαδικασίες και τα εργαλεία δεν πρόκειται να αποτελέσουν σημαντική βοήθεια, αν η συνεργασία αυτή δεν είναι επιτυχής.

Δεύτερον, το μανιφέστο υπογραμμίζει τον πρωταρχικό στόχο του έργου, ο οποίος είναι η παραγωγή λειτουργικού λογισμικού. Οι προγραμματιστές προτρέπονται να διατηρήσουν τον κώδικα όσο πιο απλό γίνεται, με αποτέλεσμα το ίδιο το λογισμικό να είναι πιο κατανοητό σε σχέση με μακροσκελή έγγραφα και διαγράμματα τεκμηρίωσης. Η ολοκληρωμένη τεκμηρίωση είναι σημαντική για να εξηγεί το στόχο της εφαρμογής, αλλά ο ρόλος της θα είναι πάντα δευτερεύων και συμπληρωματικός.

Τρίτον, η συνεργασία και η επικοινωνία της ομάδας ανάπτυξης του λογισμικού με τον πελάτη προτιμάται σε σύγκριση με αυστηρά συμβόλαια

¹<http://agilemanifesto.org/>

και συμβάσεις. Η συνεργασία αυτή είναι συνήθως δύσκολη, αφού ο πελάτης αλλάζει συχνά γνώμη σχετικά με τις απαιτήσεις της εφαρμογής. Η ύπαρξη ενός συμβολαίου είναι σημαντική, ωστόσο δεν μπορεί να υποκαταστήσει την επικοινωνία.

Τέταρτον, οι συμμετέχοντες στο έργο πρέπει να είναι καλά πληροφορημένοι, έτοιμοι να εξετάσουν και να αντιμετωπίσουν τις ανάγκες που θα εμφανιστούν κατά τη διάρκεια του έργου. Οι απαιτήσεις των πελατών, το επιχειρηματικό περιβάλλον ακόμα και η ίδια η τεχνολογία, μεταβάλλονται συνεχώς και δεν αφήνουν ανεπηρέαστη την ανάπτυξη. Για αυτό θα πρέπει η ανάπτυξη να ικανοποιεί άμεσα την ανάγκη για αλλαγές. Αυτό δεν σημαίνει ότι απορρίπτεται το σχέδιο χρονοπρογραμματισμού αλλά τα σχέδια θα πρέπει να είναι περισσότερο ευέλικτα [30].

Στη διακήρυξη των ευέλικτων μεθόδων καθορίστηκαν επιπλέον δώδεκα αρχές που καθοδηγούν τον τρόπο ανάπτυξης:

Οι 12 αρχές του ευέλικτου μανιφέστο

1. Βασική προτεραιότητα αποτελεί η ικανοποίηση του πελάτη με την συνεχή παράδοση σημαντικού τμήματος του λογισμικού από τα πρώτα κιόλας στάδια της παραγωγής.
2. Οι μεταβαλλόμενες απαιτήσεις είναι καλοδεχούμενες ακόμα και σε προχωρημένο στάδιο ανάπτυξης. Οι διαδικασίες, που εφαρμόζονται έχουν την δυνατότητα να περιλαμβάνουν την αλλαγή προς όφελος του πελάτη.
3. Η παράδοση τμημάτων λογισμικού γίνεται ανά δύο εβδομάδες έως και ανά δύο μήνες. Σημαντική είναι η όσο το δυνατόν συχνότερη παράδοση.
4. Οι πελάτες και οι υπεύθυνοι ανάπτυξης του συστήματος πρέπει να συνεργάζονται καθημερινά μέχρι την παραγωγή του τελικού προϊόντος.
5. Σημαντική είναι η ανάθεση των τμημάτων του συστήματος σε ικανό και αποτελεσματικό προσωπικό και η εξασφάλιση ευνοϊκού περιβάλλοντος εμπιστοσύνης και υποστήριξης.
6. Ο καλύτερος τρόπος μεταβίβασης και ανταλλαγής πληροφοριών με την ομάδα παραγωγής είναι η διαπροσωπική συζήτηση και οι συνολικές.
7. Το καλύτερο και πιο αξιόπιστο μέτρο, που επιβεβαιώνει την πρόοδο, είναι το να λειτουργούν σωστά τα τμήματα λογισμικού τα οποία κατασκευάζονται.
8. Οι ευέλικτες διαδικασίες προωθούν τον σταθερό ρυθμό ανάπτυξης του συστήματος, ο οποίος πρέπει να ακολουθείται τόσο από τον πελάτη όσο και από τους υπεύθυνους παραγωγής.
9. Η ευελιξία ενισχύεται από την συνεχή προσπάθεια για τεχνική αρ-

τιότητα και καλό σχεδιασμό.

10. Η απλοποίηση, με την έννοια της υλοποίησης στόχων με σύντομο αλλά αποτελεσματικό τρόπο, είναι ουσιαστική.
11. Οι καλύτερες αρχιτεκτονικές, απαιτήσεις και σχέδια προκύπτουν από ομάδες, που αυτό-οργανώνονται.
12. Σε τακτά χρονικά διαστήματα, η ομάδα συζητά τρόπους, που την βοηθούν να γίνει περισσότερο αποτελεσματική και επαναπροσδιορίζει τη συμπεριφορά της.

Η τήρηση των βασικών αυτών αρχών του μανιφέστο μειώνουν την πιθανότητα να εμφανιστούν κίνδυνοι και λάθη κατά την διαδικασία ανάπτυξης του λογισμικού. Συνοψίζοντας, λοιπόν, οι ευέλικτες μέθοδοι ενσωμάτωσαν μια ευρεία συλλογή από καλές και δοκιμασμένες αξίες και πρακτικές που βοηθούν στην ανάπτυξη ποιοτικού λογισμικού. Η ανάπτυξη του λογισμικού γίνεται σε σύντομους επαναληπτικούς (iterative) και αυξητικούς (incremental) κύκλους, παρέχοντας την δυνατότητα της προσαρμογής και αντίδρασης στις αλλαγές που τίθενται από το διαρκώς μεταβαλλόμενο επιχειρησιακό περιβάλλον [24].

Με σαφή προτίμηση στην πρόσωπο με πρόσωπο επικοινωνία για τον περιορισμό των παραγομένων εγγράφων και σε στενή συνεργασία με τον πελάτη οι κατασκευαστές και οι διαχειριστές του έργου επικεντρώνονται στην ανάπτυξη λογισμικού, που θεωρείται το πρωταρχικό μέτρο προόδου. Σαν προσαρμόσιμες (adaptive) μέθοδοι και όχι προφητικές, όπως οι παραδοσιακές μέθοδοι, καλωσορίζουν τις αλλαγές ιδιαίτερα στις απαιτήσεις των πελατών.

5.1.3 Τα χαρακτηριστικά των ευέλικτων μεθόδων

”Η καινοτομία για τις ευέλικτες μεθόδους δεν είναι οι πρακτικές που χρησιμοποιούν, αλλά η αναγνώριση τους από τους ανθρώπους ως τους κύριους συντελεστές της επιτυχίας του έργου, σε συνδυασμό με την έμφαση στην αποτελεσματικότητα και την ευελιξία” [24].

Μια μέθοδος μπορεί να θεωρηθεί ως ευέλικτη όταν είναι επαυξητική (παραδίδονται μικρά τμήματα του λογισμικού, σε τακτά χρονικά διαστήματα), συνεταιριστική (υπάρχει επαρκής επικοινωνία μεταξύ της ομάδας ανάπτυξης και των πελατών), απλή (είναι εύκολη, κατανοητή και τροποποιήσιμη) και προσαρμοστική (μπορεί εύκολα να προσαρμόζεται σε απόβλεπτες αλλαγές).

Οι ακόλουθες αρχές των ευέλικτων πρακτικών θεωρούνται ως οι κύριες διαφορές μεταξύ ευέλικτων και παραδοσιακών μεθόδων.

Προσανατολισμένες στον άνθρωπο. Οι ευέλικτες μέθοδοι θεωρούν τους ανθρώπους – πελάτες, προγραμματιστές και γενικά τους συμμετέχοντες – ως το σημαντικότερο παράγοντα στην διαδικασία ανάπτυξης λογισμικού.

Προσαρμοστικότητα. «Οι εξωτερικές περιβαλλοντικές αλλαγές προκαλούν κρίσιμες παραλλαγές. Επειδή δεν μπορούμε να περιορίσουμε αυτές τις αλλαγές, η προσπάθεια ώστε να αντιμετωπιστούν με το λιγότερο δυνατό κόστος, είναι η μόνη βιώσιμη στρατηγική». Οι συμμετέχοντες σε μια ευέλικτη διαδικασία δεν φοβούνται τις αλλαγές αλλά τις καλωσορίζουν σε όλα τα στάδια του έργου. Αντιμετωπίζουν τις αλλαγές στις απαιτήσεις θετικά, γιατί σημαίνει ότι η ομάδα έχει κατανοήσει καλύτερα τις ανάγκες του πελάτη και συνεπώς γνωρίζει πώς να τις ικανοποιήσει και να καταλήξει σε ένα ολοκληρωμένο προϊόν [26]. Στις ευέλικτες μεθόδους, σκοπός δεν είναι να περιοριστούν οι αλλαγές που παρουσιάζονται, αλλά η εύρεση τρόπων που θα επιτρέψει τον καλύτερο χειρισμό τους.

Έμφαση στην πραγματικότητα. Οι ευέλικτες μέθοδοι δίνουν έμφαση στα δεδομένα σε κάθε φάση της ανάπτυξης και όχι στην τήρηση ενός καλά ορισμένου σχεδίου. Κάθε επανάληψη προσθέτει αξία στο συνεχώς αναπτυσσόμενο προϊόν. Η απόφαση για το αν η επιχειρηματική αξία έχει προστεθεί ή όχι δεν θα δοθεί από τους προγραμματιστές, αλλά από τους τελικούς χρήστες και τους πελάτες.

Εξισορρόπηση ευελιξίας και σχεδιασμού. Ο χρονικός προγραμματισμός είναι σημαντικός, αλλά το πρόβλημα είναι ότι τα έργα λογισμικού δεν μπορούν να προβλεφθούν με ακρίβεια γιατί επηρεάζονται από πολλούς παράγοντες που πρέπει να ληφθούν υπόψη. Μια καλύτερη στρατηγική είναι να γίνονται λεπτομερή σχέδια για το άμεσο μέλλον, για τις επόμενες εβδομάδες, λιγότερο αναλυτικά σχέδια για τους επόμενους μήνες, και λιγότερα σχέδια πέρα από αυτό το χρονικό διάστημα. Η ευελιξία στην αλλαγή των αποφάσεων είναι σημαντική για ένα έργο λογισμικού. Μια απόφαση δεν πρέπει να παρθεί άμεσα αλλά να αναζητηθεί ένας τρόπος ώστε να αναβληθεί για αργότερα ή να παρθεί με τρόπο ώστε να εξασφαλιστεί το ενδεχόμενο αλλαγής της αργότερα χωρίς δυσκολία [26].

Εμπειρική διαδικασία. Οι ευέλικτες μέθοδοι ανάπτυξης λογισμικού είναι εμπειρικές διαδικασίες. Η ανάπτυξη λογισμικού δεν μπορεί να θεωρηθεί καθορισμένη διαδικασία γιατί πολλές αλλαγές προκύπτουν κατά τη διάρκεια της ανάπτυξης. Είναι εξαιρετικά απίθανο οποιοδήποτε σύνολο προκαθορισμένων βημάτων να οδηγήσουν σε ένα επιθυμητό, προβλέψιμο αποτέλεσμα, επειδή οι απαιτήσεις αλλάζουν, παρουσιάζονται αλλαγές στην χρησιμοποιούμενη τεχνολογία, άνθρωποι προστίθενται και αφαιρούνται από την ομάδα ανάπτυξης, και ούτω καθεξής [27].

Αποκεντρωμένη προσέγγιση (decentralized approach). Η αποκεντρωμένη διαχείριση επηρεάζει σημαντικά ένα σύστημα λογισμικού μέσω της εξοικονόμησης περισσότερου χρόνου από ότι σε μια διαδικασία όπου οι λήψεις αποφάσεων περιορίζονται σε ένα πρόσωπο. Η ευέλικτη ανάπτυξη λογισμικού ενθαρρύνει την λήψη αποφάσεων από τους προγραμματιστές, χωρίς αυτό να σημαίνει ότι οι προγραμματιστές θα αναλάβουν το ρόλο της διοίκησης. Η διοίκηση παραμένει αναγκαία για την αντιμετώπιση των προβλημάτων που παρουσιάζονται κατά το διάστημα της ανάπτυξης αλλά θα

πρέπει να αναγνωρίζει την ικανότητα των μελών της ομάδας να λαμβάνουν από μόνοι τους αποφάσεις.

Απλότητα. Μια ομάδα ανάπτυξης λογισμικού, για να είναι ευέλικτη, θα πρέπει να λαμβάνει πάντα την πιο απλή απόφαση που μπορεί να εξυπηρετήσει τους στόχους της. Δεν πρέπει να προβλέπει μελλοντικά προβλήματα και να προσπαθεί να τα επιλύσει σήμερα. Η απλότητα είναι αναγκαία ώστε να είναι εύκολο να αλλάξει το σύστημα. Ποτέ δεν παράγεται περισσότερο από ό,τι είναι απαραίτητο.

Συνεργασία. Η συνεχής συνεργασία μεταξύ των μελών της ομάδας είναι απαραίτητη. «Οι ευέλικτες ομάδες δεν μπορούν να υπάρξουν μόνο με την περιστασιακή επικοινωνία. Χρειάζεται συνεχής πρόσβαση και εμπλοκή στις διαδικασίες της επιχείρησης για την οποία αναπτύσσεται το σύστημα» [26]. Επίσης, ο πελάτης πρέπει να συνεργάζεται στενά με την ομάδα ανάπτυξης, παρέχοντας συχνή ανατροφοδότηση για το υπό ανάπτυξη σύστημα.

Μικρές αυτό-οργανωμένες ομάδες. Οι ευέλικτες μέθοδοι ανάπτυξης λογισμικού λειτουργούν καλύτερα με μικρές ομάδες οι οποίες πρέπει να οργανώνονται από μόνες τους. Είναι συνηθισμένη η πρακτική να κοινοποιούνται οι ευθύνες στην ομάδα ως σύνολο, και έπειτα η ομάδα να καθορίζει τον καλύτερο τρόπο για την επίτευξή τους μέσω συζητήσεων και συχνής επικοινωνίας μεταξύ των μελών της για όλες τις παραμέτρους του έργου.

Υπάρχει μία μερίδα επιστημόνων, που αντιμετωπίζει τις ευέλικτες μεθόδους με σκεπτικισμό και πολλές φορές παρερμηνεύει τις βασικές αρχές και πρακτικές που εισάγουν. Η ανταπόκριση στις αλλαγές αντί ακολουθούμενου σχεδίου, το οποίο είναι μία από τις τέσσερις προτροπές των ευέλικτων μεθόδων, παρερμηνεύεται σαν έλλειψη σχεδιασμού στην ανάπτυξη και συντελεί στην αύξηση των κινδύνων. Αυτό όμως δεν ισχύει, καθότι ο σχεδιασμός για κάθε επαναληπτικό κύκλο, είναι πλήρης και απόλυτα ακριβής, επιτρέποντας την διασπορά των κινδύνων αλλά και την άμεση ανατροφοδότηση, που είναι ένας μηχανισμός μείωσης των κινδύνων.

5.2 Μέθοδοι

5.2.1 Ακραίος προγραμματισμός

Ο ακραίος προγραμματισμός είναι η πιο δημοφιλής από τις ευέλικτες μεθόδους και δημιουργήθηκε στα τέλη της δεκαετίας του '90 από τον Kent Beck^{1 2}. Είναι μία πειθαρχημένη προσέγγιση που υποστηρίζει την ταχύτερη ολοκλήρωση του έργου σε μικρούς επαναληπτικούς κύκλους, επιτρέποντας στους υπεύθυνους ανάπτυξης να ανταποκρίνονται στις με-

¹http://en.wikipedia.org/wiki/Extreme_Programming

²http://en.wikipedia.org/wiki/Kent_Beck

ταβαλλόμενες απαιτήσεις του πελάτη καθ' όλη τη διάρκεια του κύκλου παραγωγής. Βασίζεται σε τέσσερις αρχές και δώδεκα πρακτικές [29] οι οποίες βελτιώνουν και απλοποιούν την ανάπτυξη. Οι τέσσερις αρχές είναι:

- **Επικοινωνία (communication).** Η κοινή αντιμετώπιση των προβλημάτων απαιτεί κοινή κατανόηση μέσω διαπροσωπικής επικοινωνίας.
- **Απλότητα (simplicity).** Η ομάδα εκτελεί πάντα το πιο απλό σχέδιο που πιθανόν θα δουλέψει. Συνεχώς απλοποιεί και βελτιώνει τον κώδικα με αναδόμηση. Όταν ο κώδικας είναι απλός, είναι περισσότερο κατανοητός και καλύτερα τεκμηριωμένος, ενώ τα λάθη είναι πολύ λιγότερα.
- **Ανατροφοδότηση (feedback).** Η τακτική παράδοση λογισμικού στον πελάτη, σχεδόν ανά εβδομάδα, βοηθά στην συνειδητοποίηση προβλημάτων πριν την ολοκλήρωση του έργου.
- **Κουράγιο (courage).** Το κουράγιο και η καλή ψυχολογία είναι δύο σημαντικοί παράγοντες. Επιτυχημένες ομάδες ανάπτυξης λογισμικού συχνά ενεργούν στην κόψη του ξυραφιού, καθώς πρέπει να εργάζονται πολύ γρήγορα, δίχως όμως να χάνουν τον έλεγχο. Αυτό σημαίνει πως συχνά αποτυγχάνουν. Αν οι προγραμματιστές φοβούνται την αποτυχία, θα ενεργούν πολύ αργά. Σε μια αποτυχία η ομάδα πρέπει να βρίσκει το κουράγιο να συνεχίζει ή να ξεκινά από την αρχή.

Οι αξίες αυτές διαμορφώνουν ένα διαφορετικό πλαίσιο προσέγγισης της ανάπτυξης λογισμικού. Στο πλαίσιο αυτό τα στελέχη της ανάπτυξης επικοινωνούν μεταξύ τους και με τους πελάτες μέσω συνεχούς διαπροσωπικής επικοινωνίας, εξασφαλίζοντας την άμεση ανατροφοδότηση. Χρησιμοποιούν απλό σχεδιασμό διατηρώντας τον κώδικα απλό και βρίσκουν το κουράγιο να συνεχίσουν μετά από κάποια αποτυχία. Το πλαίσιο αυτό συμπληρώνεται με δώδεκα πρακτικές που αλληλεπιδρούν μεταξύ τους και πρέπει να εφαρμόζονται ενιαία σε όλες τις φάσεις της ανάπτυξης [32]. Αυτές είναι:

- Το παιχνίδι του σχεδιασμού (planning game)
- Προγραμματισμός σε ζεύγη (pair programming)
- Έλεγχοι πριν την κωδικοποίηση (test-first-design)
- Πρότυπα κωδικοποίησης (coding standards)
- Απλός σχεδιασμός (simple design)
- Ανακατασκευή κώδικα (refactoring)
- Διαρκείς ενσωματώσεις στον κώδικα (continuous integration)
- Συλλογική ιδιοκτησία του κώδικα (collective code ownership)
- Διαρκής παρουσία πελάτη (on-site customer)
- Μικρές εκδόσεις (small releases)

- Υποφερτός ρυθμός εργασίας (sustainable pace)
- Συνολική εικόνα του συστήματος (system metaphor)

Οι αξίες μαζί με τις πρακτικές τίθενται σε καθημερινή δοκιμασία στην διαδικασία ανάπτυξης του λογισμικού. Το έργο λογισμικού ξεκινά με μία φάση διερεύνησης, όπου οι πελάτες παραθέτουν στους προγραμματιστές και στον διευθυντή του έργου τις απαιτήσεις του συστήματος υπό μορφή γραπτών κειμένων – ιστοριών χρήστη (user stories). Κάθε μία από αυτές τις ιστορίες χρήστη περιγράφει μία απλή λειτουργία του συστήματος και διασπάται από την ομάδα σε τμήματα εργασίας, τις προγραμματιστικές διεργασίες (tasks). Για κάθε μία από αυτές τις διεργασίες εκτιμάται ο χρόνος υλοποίησης χρησιμοποιώντας σαν μέτρο μέτρησης τις ιδανικές προγραμματιστικές μέρες (ideal programming days). Αν κάποια ιστορία χρήστη είναι πολύπλοκη, τότε διασπάται σε μικρότερες ιστορίες. Στην συνέχεια οι ιστορίες χρήστη ταξινομούνται σύμφωνα με ποιιά προτεραιότητα έχουν στην ανάπτυξη και ανάλογα με τη βαρύτητα που προσδίδουν σε αυτές οι πελάτες. Από την αρχή της διερευνητικής φάσης ξεκινά το ‘παιχνίδι του σχεδιασμού’ σύμφωνα με το οποίο θα αποφασιστεί από την ομάδα το πλήθος των επί μέρους εκδόσεων (small releases) και των επί μέρους επαναλήψεων (iterations) κάθε έκδοσης.

Ονομάστηκε το παιχνίδι του σχεδιασμού, γιατί ο καθορισμός και η υλοποίηση τόσο των ιστοριών χρήστη, όσο και των εκδόσεων και επαναλήψεων, μοιάζει με παιχνίδι το οποίο διέπουν κανόνες που πρέπει να ακολουθούνται και το οποίο πρέπει να κερδηθεί. Πόσες και ποιες ιστορίες θα υλοποιηθούν σε μία επί μέρους έκδοση, καθώς και το πλήθος των επαναλήψεων της έκδοσης, καθορίζονται σε μία συνάντηση σχεδιασμού έκδοσης (release planning meeting), στην οποία συμμετέχουν όλα τα μέλη της ομάδας και οι πελάτες. Οι ιστορίες, που θα υλοποιηθούν, επιλέγονται από τους πελάτες οι οποίοι επιλέγουν και τους ελέγχους αποδοχής ή ορθότητας του συστήματος (acceptance tests). Οι προγραμματιστές εκτιμούν το χρόνο και το κόστος υλοποίησης της κάθε ιστορίας χρήστη. Σε μία επανάληψη μπορούν να αφαιρεθούν ή να προστεθούν ιστορίες ή και επιπλέον διεργασίες ανάλογα με τα χρονικά περιθώρια και τις δυνατότητες που παρέχονται.

Η καθοδηγούμενη από ελέγχους ανάπτυξη (Test Driven Development - TDD) αποτελεί το κέντρο βάρους της ανάπτυξης λογισμικού του ακραίου προγραμματισμού. Αυτή η προσέγγιση έγινε αποδεκτή ακόμη και από τους θιασώτες των παραδοσιακών μεθόδων και υλοποιείται από δύο σημαντικές πρακτικές που ήδη εφαρμόζονται και σαν μεμονωμένες πρακτικές στην βιομηχανία λογισμικού: του προγραμματισμού σε ζεύγη προγραμματιστών και τους ελέγχους πριν την κωδικοποίηση. Προγραμματισμός σε ζεύγη σημαίνει ότι η ομάδα χωρίζεται σε ζεύγη προγραμματιστών και το κάθε ζεύγος αναλαμβάνει την ολοκλήρωση μιας διεργασίας. Κάθε ζεύγος προγραμματιστών χρησιμοποιεί έναν υπολογιστή. Ο ένας προγραμματιστής ο οδηγός (driver) κωδικοποιεί και ο άλλος ο πλοηγός (navigator) ελέγχει τον κώδικα συνεχώς για την αποφυγή λαθών και προτείνει λύσεις για την

ανάπτυξη. Οι προγραμματιστές αλλάζουν συνεχώς ρόλους αλλά και συνεργάτες, όταν το απαιτούν κάποιες διεργασίες. Τα πλεονεκτήματα του προγραμματισμού σε ζεύγη είναι η ταχύτερη ανάπτυξη του κώδικα και ο περιορισμός των λαθών, που σημαίνει ποιοτικότερο κώδικα. Επιπλέον η συχνή αλλαγή συνεργάτη βοηθά στη μεταφορά της εμπειρίας και γνώσης σε όλα τα μέλη της ομάδας.

Οι έλεγχοι του κώδικα είναι συνεχείς, ποιοτικοί και αυτοματοποιημένοι, παρέχοντας μία δικλείδα ασφαλείας σε κατασκευαστές και πελάτες. Για να διατηρήσει η ομάδα ανάπτυξης τον κώδικα απλό και τεκμηριωμένο τον ανακατασκευάζει συχνά ακολουθώντας κάποια σταθερά πρότυπα κωδικοποίησης, που είναι κοινά για τους προγραμματιστές και την εταιρία. Το μεγαλύτερο πλεονέκτημα από τον απλό και κατανοητό κώδικα είναι ότι μπορεί εύκολα να ανταποκριθεί σε νέες απαιτήσεις και επεκτάσεις. Στις περιπτώσεις των αλλαγών και επεκτάσεων το κόστος παραμένει σχεδόν σταθερό καθ'όλη τη διάρκεια του κύκλου ανάπτυξης του συστήματος, σε αντίθεση με τις άλλες μεθόδους.

Με τις συνεχείς και καθημερινές ενσωματώσεις ή ενοποιήσεις του κώδικα το σύστημα διατηρείται ολοκληρωμένο, παρέχοντας ακόμη και από τα πρώτα στάδια της ανάπτυξης την δυνατότητα της παράδοσης ωφέλιμου λογισμικού σε μικρές εκδόσεις. Η συνεχής ενσωμάτωση του κώδικα βοηθά στο σχηματισμό της τρέχουσας εικόνας του συστήματος σε σχέση με το επιδιωκόμενο τελικό προϊόν και στη μείωση της ασυμβατότητας των διαφορετικών τμημάτων κώδικα των υποομάδων. Κατά την ενσωμάτωση του έτοιμου κώδικα εκτελούνται οι έλεγχοι ενσωμάτωσης για την εύρεση πιθανών σφαλμάτων σε επίπεδο συστήματος. Είναι πιο εύκολη η διόρθωση ενός σφάλματος που προέκυψε την προηγούμενη μέρα παρά η διόρθωση πολλαπλών σφαλμάτων προηγούμενων εβδομάδων.

Ο ακραίος προγραμματισμός δίνει ιδιαίτερη έμφαση στην ομαδική εργασία. Οι ομάδες ανάπτυξης πέρα από τους κατασκευαστές και τους διευθυντές των έργων συμπεριλαμβάνουν και τον πελάτη / χρήστη που έχει καθημερινή παρουσία στον χώρο παραγωγής. Στόχος της ομάδας είναι η ανάπτυξη ποιοτικού λογισμικού μέσα στα προβλεπόμενα χρονικά διαστήματα για την ικανοποίηση του πελάτη. Ο πελάτης πρέπει να είναι πλήρως καταρτισμένος και ενημερωμένος, ώστε μέσω της διαπροσωπικής επικοινωνίας να βοηθά στην υλοποίηση του συστήματος. Είναι αυτός που καθορίζει την προτεραιότητα εκτέλεσης των ιστοριών, βοηθώντας την ομάδα να προβλέπει με μεγαλύτερη ακρίβεια πόσο έργο μπορεί να παραχθεί σε δοθέν χρονικό διάστημα. Στα καθήκοντα του πελάτη συμπεριλαμβάνεται και ο καθορισμός των ελέγχων αποδοχής για να διασφαλιστεί η σωστή υλοποίηση των ιστοριών. Οι έλεγχοι αυτοί δημιουργούνται από τους πελάτες ή από την ομάδα των προγραμματιστών, ή από κάποιο ανεξάρτητο πρόσωπο.

Η συνολική εφαρμογή των πρακτικών του ακραίου προγραμματισμού σε μία εταιρία που εφαρμόζει παραδοσιακές μεθοδολογίες ίσως φανεί δύ-

σκολη και απαιτήσει κάποιο χρονικό διάστημα προσαρμογής. Όμως δεν θα πρέπει να εκληφθεί σαν την αναγκαστική ακολουθία μιας λίστας από απαράβατους και δύσκαμπτους κανόνες αλλά σαν μια νέα ενιαία προσέγγιση ανάπτυξης λογισμικού, που πρέπει να ενσωματωθεί με ευελιξία και φυσικότητα στις επικρατούσες συνθήκες και πιθανές ιδιαιτερότητες του εργασιακού μας περιβάλλοντος. Από τις μέχρι τώρα ερευνητικές μελέτες αλλά και από τις εμπειρίες των ανθρώπων της βιομηχανίας λογισμικού, οι οποίες εφαρμόζουν τον ακραίο προγραμματισμό, τα συμπεράσματα είναι εντυπωσιακά. Η φιλοσοφία της καθοδηγούμενης από ελέγχους ανάπτυξης έχει αναγνωριστεί πλέον από όλους σαν την καλύτερη προσέγγιση ανάπτυξης λογισμικού και εφαρμόζεται ανεξάρτητα από την μεθοδολογία που ακολουθείται. Η έμφαση που δίνει στον τρόπο υλοποίησης και αποτίμησης του εκτελέσιμου κώδικα, αποσκοπώντας πάντα στην ανάπτυξη ποιοτικότερου κώδικα και στην ικανοποίηση του πελάτη, κατέστησαν τον ακραίο προγραμματισμό την πιο επιτυχημένη μέθοδο.

5.2.2 Lean Ανάπτυξη Λογισμικού

H Lean ανάπτυξη λογισμικού (Lean Software Development) έχει τις ρίζες της στο σύστημα παραγωγής της Toyota το οποίο άρχισε να αναπτύσσεται από τη δεκαετία του '50 [33]. Έχει δανειστεί και προσαρμόσει τις αρχές και τις πρακτικές που διέπουν αυτό το σύστημα παραγωγής στον τομέα της ανάπτυξης λογισμικού. Η έννοια της Lean ανάπτυξης λογισμικού αν και προϋπήρχε από το 1993 οι Mary και Tom Poppendieck ήταν εκείνοι οι οποίοι την απογείωσαν δείχνοντας πως η λογική του Lean μπορούσε να εφαρμοστεί και στην ανάπτυξη λογισμικού παρέχοντας παρόμοια βελτίωση με αυτή που παρατηρήθηκε στον τομέα της παραγωγής προϊόντων.

To Lean Software Development προσβέυει 7 βασικές αρχές [34]:

- **Περιορισμός σπατάλης (Eliminate waste):** Η φιλοσοφία αυτή ορίζει πως πρέπει να εξαλείφονται τα περιττά στοιχεία τα οποία δεν αποδίδουν κάποια αξία στο προϊόν σε σχέση με τις ανάγκες του πελάτη. Στην ανάπτυξη λογισμικού παράδειγμα σπατάλης θεωρείται η συγγραφή κώδικα ή η πρόσθεση χαρακτηριστικών στο λογισμικό που δεν έχουν αξία για τον πελάτη και πιθανότατα δεν θα τα χρησιμοποιήσει πολύ.
- **Ενίσχυση γνώσης (Amplify learning):** Η σημασία αυτής της αρχής έγκειται στο ότι καθ' όλη τη διάρκεια παραγωγής του λογισμικού η γνώση θα πρέπει να ενισχύεται προκειμένου το τελικό αποτέλεσμα να έχει αξία για τον πελάτη. Έτσι απαιτείται μια σειρά επαναληπτικών φάσεων κατά τις οποίες ο πελάτης ανατροφοδοτεί την ομάδα που παράγει το λογισμικό, ο κώδικας τεστάρεται και επαναπροσαρμόζεται και η ομάδα αξιοποιεί τη γνώση που αποκτά από κάθε φάση.

- **Αναστολή δεύσμευσης (Decide as late as possible):** Η ανάπτυξη λογισμικού διακτέχεται από μια αβεβαιότητα, συνεπώς καθυστερώντας τις τελικές αποφάσεις που μπορούν να επιτευχθούν καλύτερα αποτελέσματα. Θα πρέπει οι τελικές αποφάσεις να βασίζονται σε γεγονότα και όχι σε αρχικές θεωρήσεις ή υποθέσεις. Οι ευέλικτες μέθοδοι επιτυγχάνουν αυτή τη λογική δίνοντας χρόνο στο πελάτη με την επαναληπτική τους διαδικασία να κατανοήσει και να αξιολογήσει ποιές είναι οι ουσιαστικές του ανάγκες. Άλλωστε κάθε πελάτης στην αρχή οραματίζεται το τελικό προϊόν με τρόπο που πιθανότατα εμπεριέχει χαρακτηριστικά τα οποία στην πράξη δεν τα χρειάζεται. Για να αποφευχθεί, απαιτείται αξιολόγηση των απαιτήσεων και οι τελικές δεσμεύσεις να γίνονται αργά, διαφορετικά το κόστος μπορεί να εκτοξευτεί.
- **Γρήγορη παράδοση (Deliver as fast as possible):** Στην εποχή της ταχύτατης ανάπτυξης της τεχνολογίας δεν είναι ο πιο ισχυρός εκείνος που κερδίζει αλλά ο πιο γρήγορος. όσο πιο γρήγορα βγει ένα προϊόν λογισμικού στην αγορά, τόσο πιο γρήγορα θα επιτευχθεί η ανατροφοδότηση από τους πελάτες και τόσο πιο αργά θα ληφθούν οι τελικές αποφάσεις. Όσο πιο σύντομες είναι οι επαναλήψεις παράδοσης λειτουργικού λογισμικού η ομάδα επικοινωνεί πιο αποδοτικά ενώ ταυτόχρονα ικανοποιούνται οι απαιτήσεις του πελάτη βάσει των τρεχόντων αναγκών του και όχι βάσει των υποθέσεων ή των οραμάτων του. Επίσης η ταχεία παράδοση του λογισμικού είναι κάτι που ευχαριστεί τους τελικούς πελάτες και η διάδρασή του με αυτό τους βοηθάει να κατανοήσουν καλύτερα ποιες είναι οι ανάγκες τους.
- **Εξασφάλιση ποιότητας (Build Quality In):** Είναι λογικό πως πάντα σε μια διαδικασία παραγωγής ενός προϊόντος θα προκύψουν προβλήματα. Σύμφωνα με το [35] υπάρχουν δυο προσεγγίσεις: είτε ελέγχεις το όποιο πρόβλημα αφότου έχει εμφανιστεί είτε κάνεις τους απαραίτητους ελέγχους προκειμένου να το αποφύγεις. Για να επιτευχθεί η ποιότητα όμως είναι σημαντικό να προληφθεί το όποιο ελάττωμα πριν προκύψει. Αυτή η διαδικασία μπορεί να γίνει μέσω unit tests (Test-Driven Development) και με τη συχνή διανομή μικρών τμημάτων λειτουργικού λογισμικού επαυξητικά. Η ποιότητα είναι σημαντική για τον πελάτη και θα πρέπει το παραληφθέν προϊόν να λύνει τα προβλήματα για τα οποία φτιάχτηκε αποδοτικά. Από την οπτική του πελάτη-χρήστη η ποιότητα ορίζεται από τη λειτουργικότητα, το τρόπο που διανέμεται, την τιμή καθώς και την ανταπόκριση σε όποιο πρόβλημα προκύπτει.
- **Σεβασμός στον άνθρωπο (Respect People):** Η σημασία ενωμένων και σκεπτόμενων ανθρώπων που δουλεύουν αρμονικά μεταξύ τους αποτελεί σημαντικό παράγοντα επιτυχίας στην ανάπτυξη λογισμικού. Για να επιτευχθεί αυτό απαιτείται σεβασμός προς τον καθένα και σωστή ηγεσία.

- **Βελτιστοποίηση του συνόλου (Optimize the Whole):** Το τελικό παραδοτέο πρέπει να διασπάται σε μικρότερα τμήματα. Σε κάθε ένα από αυτά θα πρέπει να υπάρχει σωστός έλεγχος προκειμένου να αποφευχθούν τα όποια ελαττώματα και κάθε ένα από αυτά θα πρέπει να λειτουργεί αρμονικά με τα υπόλοιπα. Η σημασία της βελτιστοποίησης του συνόλου είναι ότι όλοι οι συμμετέχοντες στην διαδικασία παραγωγής λογισμικού θα πρέπει να είναι σε θέση να δουν την τελική εικόνα και οι πελάτες να έχουν πλήρη ενημέρωση σχετικά με τις όποιες αλλαγές που μπορούν να τους επηρεάσουν.

Οι προαναφερθείσες αρχές τείνουν να είναι ιδιαίτερα συμβατές με τη φύση του web-based λογισμικού καθώς επικεντρώνουν στα στοιχεία που μπορούν να καταστήσουν μια εφαρμογή ή πλατφόρμα αποδοτική για τους τελικούς πελάτες σε γρήγορο χρονικό διάστημα επικεντρώνοντας στις πραγματικές τους ανάγκες. Όλο αυτό επιτυγχάνεται χωρίς σπατάλη χρόνου και πόρων διατηρώντας ταυτόχρονα την επιθυμητή ποιότητα. Οι παραπάνω αρχές ξεκαθαρίζουν πως [36]:

- **Η ικανοποίηση του πελάτη είναι η υψηλότερη προτεραιότητα:** Η ομάδα ανάπτυξης πρέπει να έχει πρακτικές για να καθορίσουν την προτεραιότητα του πελάτη και να ακούσουν τις απαντήσεις. Ο στόχος είναι η μεγιστοποίηση της ικανοποίησης των πελατών, όχι η τεχνική ποιότητα. Το να μην ανταποκρίνονται στις προσδοκίες των πελατών θεωρείται αποτυχία.
- **Πάντα να παρέχεται η καλύτερη αξία για τα χρήματα:** το λογισμικό θα πρέπει να βοηθήσει στην επίλυση των προβλημάτων των πελατών ή να τους παρέχουν μια νέα ευκαιρία σε λογικό κόστος. Η αξία και όχι η τελειότητα είναι ο στόχος. Η αξία είναι ένας συνδυασμός των χαρακτηριστικών των προϊόντων που ανταποκρίνονται στις ανάγκες των πελατών σε μια συγκεκριμένη χρονική στιγμή και για συγκεκριμένη τιμή. Για παράδειγμα, περιγράφοντας μια τυπική γερμανική κουλτούρα management, οι Womack και Jones ανέφεραν πως η ισχυρή τους τεχνογνωσία οδήγησε πολλές επιχειρήσεις στο να «προχωρήσουν με βελτιώσεις και πολυπλοκότητες που παρουσίαζαν μικρό ενδιαφέρον σε όλους πέραν από τους ίδιους τους εμπειρογνώμονες»(1996).
- **Η επιτυχία εξαρτάται από την ενεργό συμμετοχή των πελατών:** ενεργή προσπάθεια είναι μία συνεργατική προσπάθεια ένταξης, όχι για προσχηματικούς λόγους. Η συμμετοχή των πελατών είναι κάτι περισσότερο από “buy-in”: η ενεργός συμμετοχή είναι απαραίτητη για την προσαρμογή στην αλλαγή και στο να παίρνεις σε πραγματικό χρόνο αποφάσεις.
- **Κάθε lean ανάπτυξη είναι μια ομαδική προσπάθεια:** Οι αλληλοεξαρτώμενες, αλληλοελεγχόμενες ομάδες σχεδιαστών, σε αντίθεση με τα μεμωνομένα άτομα, χρειάζονται καθώς η ποικιλομορφία είναι

το κλειδί για την καινοτομία και της γρήγορου κύκλου ανάπτυξης. Ωστόσο, όσο πιο ποικιλόμορφη η ομάδα, τόσο πιο δύσκολο είναι να δημιουργηθεί ένα περιβάλλον συνεργασίας.

- **Τα πάντα είναι ευμετάβλητα:** Οι πρακτικές μαζικής ανάπτυξης βασίζονται αρχικά στην εξάλειψη των αλλαγών με τον καθορισμό των απαιτήσεων και στη συνέχεια με το να ελέγχονται οι αλλαγές που εντάσσονται. Η Lean ανάπτυξη διαβεβαιώνει πως οι απαιτούμενες αλλαγές θα είναι συνεχής και πως το να μάθει κανείς να προσαρμόζεται στις αλλαγές είναι καλύτερη στρατηγική από το να προσπαθεί να τις ελέγξει. Οι συνεχείς ερωτήσεις που υπάρχουν στην προσπάθεια ανάπτυξης Lean είναι: τι είδους αλλαγές μπορεί να συμβούν; Τι θα κάνουμε αν συμβούν; Πώς μπορούμε να οικοδομήσουμε την εφαρμογή ώστε να είναι πιο ανεκτική σε αυτού του είδους τις αλλαγές;
- **Ολοκλήρωσε, μην κατασκευάζεις:** Η «καλύτερα αγόρασε και μη χτίζεις» είναι εδώ και καιρό μια βιώσιμη στρατηγική για τις περισσότερες ομάδες ανάπτυξης εφαρμογών. Η χρήση ήδη υπαρχόντων εργαλείων μπορεί να διευκολύνει τη διαδικασία ανάπτυξης.
- **Μία λύση με βαθμό επίλυσης 80% σήμερα** αντί για μια λύση με βαθμό επίλυσης 100% αύριο: Οι αγορές κινούνται πάρα πολύ γρήγορα για να παρέχουν το 100% των λύσεων. Εμπειρικά, όλοι οι επαγγελματίες του κλάδου ανάπτυξης λογισμικού έχουν παρατηρήσει πως οι πελάτες είναι σχεδόν πάντα να εγκαταλείψουν την λειτουργικότητα για την έγκαιρη παράδοση ενός συστήματος που δουλεύει. Επιπλέον, έρευνες έχουν δείξει πως περισσότερο από το 45% της λειτουργικότητας των εφαρμογών λογισμικού δε χρησιμοποιείται ποτέ [36].
- **Οι ανάγκες καθορίζουν την τεχνολογία:** Επιλέξτε τους στόχους της LD και έπειτα την τεχνολογία που θα τους υποστηρίξει, όχι το αντίστροφο. Οι επιλογές της τεχνολογίας σήμερα είναι τόσο μεγάλες και αυτό καθιστά πιο εύκολο το να περνά κανείς περισσότερο χρόνο μεταβάλλοντας τεχνολογίες παρά δημιουργώντας επιχειρηματικές εφαρμογές.
- **Η αύξηση του προϊόντος σημαίνει αύξηση της λειτουργικότητας και όχι αύξηση του μεγέθους:** Κρίσιμο σημείο στην LD φιλοσοφία είναι διανομή λογισμικού που αντέχει τις αλλαγές χωρίς ιδιαίτερο κόστος και κόπο. Πρέπει να είναι κατανοητό ότι είναι σημαντικό να αξιολογούνται τα νέα χαρακτηριστικά που προτείνονται να προστεθούν και όχι απλά να προστίθενται.
- **Τελος η μέθοδος Lean θα πρέπει να εφαρμόζεται μόνο εάν ταιριάζει** στο εκάστοτε εγχείρημα. Δεν αποτελεί πανάκεια αλλά ένα πλαίσιο αρχών που εξυπηρετεί την ανάπτυξη λογισμικού.

5.2.3 Ευέλικτες Μέθοδοι - Καινοτόμες Προσεγγίσεις

TΗΝ τελευταία δεκαετία έχουν αναπτυχθεί αρκετές νέες μέθοδοι διοίκησης καινοτόμων ομάδων μιας και οι συνεχείς τεχνολογικές εξελίξεις έχουν οδηγήσει στη ραγδαία αύξηση των τεχνολογικών καινοτόμων επιχειρήσεων οι οποίες πρέπει να προσαρμόζονται στις συνεχείς αλλαγές του οικοσυστήματός τους. Για το λόγο αυτό έχουν αναπτυχθεί μοντέλα διοίκησης τα οποία κατά κύριο λόγο αναφέρονται σε ομάδες οι οποίες χρησιμοποιούν ευέλικτες μεθόδους ανάπτυξης λογισμικού. Παρακάτω παρουσιάζουμε δυο βασικές καινοτόμες προσεγγίσεις οι αρχές των οποίων μπορούν να εφαρμοστούν στην ανάπτυξη web-based λογισμικού που απευθύνεται σε αναρίθμητους πιθανούς τελικούς πελάτες-χρήστες και σαν στόχο έχουν την προσέλκυση και την ικανοποίηση αυτού χωρίς να απαιτείται σπατάλη χρόνου και χρήματος.

Lean startup μέθοδος: Η μέθοδος αυτή για την ανάπτυξη των επιχειρήσεων και των προϊόντων, δημοσιεύτηκε από τον Ries [38], ο οποίος ανέπτυξε την ιδέα αυτής της μεθοδολογίας από την εμπειρία του ως σύμβουλος και ιδρυτής startup επιχειρήσεων. Ο Ries ισχυρίζεται ότι οι startups μπορούν να συντομεύσουν τον κύκλο ανάπτυξης των προϊόντων ή υπηρεσιών τους, με την υιοθέτηση ενός συνδυασμού του επιχειρηματικού πειραματισμού, της εφαρμογής, της επαναληπτικής κυκλοφορίας προϊόντων και της μάθησης.

Οι startup επιχειρήσεις ξεκινούν με την αναζήτηση ενός επιχειρηματικού μοντέλου. Δοκιμάζουν, αναθεωρούν, θέτουν και απορρίπτουν υποθέσεις, συγκεντρώνουν συνεχώς πληροφόρηση από τους πελάτες και μέσω της διαδικασίας ταχέων επαναλήψεων ανασχεδιάζουν το προϊόν τους. Η στρατηγική αυτή μειώνει κυρίως τις πιθανότητες σπατάλης χρόνου και χρημάτων από την προώθηση προϊόντων που δεν θα είχαν τελικά απήχηση [39]. Η φιλοσοφία της τεχνικής Lean Startup επιδιώκει να εξαλείψει περιττές πρακτικές και να αυξήσει την αξία εκείνων των που δημιουργούνται κατά το στάδιο ανάπτυξης του προϊόντος, έτσι ώστε οι startups να μπορούν να έχουν μεγαλύτερη πιθανότητα για επιτυχία.

Οι τρεις βασικές αρχές του Lean Startup είναι οι ακόλουθες:

1. Δημιουργία στενής επαφής με δυνητικούς πελάτες για να δοκιμάσουν την ισχύ των υποθέσεων των επιχειρήσεων, λαμβάνοντας ανατροφοδότηση από όλα τα ενδιαφερόμενα μέρη (πελάτες, αγοραστές, συνεργάτες) σχετικά με όλα τα συμπεριλαμβανόμενα στο σχέδιό τους στοιχεία (χαρακτηριστικά του προϊόντος, τιμολόγηση, κανάλια διανομής).
2. Δημιουργία υποθέσεων οι οποίες ενσωματώνονται σε ένα πλαίσιο που αποκαλείται καμβάς επιχειρηματικού μοντέλου (business model canvas) και είναι το διάγραμμα του πώς η επιχείρηση θα δημιουργήσει αξία για αυτήν και τους πελάτες, αντί χρονοβόρου σχεδιασμού

και έρευνας. Αποτελεί μία στρατηγική διαχείρισης για την ανάπτυξη νέων ή την επιβεβαίωση υπαρχόντων επιχειρηματικών μοντέλων περιλαμβάνοντας στοιχεία που περιγράφουν την πρόταση της επιχείρησης, τις υποδομές, τους πελάτες, καθώς και τα οικονομικά στοιχεία. Βοηθά στην κατανόηση, επικοινωνία, δημιουργικότητα και ανάλυση της επιχείρησης και στην ευθυγράμμιση των δραστηριοτήτων της.

3. Περιορισμός του χρόνου και πόρων δημιουργώντας προϊόντα σταδιακά, βασισμένα στην ήδη υπάρχουσα γνώση για τις ανάγκες του καταναλωτή και ακολουθώντας τη διαδικασία δημιουργίας του ελάχιστου βιώσιμου προϊόντος (MVP) βάσει δοκιμών. Η έκδοση ενός νέου προϊόντος, το οποίο επιτρέπει σε μια ομάδα να συλλέξει τη μέγιστη απαιτούμενη πληροφόρηση για τους πελάτες με την ελάχιστη προσπάθεια, έχει στόχο να δοκιμάσει τις θεμελιώδεις επιχειρηματικές υποθέσεις και να βοηθήσει τους επιχειρηματίες να ξεκινήσουν τη διαδικασία της μάθησης όσο το δυνατόν γρηγορότερα. Κατά την εφαρμογή αυτής της πρακτικής (μοντέλου MVP), της δοκιμής δηλαδή ενός ευρέως φάσματος προτύπων και της ανάλυσης των αντιδράσεων των χρηστών, πριν την τελική επιλογή των καλύτερων προϊόντων, πρέπει κάθε στοιχείο σχεδιασμού να επιλέγεται επειδή υπερίσχυσε έναντι άλλων λιγότερο ικανοποιητικών εναλλακτικών, κατά τα πρώτα στάδια των δοκιμών. Εκμεταλλευόμενη η επιχείρηση το πλεονέκτημα δοκιμής και σε περίπτωση αποτυχίας, εγκατάλειψης του προϊόντος στα αρχικά στάδια, εξοικονομεί πολύτιμους πόρους σε έρευνα και ανάπτυξη, μάρκετινγκ καθώς και στο κόστος παραγωγής.

Βασικό μέρος της μεθοδολογίας Lean Startup αποτελεί η διαδικασία μετατροπής των ιδεών σε προϊόντα, μετρώντας τους πελάτες βάσει της αντίδρασης και της συμπεριφοράς τους έναντι των προϊόντων και στη συνέχεια στη λήψη απόφασης για το εάν θα επιμείνει ή εάν θα αναστραφεί η ιδέα μέσω της εφαρμογής μιας διαδικασίας διόρθωσης, σχεδιασμένης για να δοκιμάσει μία νέα θεμελιώδη υπόθεση σχετικά με το προϊόν, τη στρατηγική και την ανάπτυξη. Επίσης η μεθοδολογία του Lean Startup εστιάζει στην ταχύτητα ως ένα από τα πιο κρίσιμα συστατικά για την ανάπτυξη του προϊόντος. Η αποτελεσματικότητα μιας ομάδας καθορίζεται από την ικανότητά της να συλλαμβάνει ιδέες, να κατασκευάζει γρήγορα ένα αρχικό δείγμα της ιδέας, να μετρά την αποτελεσματικότητά του στην αγορά και να μαθαίνει από αυτό το πείραμα. Αυτή η ταχεία επανάληψη επιτρέπει στην ομάδα να καθορίσει την πιο εφικτή διαδρομή για την κατάκτηση της αγοράς [39].

Οι ιδρυτές που υιοθετούν αυτή τη μέθοδο εστιάζουν στην εκτέλεση του έργου τους μόνο κατόπιν πολλών πειραματισμών και δοκιμών και μόνο όταν η πληροφόρησή τους από τους πελάτες τους οδηγεί προς αυτή την κατεύθυνση. Αυτή η νέα μέθοδος αναγνωρίζει ότι η αναζήτηση για ένα επιχειρηματικό μοντέλο που είναι το πρώτο θέμα που αντιμετωπίζει μια startup επιχείρηση, είναι εντελώς διαφορετική από την εκτέλεση του

μοντέλου (πρακτική που οι εδραιωμένες επιχειρήσεις εφαρμόζουν).

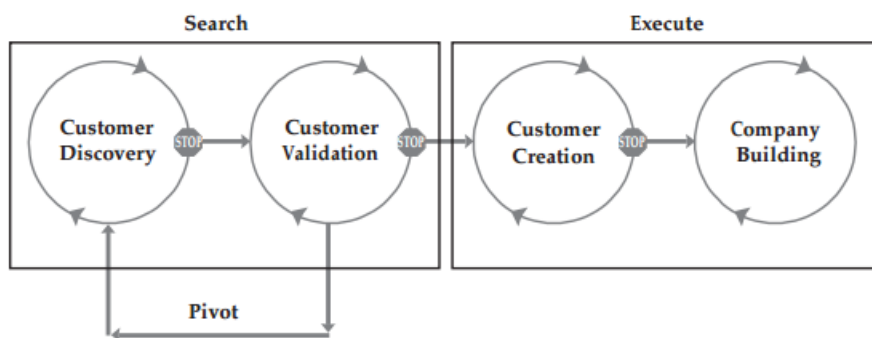
Customer Development: Ο S. Blank αναπτύξε την Customer Development μεθόδο [40], η οποία επικεντρώνεται στη δημιουργία μίας ομάδας έργου – της Customer Development Team- η οποία απαρτίζεται από μηχανικούς λογισμικού και στελέχη του εμπορικού τμήματος της ομάδας με στόχο την πληρέστερη κατανόηση των απαιτήσεων των πελατών τους και την προσαρμογή στις συνεχείς μεταβολές. Η συγκεκριμένη μέθοδος αποτελείται από τρία επαναλαμβανόμενα βήματα, τα οποία έχουν ως στόχο την πλήρη αποσαφήνιση των απαιτήσεων των αναγκών των πελατών και τον έλεγχο ότι το υπό ανάπτυξη σύστημα ικανοποιεί τις συγκεκριμένες απαιτήσεις. Τρία συγκεκριμένα βήματα (Customer Discovery, Customer Validation, Customer Creation) επαναλαμβάνονται μέχρι να βρεθεί πραγματική λύση για τις απαιτήσεις των πελατών και εν τέλη να πραγματοποιηθεί το τελευταίο βήμα που είναι η δημιουργία της επιχείρησης (Company Building) [Σχήμα 5.1].

- **Customer Discovery:** Η φάση του Customer Discovery μπορεί να οριστεί ως η φάση κατά την οποία η εταιρεία εντοπίζει ένα πρόβλημα, το μετατρέπει σε μια ιδέα και έπειτα δημιουργεί κάποια πειράματα έτσι ώστε να τεστάρει την ιδέα αυτή. Για να μπορέσει να συμβεί αυτό, οι εταίροι αφήνουν τα γραφεία τους και έρχονται σε αλληλεπίδραση με τους εν δυνάμει πελάτες έτσι ώστε να διεξάγουν τα πειράματά τους, να λάβουν ανατροφοδότηση και να προσαρμόσουν το επιχειρηματικό τους μοντέλο αναλόγως. Η διαδικασία της διάδρασης με τους πελάτες είναι η πιο σημαντική του Customer Development καθώς έτσι μπορεί κανείς να μάθει σε βάθος τα προβλήματά τους και την άποψή τους σχετικά με το ποια χαρακτηριστικά των προϊόντων θα μπορούσαν να τα επιλύσουν. Κατά τη φάση αυτή προτείνεται από την εταιρεία μια πιθανή πρόταση για την επίλυση των προβλημάτων των πελατών και αξιολογείται η αντίδρασή τους. Στη δημιουργία μιας startup, οι ιδρυτές καθορίζουν το όραμά τους και χρησιμοποιούν το Customer Discovery για την εύρεση πελατών και αγοράς που να μπορεί να υποστηρίξει το όραμά τους. Επαναλήψεις και αποτυχίες των σταδίων μπορεί να συμβούν στη φάση του customer discovery. Είναι συνήθη μέρη της διαδικασίας. Οι επαναλήψεις έχουν ρόλο ζωτικής σημασίας καθώς αποτελούν τη λύση στο πρόβλημα της παρεξήγησης ή των λανθασμένων συμπερασμάτων και δεν αποτελούν αποτυχία, αντιθέτως εγγυώνται πιο ακριβή αποτελέσματα.

Customer Validation: Σε αυτή τη φάση αποδεικνύεται πως η εταιρεία που δοκιμάστηκε κατά τη φάση του customer discovery έχει ένα επαναλαμβανόμενο, κλιμακωτό επιχειρηματικό μοντέλο που αντιπροσωπεύει τους απαραίτητους για τη δημιουργία μιας επιτυχημένης εταιρείας, πελάτες. Κατά την φάση του validation, η εταιρεία διεξάγει ακόμα μία σειρά από τεστ στους πελάτες που είναι πιο απαιτητικά. Στις web/mobile εφαρμογές, το στάδιο του customer validation ζητάει

την ανάπτυξη μιας υψηλής αξιοπιστίας εκδοχή του μικρής διάρκειας ζωής προϊόντος για να δοκιμάσει τα χαρακτηριστικά-κλειδιά μπροστά στους πελάτες. Το στάδιο του customer validation αποδεικνύει την ύπαρξη πελατών, επιβεβαιώνει ότι οι πελάτες αποδέχονται το MVP προϊόν και επικυρώνει σοβαρές και μετρήσιμες συναλλαγές με τους πελάτες μέσω δοκιμαστικών πωλήσεων. Όταν συνδυάζεται με την agile μηχανική, μειώνει την ποσότητα του χαμένου κώδικα, χαρακτηριστικών και hardware. Η agile ανάπτυξη δημιουργεί το προϊόν σε μικρά κομμάτια, πράγμα που επιτρέπει στην εταιρία να δοκιμάσει και να μετρήσει τις αντιδράσεις των πελατών για το κάθε κομμάτι που δημιουργείται. Έτσι, σε μικρό χρονικό διάστημα μπορεί κανείς να καταλάβει τις πραγματικές ανάγκες των πελατών σε σχέση με το προϊόν.

Customer Creation: Η δημιουργία πελατών στηρίζεται στην αρχική επιτυχία της εταιρίας στις πωλήσεις. Σε αυτό το σημείο η εταιρία ξοδεύει μεγάλα ποσά για την παραγωγή του προϊόντος της και για να μπορέσει να δικτυωθεί στην αγορά. Η δημιουργία πελατών ακολουθεί έπειτα από το βήμα του customer validation και μπορεί να διαφέρει ανάλογα με τον τύπο της εταιρίας.



Σχήμα 5.1: Διαδικασία Ανάπτυξης Πελατών - Πηγή [2]

5.3 Ανάγκη Ευελιξίας στην Ανάπτυξη Web 2.0 Λογισμικού

Οι Web 2.0 εφαρμογές στις μέρες μας καταλαμβάνουν μεγάλο κομμάτι της καθημερινότητας πολλών ανθρώπων οι οποίοι δεν θα μπορούσαν να φανταστούν ούτε μια μέρα χωρίς την ύπαρξη αυτών. Ταυτόχρονα αποτελεί και κομμάτι ενδιαφέροντος και ενασχόλησης πολλών εν δυνάμει νεοφυών επιχειρήσεων αλλά και μεμονωμένων ανθρώπων. Η δυναμικότης τόσο οικονομικά όσο και κοινωνικά είναι μεγάλη και το να καταφέ-

ρει να αναπτύξει κάποιος τέτοιο λογισμικό ποιοτικά και αποδοτικά είναι δύσκολη πρόκληση [41].

Ο όρος Web 2.0 είναι άμεσα συνδεδεμένος με τον Tim O'Reilly ο οποίος αναφέρει πως οι web-based εφαρμογές προσφέρουν με τη λειτουργικότητά τους πλούσια εμπειρία χρήστη (UX). Αναφέρει πως παρέχουν διανομή πληροφορίας, εμπεριέχουν συνδρομή του τελικού χρήστη και πως το περιεχόμενο τους βασίζεται στο χρήστη πράγματα τα οποία ενισχύουν την αξία του web-based λογισμικού [42].

Στα χαρακτηριστικά που προσδιορίζουν το web 2.0, αρχικά, είναι ότι δίνεται τη δυνατότητα να γίνεται έλεγχος και ενημέρωση της όποιας εφαρμογής ή υπηρεσίας άμεσα μέσα από το διαδίκτυο γεγονός που οδηγεί στο να γίνονται διορθώσεις χωρίς να χρειάζεται επαναδιανομή. Για παράδειγμα το Flickr ανανεώνει τις εκδόσεις των υπηρεσιών του κάθε μισή ώρα. Αυτό σημαίνει πως οι web 2.0 εφαρμογές και υπηρεσίες είναι μια αέναη δοκιμαστική έκδοση για μεγάλο χρονικό διάστημα [43]. Επίσης το web 2.0 έχει δώσει κυρίαρχη θέση στον τελικό χρήστη επιτρέποντάς του να συμβάλλει τόσο στο περιεχόμενο του διαδικτύου όσο και στο να δημιουργεί αξία στο εκάστοτε προϊόν ενώ ταυτόχρονα μπορεί να επηρεάσει και άλλους χρήστες. Αυτό συνεπάγεται πως χαρακτηρίζεται από χρηστοκεντρικότητα άρα η αξία και η επιτυχία κάθε προϊόντος, εφαρμογής ή υπηρεσίας ορίζεται βάσει της ικανοποίησης των αναγκών του χρήστη και της ποιότητας εμπειρίας που το παρέχει.

Από τα παραπάνω διαφαίνεται πως το όποιο web 2.0 προϊόν απαιτεί προσαρμοστικότητα, γρήγορη απόκριση στις αλλαγές με κύριο γνώμονα τον τελικό χρήστη ο οποίος μπορεί να επηρεάσει σημαντικά την πορεία του. Συνεπώς όλα αυτά τα χαρακτηριστικά καταδεικνύουν την απαίτηση για υιοθέτηση κάποιου μοντέλου διαχείρισης της διαδικασίας παραγωγής των web 2.0 εφαρμογών και υπηρεσιών. Οι παραδοσιακές μέθοδοι αδυνατούν να συμβαδίσουν με τις συνεχείς αλλαγές απαιτήσεων και τη συνεχή συμβολή του χρήστη. Αντίθετα οι ευέλικτες μέθοδοι και οι αρχές που πρεσβεύουν είναι πιο συμβατές καθώς βασίζονται σε μια συχνή, επαναληπτική και επαυξητική διανομή λογισμικού, ορίζουν συχνή αλληλεπίδραση και ανατροφοδότηση από τον τελικό χρήστη και τονίζουν τη σημασία αξιοποίησης της ήδη υπάρχουσας γνώσης (ανταγωνισμός, διαθέσιμα εργαλεία) και την απόκτηση νέας. Άρα μπορούν να εγγυηθούν, εάν αξιοποιηθούν και προσαρμοστούν κατάλληλα οι αρχές τους στο εκάστοτε προϊόν, ποιότητα, ανταγωνιστικότητα, γρήγορη είσοδο στην αγορά με χαμηλό κόστος και περιορισμό σπατάλης χρόνου και πόρων.

5.4 Σύγκριση Ευέλικτων και Παραδοσιακών Μεθόδων

Πολλές φορές παρατηρείται ότι οι βαριές παραδοσιακές μέθοδοι δεν ανταποκρίνονται στις απαιτήσεις του ηλεκτρονικού επιχειρείν και πως δύσκολα συμβαδίζουν με το ρυθμό εξέλιξης ενός έργου. Οι ευέλικτες μέθοδοι υπερπηδούν τα παραπάνω εμπόδια, καθώς η πλειοψηφία τους μειώνει τις απαιτήσεις οργάνωσης (managerial requirements) και τεκμηρίωσης (documentation).

Τα σημαντικότερα μειονεκτήματα των παραδοσιακών μεθόδων έναντι των ευέλικτων είναι η γραμμικότητα, η έλλειψη ευελιξίας στην αλλαγή απαιτήσεων των πελατών και οι αυστηρές, δύσκαμπτες διαδικασίες ανεξάρτητα από το μέγεθος του έργου. Τα μειονεκτήματα αυτά αντιμετωπίστηκαν σε πρώτη φάση με την εισαγωγή του Extreme Programming από τον Kent Beck, που ήταν η πρώτη Agile μεθοδολογία που παρουσιάστηκε. Οι Agile μέθοδοι αντιμετωπίζουν τις ασταθείς και συχνά μεταβαλλόμενες απαιτήσεις εστιάζοντας στη συνεργασία των σχεδιαστών με τους πελάτες και στη γρήγορη παράδοση προϊόντος. Στον πίνακα 5.1 παρουσιάζονται οι κύριες διαφορές των δύο ομάδων μεθόδων ως προς βασικά χαρακτηριστικά των διαδικασίας ανάπτυξης λογισμικού.

Χαρακτηριστικό	Ευέλικτες Μέθοδοι	Παραδοσιακές Μέθοδοι
Προσέγγιση	Προσαρμοζόμενη	Προβλεφθείσα
Μέτρο Επιτυχίας	Εμπορική αξία	Προσήλωση στο σχέδιο
Μέγεθος Έργου	Μικρό	Μεγάλο
Τρόπος Διαχείρισης	Αποκεντρωμένος	Αυτοκρατορικός
Ευκολία Αλλαγών	Μεγάλη	Μικρή
Φιλοσοφία	Ηγεσία και συνεργασία	Διαταγές και έλεγχος
Τεκμηρίωση	Χαμηλή	Υψηλή
Έμφαση	Στον άνθρωπο	Στη διαδικασία
Κύκλοι	Πολλοί	Περιορισμένοι
Τομέας	Απρόβλεπτος / Εξερεύνηση	Προβλέψιμος
Σχεδιασμός εξ' αρχής	Ελάχιστος	Μέγιστος
Απόσβεση Επένδυσης	Νωρίς κατά την ανάπτυξη	Στο τέλος της ανάπτυξης
Ομάδα	Μικρή και δημιουργική	Μεγάλη

Πίνακας 5.1: Διαφορές Παραδοσιακών και Ευέλικτων Μεθόδων

Και οι δύο μεθοδολογίες χαρακτηρίζονται από δυνάμεις και αδυναμίες. Οι σχεδιαστές συχνά επιλέγουν να ακολουθήσουν τη μία από τις δύο θεωρήσεις είτε συνδυασμό αυτών. Οι παράγοντες που επηρεάζουν συνήθως την απόφαση των σχεδιαστών είναι το μέγεθος του έργου, οι άνθρωποι και ο κίνδυνος.

Μέγεθος Έργου. Τα βασικά στοιχεία που συνθέτουν το μέγεθος ενός έργου είναι ο προϋπολογισμός, η διάρκεια και η οργάνωση της ομάδας σχεδίασης/υλοποίησης. Όσο μεγαλύτερες είναι οι απαιτήσεις προϋπολογισμού ή ατόμων ομάδας, τόσο μεγαλύτερο είναι το έργο. Στην περίπτωση μεγάλων έργων πιο συμφέρουσα κρίνεται η χρήση παραδοσιακών μεθόδων, ώστε να επιτυγχάνεται καλύτερος συντονισμός μεταξύ των ατόμων. Στην περίπτωση πολύ μεγάλου αριθμού ατόμων, ειδικά, ενέχει ο κίνδυνος αύξησης του φόρτου επικοινωνίας μεταξύ των ομάδων σε τέτοιο βαθμό, που τελικά να επηρεάζεται η αποδοτικότητα του κάθε ατόμου προσωπικά. Επομένως, μία άποψη είναι ότι σε ομάδες μεγαλύτερες των 40 ατόμων θα πρέπει να προτιμούνται οι παραδοσιακές μέθοδοι έναντι των ευέλικτων. Μία άλλη άποψη, ωστόσο, θεωρεί χάσιμο χρόνο την αυστηρή οργάνωση και επικοινωνία μεταξύ των ομάδων, καθώς αυτή παράγει ογκώδη τεκμηρίωση, αναφορές, αναλύσεις κλπ.

Χαρακτηριστικά Έργου	Ευέλικτες Μέθοδοι	Παραδοσιακές Μέθοδοι
Πρωταρχικός Στόχος	Γρήγορη αξία	Υψηλή εξασφάλιση
Απαιτήσεις	Γρήγορες και συνεχείς αλλαγές, άγνοια για το πότε αυτές θα προκύψουν	Σταθερές και γνωστές σε μεγάλο βαθμό
Μέγεθος	Μικρές ομάδες και έργα	Μεγάλες ομάδες και έργα
Αρχιτεκτονική	Σχεδιασμός με βάση τρέχουσες απαιτήσεις	Σχεδιασμός με βάση τρέχουσες και μελλοντικές απαιτήσεις
Σχεδίαση και Έλεγχος	Εσωτερικός σχεδιασμός και ποιοτικός έλεγχος	Τεκμηριωμένος σχεδιασμός, ποσοτικός έλεγχος
Σχεδιαστές	Ευέλικτοι, γνώστες, συνεργάσιμοι	Ακολουθούντες το σχεδιασμό, με ικανές δεξιότητες και πρόσβαση σε εξωτερική γνώση
Επαναπροσδιορισμός	Πολύ μικρό έως μέτριο κόστος	Μέτριο έως πολύ μεγάλο κόστος
Κίνδυνοι	Άγνωστοι με μεγάλη επίδραση	Καλά κατανοητοί με μικρή επίδραση

Πίνακας 5.2: Διαφορές Παραδοσιακών και Ευέλικτων Μεθόδων

Άνθρωποι. Το κλειδί για την επιτυχημένη χρήση των ευέλικτων μεθοδολογιών είναι οι άνθρωποι που συνθέτουν την ομάδα σχεδίασης / υλοποίησης να έχουν αυξημένες δεξιότητες και εμπειρία. Αυτό επιτρέπει τη γρήγορη και επαρκή ανατροφοδότηση μεταξύ τους και την αντιμετώπιση εμπλοκών στη σχεδίαση κατά τη φάση της υλοποίησης. Ένας άλλος σημαντικός παράγοντας είναι η προσαρμοστικότητα στις μεταβαλλόμενες απαιτήσεις των πελατών. Οι πελάτες έχουν τη δυνατότητα να ελέγχουν την πρόοδο του έργου, αλλά και να αλλάζουν την πορεία του έργου κατά τη διάρκεια κάθε κύκλου. Επομένως, όσον αφορά τον παράγοντα «άνθρωπο», η επιλογή ομάδας μεθοδολογιών έχει να κάνει με τη φιλοσοφία της επιχείρησης. Εάν αυτή περιλαμβάνει την επιδίωξη υψηλής δέσμευσης πελατών (customer commitment) και τη γρήγορη ευελιξία και προσαρμογή στις απαιτήσεις τους, τότε οι ευέλικτες μέθοδοι ενδείκνυνται.

Κίνδυνος. Οι πιο σημαντικοί παράγοντες κινδύνου στην ανάπτυξη ενός έργου είναι η κρισιμότητα του έργου και η ανταπόκριση στις αλλαγές. Οι

ευέλικτες μέθοδοι χρησιμοποιούνται σε εφαρμογές που μπορούν να αναπτυχθούν γρήγορα και δεν απαιτούν εκτεταμένη διασφάλιση ποιότητας (quality assurance). Τα έργα που θεωρούνται κρίσιμα (critical), αυξημένης εμπιστοσύνης (reliable) και αυξημένης ασφάλειας (safe) καλό είναι να υλοποιούνται με χρήση παραδοσιακών μεθόδων. Η απαίτηση για προσαρμοστικότητα στις αλλαγές, από την άλλη, μπορεί να καλυφθεί με τη χρήση ευέλικτων μεθόδων.

Λαμβάνοντας υπόψη τις παραπάνω παρατηρήσεις, μπορεί κανείς να καταλήξει σε όσα περιγράφονται στον πίνακα 5.2, ο οποίος περιέχει τους όρους που θα πρέπει να ισχύουν ώστε να είναι περισσότερο πιθανή η επιτυχία των ευέλικτων και των παραδοσιακών μεθόδων, αντίστοιχα.

ΚΕΦΑΛΑΙΟ

6

ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ ΕΥΕΛΙΚΤΩΝ ΜΕΘΟΔΩΝ

6.1 Εισαγωγή

Εχοντας μελετήσει τις αρχές των ευέλικτων μεθόδων και εντοπίζοντας τη συμβατότητα τους με τη φύση των Web 2.0 εφαρμογών έγινε κατανοητό πως η αξιοποίηση των αρχών που πρεσβεύουν μπορεί να βοηθήσει ώστε η ανάπτυξη τέτοιων εφαρμογών να γίνεται γρήγορα και αποδοτικά. Οι μέθοδοι αυτοί σε σύγκριση με τις παραδοσιακές υιοθετούν μια πιο πελατοκεντρική προσέγγιση σε σύγκριση με τις παραδοσιακές μεθόδους. Έρευνες που έχουν γίνει έχουν δείξει πως σχεδόν το 75% νεοφυών τεχνολογικών επιχειρήσεων αποτυγχάνουν [47]. Ένας βασικός λόγος αυτής της αποτυχίας όπως προκύπτει από σχετικές έρευνες είναι το γεγονός ότι πολλοί από τους εμπνευστές τους καταναλώνουν μεγάλο μέρος τις ενέργειας και των πόρων τους στην ανάπτυξη του προϊόντος και τη διανομή του χωρίς προηγουμένως να έχουν κάνει την απαραίτητη έρευνα αγοράς προκειμένου να κατανοήσουν τον τελικό τους πελάτη και να αξιολογήσουν την αξία του προϊόντος τους πριν το ετοιμάσουν. Σχεδόν το 49% των δημιουργών εφαρμογών γενικότερα σχεδίαζαν και υλοποιούσαν, μέχρι πρότινος, το προϊόν τους έχοντας σαν κύριο γνώμονα τις δικές τους επιθυμίες και απόψεις [49] χωρίς να λαμβάνουν υπόψη το τελικό πιθανό χρήστη. Από αυτό μπορούμε να καταλάβουμε γιατί η πελατοκεντρικότητα δεν είναι απλά μια επιλογή, είναι αναγκαιότητα [31] όσο δύσκολο και αν είναι να εφαρμοστεί.

Όπως είδαμε και στο προηγούμενο κεφάλαιο η Lean μέθοδος έχει σαν βασική αρχή τη μείωση της σπατάλης στον μεγαλύτερο δυνατό βαθμό. Σπατάλη στην ανάπτυξη μιας εφαρμογής μπορεί να είναι ο επιπλέον κώδικας που υλοποιεί χαρακτηριστικά που κανείς δεν θα χρησιμοποιήσει ή δεν θα εκτιμήσει. Ένα ποσοστό 64% των λειτουργιών των εφαρμογών δεν χρησιμοποιείται από τους χρήστες ¹ το οποίο οφείλεται στο ότι οι απαιτήσεις όταν ορίζονται αποκλειστικά σε αρχικά στάδια δεν είναι οι πραγματικές ανάγκες των χρηστών. Άλλωστε είναι λογικό οι πελάτες-χρήστες σε πλαίσια διαπραγματεύσεως να απαιτούν πολλά στην αρχή γιατί θεωρούν ότι πολλά θα αφαιρεθούν, είτε διότι οι ίδιοι δεν έχουν αξιολογήσει σωστά τις ανάγκες τους. Αυτό καθιστά λοιπόν απαραίτητη τη συνεχή και επαναλαμβανόμενη συμμετοχή του τελικού πελάτη-χρήστη καθ' όλη τη διάρκεια ανάπτυξης μιας web 2.0 εφαρμογής ώστε να γίνουν ξεκάθαρες οι απαιτήσεις του και να εξαλειφθούν οι όποιες υπερβολές. Αυτό είναι ένας βασικός άξονας κίνησης προκειμένου να επιτευχθεί και ο κύριος στόχος των Lean-Agile μεθόδων που δεν είναι άλλος από την ικανοποίηση του τελικού πελάτη.

Επιπροσθέτως σπατάλη θεωρείται ο μη ποιοτικός κώδικας που δεν αναπροσαρμόζεται, δεν τεστάρεται και δεν είναι εύελικτος στις αλλαγές με αποτέλεσμα να μειώνεται η ποιότητα του τελικού προϊόντος, και φυσικά η μη αξιοποίηση των διαθέσιμων εργαλείων και τεχνολογιών. Άλλωστε οι εύελικτες μέθοδοι ξεκαθαρίζουν τη σημασία της υπάρχουσας γνώσης και σε επίπεδο εργαλείων και τεχνολογιών και την παραγωγή μέσω αυτών λογισμικού που είναι να είναι ευμετάβλητο και ευπροσάρμοστο στην όποια αλλαγή των απαιτήσεων. Εξού και απαιτούν καθυστέρηση της τελικής δέσμευσης και έγκαιρη, συχνή και επαναληπτική διανομή λειτουργικού λογισμικού προκειμένου οι αλλαγές να γίνονται όσο το δυνατόν νωρίτερα σύμφωνα πάντα με τις πραγματικές ανάγκες των χρηστών. Μέσω αυτής της συνεχούς αλληλεπίδρασης επιτυγχάνεται, πέραν των παραπάνω, και η δημιουργία σχέσης εμπιστοσύνης με τον χρήστη ενώ ταυτόχρονα εξασφαλίζεται και η ανάπτυξη κοινού που θα συνεχίσει να χρησιμοποιεί την εφαρμογή.

Στο παρόν κεφάλαιο αξιοποιήθηκαν όλα οι προαναφερθείσες αρχές και οι εμπειρίες που αποκομίσαμε παρουσιάζονται παρακάτω. Έγινε αρχικά μια αναζήτηση πιθανών πελατών όπως ορίζεται απο τη μέθοδο ανάπτυξης πελατών (Customer Development). Επιχειρήθηκε αποβολή της όποιας σπατάλης λόγω μη κατανόησης του πιθανού χρήστη το οποίο όπως μάθαμε και απο την προηγούμενη μας εφαρμογή μπορεί να κοστίσει πολύ. Υπήρξε επαναληπτική διάδραση με εν 'δυνάμει χρήστες μέσω γρήγορης διανομή λειτουργικής έκδοσης της εφαρμογής (2EAT) στοχεύοντας στην τόσο στην αξιολόγηση αυτής και των επαναπροσδιορισμό των απαιτήσεων όσο και στην δημιουργία πιθανόν πελατών-χρηστών. Τέλος εκμεταλλευτήκαμε υπάρχουσες τεχνολογίες προκειμένου να υπάρξει έγκαιρη διανομή και κατ' επέκταση ανατροφοδότηση.

¹<http://www.softwareresults.us/2011/07/where-do-unused-features-come-from.html>

6.2 Εμπειρική Ανάλυση - Μεθοδολογία και Στόχοι

6.2.1 Στόχοι Έρευνας

Το κεφάλαιο αυτό σαν στόχο έχει να παρουσιάσει την έρευνα που πραγματοποιήθηκε στα πλαίσια της εφαρμογής ευέλικτων μεθόδων για την ανάπτυξη Web 2.0 εφαρμογών που εξυπηρετούν την ηλεκτρονική παραγγελία φαγητού. Για την πραγμάτωση μιας πελατοκεντρικής προσέγγισης ήταν απαραίτητη η εξερεύνηση της στάσης και της προδιάθεσης των πιθανών χρηστών απέναντι στην ηλεκτρονική παραγγελία φαγητού αλλά και απέναντι στις εφαρμογές που εξυπηρετούν αυτό το σκοπό. Ο στόχος λοιπόν της έρευνας αυτής είναι μάλλον πολύπλευρος.

Αρχικά ζητούμενο αποτελούσε η εξερεύνηση της συμπεριφορά των Ελλήνων που διαθέτουν πρόσβαση στο διαδίκτυο μέσω υπολογιστή ή μέσω κάποιας έξυπνης κινητής συσκευής, σχετικά με την ηλεκτρονική παραγγελία φαγητού. Μέσω αυτής της έρευνας απαντήθηκαν ερωτήματα όπως με ποιά συχνότητα κάνουν χρήση του διαδικτύου για να παραγγείλουν φαγητό και πόσοι από αυτούς επιλέγουν ακόμα τον πατροπαράδοτο τρόπο της τηλεφωνικής κλήσης. Ποιοί είναι οι κύριοι λόγοι που τους απομακρύνουν από το να παραγγείλουν ηλεκτρονικά και ποιά θεωρούν ότι είναι τα πλεονεκτήματα και τα μειονεκτήματα αυτού του τρόπου παραγγελίας. Η συλλογή αυτής της πληροφορίας στόχευε στο να εντοπιστεί εάν το κοινό στο οποίο απευθύνεται η εφαρμογή ήταν αρκετό ώστε να αξίζει να στραφούμε προς τον τομέα της ηλεκτρονικής παραγγελίας φαγητού. Τα στοιχεία τα οποία προέκυπταν μέσω προγενέστερων πραγματοποιηθέντων ερευνών γύρω από την ηλεκτρονική παραγγελία φαγητού στη χώρα μας, όπως θα δούμε και παρακάτω, έδειχναν πως ο Έλληνας έχει εντάξει έως ένα βαθμό στην καθημερινότητα του την ηλεκτρονική παραγγελία φαγητού. Έμενε να επιβεβαιώσουμε εκ νέου αυτή την άποψη πριν προχωρήσουμε στην επόμενη φάση της ερευνάς καθώς και στην πραγμάτωση της εφαρμογής μας.

Επιπρόσθετος, αλλά διόλου δευτερεύον στόχος ήταν η εισαγωγή πληροφορίας από την πλευρά του τελικού χρήστη σχετικά με το ποιά κριτήρια θα πρέπει να πληροί μια εφαρμογή προκειμένου να είναι ελκυστική. Έτσι έπρεπε να τεθεί μια σειρά ερωτήσεων με διάφορα χαρακτηριστικά, τόσο γενικευμένα όσο και πιο ειδικά και μέσω των απαντήσεων που θα συλλέγονταν να γινόταν η αξιολόγηση και η ταξινόμηση αυτών βάσει σημαντικότητας. Τα πρώτα στην κατάταξη θα ήταν αυτά που κυρίως θα έπρεπε να καλυφθούν στην πρώτη φάση της υλοποίησής της εφαρμογής, ενώ σε επόμενη φάση, και πάλι εισάγοντας την άποψη του χρήστη, θα αξιολογούνταν και θα υλοποιούνταν τα δευτερευούσης σημασίας χαρακτηριστικά. Άλλωστε αυτή η σταδιακή διανομή λειτουργικού λογισμικού και η επαναλαμβανόμενη αλληλεπίδραση με τον χρήστη αποτελούν βασικά στοιχεία των ευέλικτων μεθοδολογιών που εκμεταλλευτήκαμε στην παρούσα εργα-

σία.

Στη συνέχεια παρουσιάζονται τα στάδια της έρευνας καθώς και η ερμηνεία των αποτελεσμάτων της εκάστοτε φάσης. Χωρίζεται σε δύο βασικές φάσεις με την πρώτη να αποτελεί μια αναγνωριστική διαδικασία της προδιάθεσης των ερωτηθέντων προς την πρότασή μας και τη δεύτερη να είναι μια επαναλαμβανόμενη διάδραση με τους πιθανούς χρήστες προκειμένου να επιτευχθούν οι προαναφερθέντες στόχοι. Η δεύτερη φάση απαρτίζεται από δύο βασικές επαναλήψεις-φάσεις για συλλογή πληροφορίας για την σχέση που έχουν οι ερωτηθέντες με την ηλεκτρονική παραγγελία φαγητού και τις διάφορες εφαρμογές που δραστηριοποιούνται στο τομέα αυτό, ενώ στη φάση αυτή τους παρουσιάζεται μιας πρώτης έκδοσης της εφαρμογής που υλοποιήσαμε.

6.2.2 Μεθοδολογία Έρευνας

Για να μπορέσουμε να προβούμε στην έρευνα αγοράς που θα μας δώσει μία εικόνα της γνώμης των πελατών σχετικά με την ηλεκτρονική μας πλατφόρμα, πρέπει αρχικά να ορίσουμε τον τύπο της έρευνας που θα διεξάγουμε. Καθώς ενδιαφερόμαστε τόσο για τη γνώμη όσο και για τα συναισθήματα των πελατών σχετικά με την εφαρμογή μας, αποφασίσαμε να προβούμε σε μικτό τύπο έρευνας αγοράς. Το δειγματοληπτικό πλαίσιο της έρευνάς μας δεν περιλαμβάνει το σύνολο του πληθυσμού της χώρας αλλά ενδιαφερόμαστε να έχουμε ανατροφοδότηση από άτομα που είναι είτε εργαζόμενοι του ιδιωτικού ή του δημόσιου τομέα είτε φοιτητές. Ως προς τη μέθοδο διεξαγωγής της έρευνας αγοράς, έγινε επιλογή χρήσης δείγματος χιονόμπαλας, ζητώντας από τους ερωτώμενους να προτείνουν και άλλα άτομα που μπορούν να συμμετάσχουν ώστε να ικανοποιηθεί το μέγεθος του δείγματός μας. Έπειτα, ως προς το μέγεθος του δείγματος, καταλήξαμε πως μας ικανοποιεί και θα μας προσφέρει αντικειμενικότητα ένα δείγμα που να αποτελείται από εξήντα άτομα και άνω. Καθώς ενδιαφερόμαστε για μια συνεχή συλλογή στοιχείων καθώς θα υπάρχουν και φάσεις επαναληπτικής δειγματοληψίας, καταλήγουμε πως ο βέλτιστος τρόπος περάτωσης της μικτής μας έρευνας αγοράς είναι η διεκπεραίωση σε βάθος συνεντεύξεων και ο διαμοιρασμός on-line ερωτηματολογίων.

Πρώτη φάση συλλογής – Αναγνώριση προδιάθεσης χρηστών: Στην παρούσα φάση ακολουθώντας την διαδικασία αναζήτησης πιθανών χρηστών [31] της εφαρμογής εγκαταλείψαμε τις υποθέσεις και ήρθαμε σε επαφή με τους εν δυνάμει πελάτες. Προβήκαμε στη διαδικασία συνεντεύξεων προκειμένου να γίνει μια αρχική αναγνώριση της προδιάθεσης του αγοραστικού κοινού. Θέσαμε σε αυτούς ένα σύνολο ερωτήσεων που στόχευαν στο να μοιραστούμε με αυτούς την ιδέα μας και να αξιολογήσουμε τις αντιδράσεις τους. Από αυτή τη φάση κατανοήσαμε πως υπάρχει διαθέσιμο πιθανό κοινό το οποίο θα ενδιαφερόταν για την ιδέα μας. Ταυτόχρονα μας δόθηκε μια γενική προσέγγιση των απαιτήσεων τους.

Δεύτερη φάση συλλογής – Επαναλαμβανόμενη διάδραση με χρήστες: Στόχος της συγκεκριμένης φάσης ήταν να επεκταθεί η αλληλεπίδραση με το πιθανό κοινό και να εμβαθύνουμε περισσότερο στην άποψή τους για την ηλεκτρονική παραγγελία φαγητού. Η φάση αυτή αποτελείται από δύο επαναλήψεις. Για την πρώτη επανάληψη έγινε σύνταξη ενός ερωτηματολογίου με ερωτήσεις οι οποίες αποβλέπουν στην διερεύνηση της εξοικείωση του δείγματος τόσο με την ηλεκτρονική παραγγελία φαγητού, τη χρήση εφαρμογών ή πλατφόρμων που υπάρχουν για το συγκεκριμένο σκοπό και την άποψή τους σχετικά με το πως πρέπει να είναι μια τέτοια εφαρμογή ώστε να είναι ελκυστική.

Η δεύτερη επανάληψη πραγματοποιήθηκε επίσης μέσω ερωτηματολογίου. Μέσω αυτού και βασιζόμενοι στις αρχές των ευέλικτων μεθοδολογιών ανάπτυξης λογισμικού, φροντίσαμε στην παρούσα φάση να τροφοδοτήσουμε τους ερωτηθέντες και με μια πρώτη λειτουργική έκδοση της εφαρμογής που αναπτύξαμε σε σύντομο χρονικό διάστημα μετά τις απαντήσεις που συλλέξαμε από αυτούς στην προηγούμενη επανάληψη. Έτσι με τη δημιουργία πρωτοτύπων (prototyping), τα οποία προέκυψαν από ήδη υλοποιημένο λειτουργικό κώδικα, παρουσιάστηκε ένα σενάριο χρήσης της εφαρμογής στους συμμετέχοντες. Μέτα την προσεκτική παρακολούθηση αυτού του σεναρίου οι ερωτηθέντες έπρεπε να απαντήσουν σε μια σειρά ερωτήσεων σχετικές με τα χαρακτηριστικά και τη λειτουργικότητα της εφαρμογής που τους παρουσιάστηκε. Ο βασικός στόχος αυτής της επανάληψης ήταν η προβολή της εφαρμογής σε εν δυνάμει χρήστες οι οποίοι θα μπορούσαν να εντοπίσουν τυχόν αδυναμίες ή να επιβεβαιώσουν ότι μέσω αυτής καλύπτουμε τις αρχικές βασικές του αξιώσεις. Οι απαντήσεις τους θα μπορούσαν να αξιοποιηθούν για την επερχόμενη δεύτερη έκδοση της εφαρμογής η οποία θα περιλαμβάνει και τις όποιες βελτιώσεις εκείνοι προτείνουν.

6.2.3 Διερεύνηση προϋπάρχουσας κατάστασης - Ηλεκτρονική παραγγελία φαγητού στην Ελλάδα

Η στρόφη προς τις ηλεκτρονικές παραγγελίες στη χώρα μας ήταν επακόλουθο της όλο και αυξανόμενης διείσδυσης του διαδικτύου και των «έξυπνων» κινητών τηλεφώνων στην ελληνική αγορά καθώς και στην εξοικείωση των καταναλωτών με τις διαδικτυακές αγορές και τα πλεονεκτήματα που αυτές προσφέρουν. Όπως προκύπτει από την έρευνα χρήσης τεχνολογιών πληροφόρησης και επικοινωνίας για το 2015 της ΕΛΣΤΑΤ [46] το 67,1%, του συνόλου των νοικοκυριών της χώρας, με ένα τουλάχιστον μέλος ηλικίας 16-74 ετών, χρησιμοποιεί ευρυζωνική σύνδεση παρουσιάζοντας, σε σχέση με το 2014, αύξηση 3,7%. Επιπροσθέτως, σύμφωνα με έρευνα της εταιρείας Temporo OMD το 2014 σε δείγμα 3.500 ανδρών και γυναικών, 18-54 ετών σχετικά με τη σχέση των Ελλήνων με την «κινητή» τεχνολογία, προκύπτει ότι το 51% των ελλήνων κατέχει «έξυπνο» κινητό με αυξητική

τάση στον τομέα αυτό.

Όπως είναι ευρέως γνωστό η Ελλάδα δείχνει να αποτελεί μια χώρα στην οποία η επιχειρηματικότητα στον τομέα της μαζικής εστίασης και διασκέδασης ανθίζει ακόμα και σε περιόδους ύφεσης. Σύμφωνα με έρευνα της Endeavor το 2012 μια στις τέσσερις επιχειρήσεις που δημιουργήθηκαν, άνηκαν στο χώρο αυτόν. Στα πλαίσια της κατάστασης αυτής ο Έλληνας έχει εντάξει στις συνήθειές του την «διανομή» (delivery) στο οποίο η χώρα μας φαίνεται να κατακτά τα πρωτεία στην Ευρώπη.

Ο συνδυασμός των προαναφερθέντων, δηλαδή η συνήθεια της «διανομής» και η δυναμική της ηλεκτρονικής παραγγελίας οδήγησε στο να παρατηρείται τα τελευταία χρόνια μια σημαντική αύξηση της ηλεκτρονικής παραγγελίας φαγητού στη χώρα μας ενώ ταυτόχρονα πολλοί επενδυτές έχουν στρέψει το ενδιαφέρον τους προς τις εταιρείες που δραστηριοποιούνται στη συγκεκριμένη αγορά. Σύμφωνα με τα αποτελέσματα της ετήσιας έρευνας ηλεκτρονικού εμπορίου που διεξάγεται από το Εργαστήριο Ηλεκτρονικού Εμπορίου (ELTRUN) του Οικονομικού Πανεπιστημίου Αθηνών, η ηλεκτρονική παραγγελία φαγητού έχει αυξήσει το ποσοστό της κατά 40% σε σχέση με το 2010 [50] με ένα σεβαστό ποσοστό αγοραστών να κάνει χρήση του κινητού του για το σκοπό αυτό. Επίσης χρόνο με το χρόνο πληθώρα εφαρμογών που εξυπηρετούν το σκοπό αυτό εμφανίζονται, ενώ όλο και πιο πολλά εστιατόρια, ακολουθώντας τη ροή της τεχνολογίας, προσφέρουν πλέον τη δυνατότητα ηλεκτρονικής παραγγελίας στους πελάτες τους είτε μέσω των πλατφορμών-εφαρμογών αυτών είτε μέσω δικού τους ηλεκτρονικού καταστήματος. Φυσικά για λόγους κόστους δημιουργίας και συντήρησης ενός ηλεκτρονικού καταστήματος αλλά και για λόγους διαφήμισης τα πιο πολλά εστιατόρια εναποθέτουν την δυνατότητα ηλεκτρονικής παραγγελίας τους σε υπάρχουσες εταιρείες που έχουν δημιουργήσει εφαρμογές και πλατφόρμες ειδικά για το σκοπό αυτό.

Μερικά ισχυρά παραδείγματα εταιρειών (start-ups) που δραστηριοποιούνται στη χώρα μας στον τομέα της ηλεκτρονικής παραγγελίας φαγητού είναι τα εξής:



Πρόκειται για ένα web portal που επιτρέπει την ηλεκτρονική παραγγελία φαγητού από διάφορα συνεργαζόμενα καταστήματα. Η ClickDelivery.gr⁷ ιδρύθηκε το 2009, τέθηκε σε πιλοτική λειτουργία το πρώτο εξάμηνο του 2010 και σε πλήρη λειτουργία το Σεπτέμβριο του ίδιου χρόνου. Για να

⁷<https://www.clickdelivery.gr/>

πραγματοποιηθεί η παραγγελία ο χρήστης καλείται να εισάγει την διεύθυνσή του στην πλατφόρμα, βλέπει τα διαθέσιμα εστιατόρια, επιλέγει το γεύμα του και αποστέλλει την παραγγελία. Η παραγγελία εκτυπώνεται στο κατάστημα στο τερματικό της Click Delivery. Το κατάστημα αποδέχεται την παραγγελία και πληκτρολογεί τον πιθανό χρόνο παράδοσης μέσω του τερματικού και έτσι ο χρήστης ενημερώνεται σε πραγματικό χρόνο μέσω της πλατφόρμας. Διαθέτει επίσης αντίστοιχη εφαρμογή συμβατή με έξυπνα κινητά. Η Click Delivery Α.Ε. χαρακτηρίζεται σήμερα ως η ηγέτιδα δύναμη στο online food delivery στη χώρα μας με με περισσότερα από 650 συνεργαζόμενα εστιατόρια σε Αθήνα, Θεσσαλονίκη, Λάρισα και Ιωάννινα., με ετήσιους ρυθμούς ανάπτυξης 430% και με επενδυτικές συμφωνίες εκατομμυρίων. Τον Οκτώβριο του 2013 υπέγραψε συμφωνία τριών εκατομμυρίων ευρώ με την τουρκική εταιρεία Yemeksepeti η οποία δραστηριοποιείται σε έξι ευρωπαϊκές χώρες και έχει εξαγοραστεί από την γερμανική Delivery Hero δραστηριοποιείται σε 34 χώρες, με περισσότερους από 1.500 εργαζόμενους στο δυναμικό της ⁸.



Αποτελεί επίσης μια από τις μεγαλύτερες δυνάμεις στη χώρα μας στον τομέα της ηλεκτρονικής παραγγελίας φαγητού. Δημιουργήθηκε το 2011 στηριζόμενο στο e-shop.gr Ακολουθεί την ίδια λογική παραγγελίας όπως η προαναφερθείσα της ClickDelivery.gr ενώ και αυτή διαθέτει αντίστοιχη εφαρμογή για κινητό. Στα 3,5 χρόνια λειτουργίας του, το e-FOOD έχει εξυπηρετήσει περισσότερες από 2.500.000 παραγγελίες. Σήμερα, εξυπηρετεί περισσότερες από 11.000 παραγγελίες την ημέρα, με 3.000 συνεργαζόμενα καταστήματα σε 42 πόλεις και 200.000 εγγεγραμμένους χρήστες, ενώ απασχολεί 67 άτομα προσωπικό με μέση ηλικία 27 ετών. Το 2015 εξαγοράστηκε από την γερμανική Delivery Hero μαζί με την τουρκική Yemeksepeti η οποία όπως προαναφέραμε αποτελεί έναν εκ των βασικών μετόχων του clickdelivery.gr. Το e-FOOD ⁹ και το clickdelivery.gr είναι τα δύο μεγαλύτερα sites, τα οποία εκτιμάται ότι ελέγχουν ποσοστό άνω του 70% στην αγορά του online food delivery στην Ελλάδα ¹⁰.

⁸<http://www.oneman.gr/Publi/to-clickdelivery-gr-ypegrapse-ependytikh-symfwnia-methn-etaireia-yemeksepeti.2438874.html>

⁹<https://www.e-food.gr/>

¹⁰<http://www.sepe.gr/gr/information/news/article/2709563/diethneis-paiktes-proselkuei-i-elliniki-agora-tou-ilektronikou-delivery/>



Ο «ιπτάμενος» μάγειρας φαίνεται να κατέχει μια σημαντική θέση στην ηλεκτρονική παραγγελία φαγητού στην Ελλάδα. Συνεργάζεται με πάνω από 1.000 καταστήματα με κάποια από αυτά να είναι καταξιωμένες αλυσίδες φαγητού όπως Domino's, Pizza Hut, L'artigiano, Benvenuto, Noodle Bar και άλλες. Δεν διαφοροποιείται ιδιαίτερα από τις δυο προαναφερθείσες εταιρείες στη διαδικασία παραγγελίας όμως διαθέτει ένα σύστημα επιβράβευσης όπου ο χρήστης μέσω των αγορών του συλλέγει πόντους οι οποίοι εξαργυρώνονται είτε με κουπόνια φαγητού τα οποία του αποστέλλονται ταχυδρομικά είτε με δωρεά στο «Χαμόγελο του Παιδιού»¹¹.

Διάφορες ακόμα πλατφόρμες ή εφαρμογές παρόμοιας λογικής με τις προαναφερθείσες είναι οι ClicknEat.gr, FooDoodelivery.gr, YourDelivery.gr, fagi.gr οι οποίες δραστηριοποιούνται στη χώρα μας έχοντας όμως μικρότερο ποσοστό εγγεγραμμένων χρηστών. Κατά προσέγγιση η ελληνική αγορά delivery ξεπερνά τις διακόσιες χιλιάδες παραγγελίες ημερησίως. Η ηλεκτρονική παραγγελιοληψία υπερβαίνει το 30% αυτού του συνόλου με βλέψεις αυξητικές. Συνεπώς τα δείγματα από τις εμπειρίες των ήδη υπάρχουσών εταιρειών μαρτυρούν πως το έδαφος είναι πρόσφορο και πως μελλοντικά η ηλεκτρονική παραγγελία θα κυριαρχήσει έναντι των πατροπαράδοτων μεθόδων.

Όλα τα παραπάνω δείχνουν πως η ανάπτυξη της ηλεκτρονικής παραγγελίας φαγητού είναι αδιαμφισβήτητη, παρ' ολ' αυτά τα προαναφερθέντα δεν αποτελούν εγγύηση πως όποιο προϊόν λογισμικού, εφαρμογή ή πλατφόρμα, που εξυπηρετεί το σκοπό αυτό βγαίνοντας στην αγορά θα έχει εξασφαλισμένη επιτυχία. Ο εν' δυνάμει πελάτης και πάλι αποτελεί μια αξιόπιστη πηγή πληροφόρησης και καθοδήγησης σχετικά με τα στοιχεία εκείνα που μπορούν να καταστήσουν ένα τέτοιο προϊόν αποδοτικό, ελκυστικό και επιτυχημένο. Η σωστή εκμετάλλευση των αναγκών, των απόψεων και των επιθυμιών αυτού μπορεί να οδηγήσει σε προϊόν το οποίο ξεφεύγει από το κοινότυπο με το λιγότερο δυνατό κόστος και κόπο από την πλευρά του δημιουργού του. Τα επόμενα κεφάλαια αναλύουν με ποίον τρόπο επετεύχθη στην παρούσα εργασία η αξιοποίηση του τελικού πελάτη προς όφελος αμφοτέρων.

¹¹<https://www.deliveras.gr/loyalty.php>

6.3 Αξιολόγηση Εμπειρικής Ανάλυσης

6.3.1 Δεύτερη Φάση - Πρώτη Διάδραση

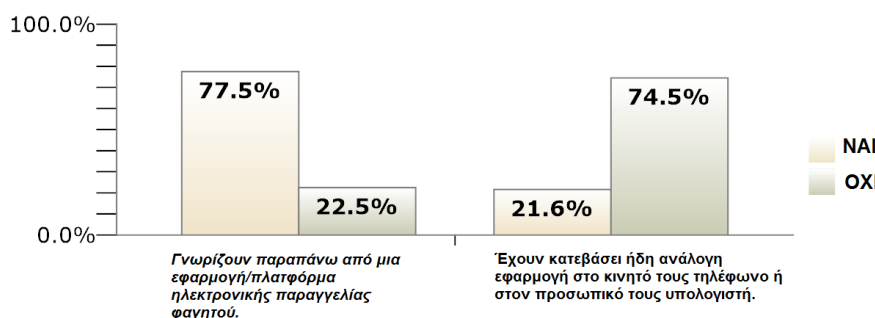
Η διαδικασία αυτή γίνεται μέσω ενός ερωτηματολογίου αποτελούμενο από είκοσι επτά ερωτήσεις με τις είκοσι έξι από αυτές να είναι πολλαπλής επιλογής και την τελευταία να είναι ανοιχτού τύπου. Το συγκεκριμένο ερωτηματολόγιο επισυνάπτεται στο τέλος της παρούσας εργασίας. Ο συνολικός αριθμός των ανθρώπων που συμπλήρωσαν το ερωτηματολόγιο ανέρχεται στα 204. Μέρος του δείγματος αποτελείται από εργαζόμενους στον ιδιωτικό και δημόσιο τομέα, φοιτητές και γενικότερα άτομα τα οποία έχουν πρόσβαση στο διαδίκτυο και κατά τις ώρες του μεσημεριανού κυρίως γεύματος βρίσκονται εκτός της εστίας τους. Η προσέγγιση τους στόχευε στο να εντοπιστούν τα λάθη και οι παραλείψεις ή οι υπερβολές που έγιναν στην πλατφόρμα που αναπτύξαμε χωρίς να βασιστούμε στις ευέλικτες μεθοδολογίες και την οποία παρουσιάσαμε παραπάνω. Έτσι μια σειρά ερωτήσεων είναι πιο στοχευμένη προς αυτό το σκοπό ενώ οι υπόλοιπες είναι άμεσα συνδεδεμένες με τους γενικούς στόχους της έρευνας.

Οι γενικευμένες ερωτήσεις εστιάζουν στην συχνότητα χρήσης του διαδικτύου από το δείγμα για την παραγγελία φαγητού, τις εμπειρίες που γενικότερα έχουν με τη συγκεκριμένη διαδικασία και την ενημέρωση σχετικά με τις υπάρχουσες εφαρμογές ηλεκτρονικής παραγγελίας φαγητού. Επίσης τίθενται ερωτήσεις σχετικά με τα χαρακτηριστικά που θέλει ο χρήστης να έχει μια τέτοια εφαρμογή ώστε να αυξήσει τα επίπεδα ικανοποίησης του και που εν τέλει θα τον οδηγήσουν στο να τη χρησιμοποιήσει. Με τον τρόπο αυτό αναδεικνύονται υπάρχουσες αδυναμίες του ανταγωνισμού ενώ ταυτόχρονα σκιαγραφείται η μορφή της εφαρμογής ή πλατφόρμας που είναι πιο φιλική στο χρήστη. Οι δημογραφικές ερωτήσεις του ερωτηματολογίου δείχνουν ότι υπάρχει μια σχεδόν ισότιμη κατανομή ως προς το φύλο των ερωτηθέντων, καθώς το 56,4% εξ αυτών είναι γυναίκες και το 43,6% άντρες. Οι ηλικίες τους κατανέμονται ως εξής: 8-27 είναι το 56.9%, 28-37 το 32.4%, 38-47 το 8.3% ενώ άνω των 48 το 2.5%. Η πλειονότητα των συμμετεχόντων ζει σε προάστια (48,5%) είτε σε αστικά κέντρα (35,3%) ενώ οι υπόλοιποι (16,2%) ζουν είτε σε μικρές πόλεις και αγροτικές περιοχές.

Μέσω των απαντήσεων που συλλέχθηκαν παρατηρείται ότι παρά το γεγονός ότι παρατηρείται αύξηση του ποσοστού των ανθρώπων που θα κάνουν χρήση κάποιας εφαρμογής ή πλατφόρμας προκειμένου να παραγγείλουν το φαγητό τους, η πλειονότητα δεν έχει ακόμα απομακρυνθεί ακόμα από τον πατροπαράδοτο τρόπο της τηλεφωνικής κλήσης. Συνεχίζει λοιπόν ακόμα στη χώρα μας το τηλέφωνο να αποτελεί τον μεγαλύτερο ανταγωνιστή της ηλεκτρονικής παραγγελίας φαγητού. Για την ακρίβεια, 63,7% των ερωτηθέντων υποστήριξαν ότι ακόμα προτιμούν ως επί το πλείστον την τηλεφωνική κλήση ενώ μόλις ένα 18,7% κάνει χρήση κυρίως του διαδικτύου για τις παραγγελίες του. Το υπολειπόμενο ποσοστό προτιμάει είτε να πη-

γαίνει απευθείας στο κατάστημα για να παραγγείλει και να παραλάβει το φαγητό του είτε αποφεύγει εντελώς την αγορά φαγητού από καταστήματα εστίασης.

Το προαναφερθέν γεγονός μπορεί να φαντάζει αποκαρδιωτικό παρ'όλα αυτά το 37,7% του δείγματος έχει πραγματοποιήσει ηλεκτρονική παραγγελία φαγητού παραπάνω από μια φορές ενώ ένα 18,1% ακολουθεί αυτή τη μέθοδο συστηματικά. Συνεπώς φαίνεται η μέθοδος αυτή παραγγελίας να μην έχει ωριμάσει ακόμα ιδιαίτερα και να αδυνατεί να αντικαταστήσει άμεσα την βαθειά ριζωμένη συνήθεια της τηλεφωνικής παραγγελίας, όμως υπάρχει μια θετική προδιάθεση και μια αυξητική τάση. Αυτό διαφαίνεται από το ότι ένα πολύ μεγάλο ποσοστό, όπως δείχνει και το παρακάτω σχεδιάγραμμα [Σχήμα 6.1] είναι ενήμερο για την ύπαρξη εφαρμογών ή πλατφορμών που δραστηριοποιούνται στον τομέα της ηλεκτρονικής παραγγελίας φαγητού. Στο ίδιο σχεδιάγραμμα φαίνεται και η απάντηση αυτών για το αν έχουν κατεβάσει κάποια τέτοια εφαρμογή στο κινητό ή τον υπολογιστή τους.



Σχήμα 6.1: Ποσοστά ερωτηθέντων που είναι ενήμεροι ή έχουν κατεβάσει εφαρμογές ηλεκτρονικής παραγγελίας φαγητού.

Είναι μεν μικρό το ποσοστό αλλά η γνώση για την ύπαρξη σε συνδυασμό με το χρόνο ωρίμανσης της συνήθειας το ποσοστό αυτό πιθανότατα θα αυξηθεί κατά πολύ.

Από τα παραπάνω προκύπτει ότι το ελληνικό κοινό δείχνει να έχει μια φιλική προδιάθεση προς την ηλεκτρονική παραγγελία φαγητού, επί της ουσίας όμως ακόμα διστάζει να την εφαρμόσει σε πολύ συχνή βάση στην καθημερινότητά του. Αντ'αυτού εμμένει σε τρόπους χρόνια δοκιμασμένους και συνηθισμένους όπως το τηλέφωνο ή η άμεση παραγγελία και παραλαβή από το κατάστημα. Οι πιο σημαντικοί από τους λόγους που τους οδηγούν στο να είναι τόσο διστακτικοί παρουσιάζονται παρακάτω, όπως οι ίδιοι τους ανέφεραν σε ανοιχτή ερώτηση για το τι θεωρούν αρνητικό στον τρόπο αυτό παραγγελίας.

Ο κυριότερος εκ των λόγων που το δείγμα υποστήριξε ότι το απωθεί

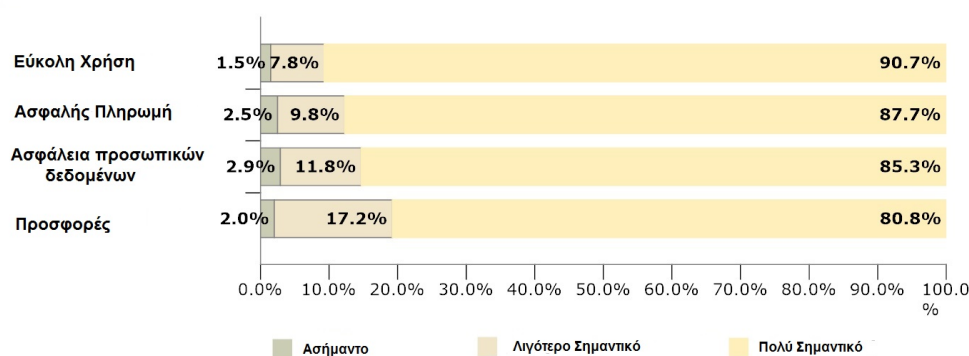
από την ηλεκτρονική παραγγελία είναι η αίσθηση ανασφάλειας τόσο όσον αφορά τα προσωπικά τους δεδομένα αλλά και την διαδικασία πληρωμής. Θεωρούν πως είναι πολλά τα στοιχεία που χρειάζεται να συμπληρώσουν προκειμένου να προβούν σε παραγγελία, ότι τα στοιχεία αυτά παραμένουν καταγεγραμμένα και πως δεν είναι διόλου ασφαλή, πως δεν είναι σαφείς οι όροι χρήσης και η πολιτική προστασίας προσωπικών δεδομένων. Έτσι δημιουργείται σε αυτούς έλλειψη εμπιστοσύνης και έντονη ανασφάλεια.

Στα πλαίσια αυτής της ανασφάλειας προσθέτουν ότι, σύμφωνα πάντα από οπτική γωνία των ερωτηθέντων, το προσωπικό των ελληνικών εστιατορίων δεν είναι επαρκώς εξοικειωμένο με τη διαχείριση των ηλεκτρονικών παραγγελιών. Έτσι φοβούνται πως, από τη στιγμή που ο τρόπος αυτός παραγγελίας δεν έχει εδραιωθεί, οι παραγγελίες τους κατά πάσα πιθανότητα θα αγνοηθούν ή θα εκτελεστούν λανθασμένα, εξού και αποτρέπονται στην παραδοσιακή μέθοδο της τηλεφωνικής κλήσης, η οποία λόγω του προαναφερθέντος φαντάζει πιο αποδοτική παρά το γεγονός ότι αναγνωρίζουν τα παρελκόμενα μειονεκτήματά της. Ταυτόχρονα θεωρούν την ηλεκτρονική παραγγελία ιδιαιτέρως απρόσωπη, μη αξιόπιστη, πιο βαρετή, πολύπλοκη με ελλειπή πληροφόρηση σχετικά με το προϊόν που θέλουν να αγοράσουν. Υποστηρίζουν ότι έτσι χάνεται η πιο άμεση επαφή με το προσωπικό του καταστήματος, ότι χάνεται η ευελιξία των επιλογών που υπάρχει στην τηλεφωνική επικοινωνία, όπως λεπτομέρειες προσωπικών προτιμήσεων για το φαγητό (χρόνος ψήσιματος, ιδιαίτεροι συνδυασμοί κ.τ.λ), ότι αδυνατείς να έχεις την άποψη ή κάποια πρόταση από τον συνομιλητή (προσωπικό κατάστημα).

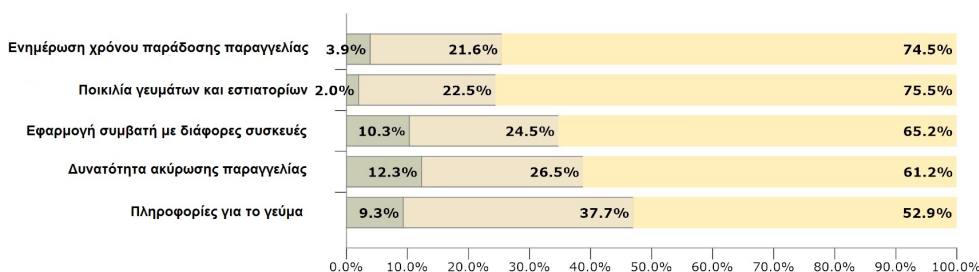
Ουκ ολίγοι εκ των ερωτηθέντων, προβάλλοντας ένα λίγο πιο ανθρώπινο πρόσωπο, υποστήριξαν ότι αν αρχίζοντας να υιοθετούν αυτό τον τρόπο παραγγελίας θα συμβάλλουν στην αύξηση της ανεργίας καθώς θα υπάρξει μείωση των θέσεων εργασίας. Επίσης τόνισαν πως πολλές από τις υπάρχουσες εφαρμογές δεν διαθέτουν επαρκή ενημέρωση, είναι αργές, είναι πρόχειρα φτιαγμένες και συχνά δυσλειτουργικές και δεν υποστηρίζουν τις ηλεκτρονικές πληρωμές μέσω PayPal. Ταυτόχρονα υπάρχει σε αυτές απουσία φιλοσοφίας συλλογής πόντων (credits) ανά χρήστη ανάλογα με τις παραγγελίες του και εξαργύρωσή τους σε μια επερχόμενη παραγγελία, όπως κάποιο δωρεάν γλυκό ή γεύμα ή κάποια έκπτωση, κάτι που ήδη εφαρμόζεται σε ηλεκτρονικές παραγγελίες ρούχων ή σε κλείσιμο ξενοδοχείων. Υποστήριξαν λοιπόν κάποιοι ότι ο μόνος λόγος που θα χρησιμοποιούσαν τέτοιες εφαρμογές θα ήταν κάποια γενναία έκπτωση η οποία δεν θα ήταν διαθέσιμη μέσω τηλεφωνικής παραγγελίας.

Δεν είναι λίγοι ούτε ασήμαντοι οι λόγοι για τους οποίους το ερωτηθέν ελληνικό κοινό αντιστέκεται ακόμα στην ηλεκτρονική παραγγελία φαγητού. Κάποιοι από αυτούς μπορούν εύκολα να ανατραπούν, αυτό όμως που προκύπτει από τις προαναφερόμενες ανησυχίες τους είναι ότι επιθυμούν οπωσδήποτε ασφάλεια σε διάφορα επίπεδα, ακρίβεια και εγγυήσεις προκειμένου να στραφούν προς τον νέο τρόπο παραγγελίας φαγητού.

Προκειμένου να εντοπιστούν πιο συγκεκριμένα τα στοιχεία εκείνα που θα κάλυπταν τις όποιες ανασφάλειες του μέσου χρήστη τέτοιων εφαρμογών, στα πλαίσια του ερωτηματολογίου τέθηκαν στην κρίση του δείγματος της έρευνας ένα σύνολο χαρακτηριστικών με τη δυνατότητα να αξιολογηθούν από αυτούς με βαθμό σημαντικότητας. Τα διαγράμματα που ακολουθούν [Σχήμα 6.2, Σχήμα 6.3, Σχήμα 6.4] δείχνουν τις απαντήσεις τους και οδηγούν σε μια κατάταξη των αναφερόμενων χαρακτηριστικών διόλου ανέλπιστη, αν σκεφτεί κανείς όσα παρουσίασαν ως αρνητικά της ηλεκτρονικής παραγγελίας φαγητού



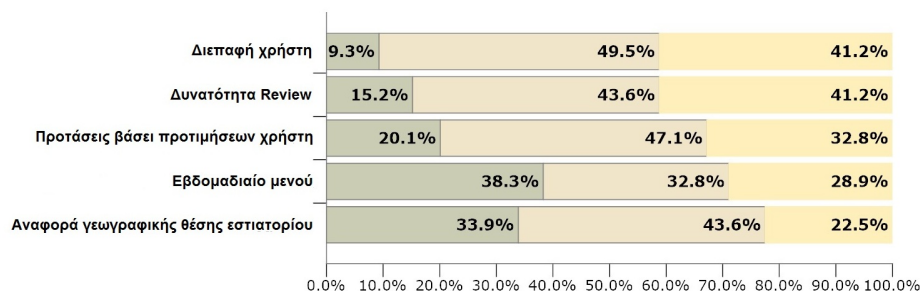
Σχήμα 6.2: Χαρακτηριστικά που αξιολογήθηκαν ως υψηλής σημασίας.



Σχήμα 6.3: Χαρακτηριστικά που αξιολογήθηκαν ως μέτριας σημασίας.

Παρατηρώντας τα παραπάνω σχεδιαγράμματα μπορεί κανείς να κατανοήσει τι είναι πιο σημαντικό για τους εν δυνάμει χρήστες και να δημιουργήσει ένα προσχέδιο μιας εφαρμογής που ως επί το πλείστον θα προσελκύσει το ενδιαφέρον και θα ικανοποιήσει τους πιθανούς τελικούς καταναλωτές. Διαφαίνεται μέσα από τις απαντήσεις ότι επιθυμίες που προέχει να ικανοποιηθούν είναι η απόδοση μια αίσθησης ασφάλειας, η ευκολία χρήσης, η αίσθηση ελέγχου της παραγγελίας και κατά τη διάρκεια αυτής αλλά και μετά την εκτέλεσή της και οι πιθανές προσφορές.

Όπως καταγράφεται στο Σχήμα 6.2 ιδιαίτερος σημαντικό για τους χρήστες είναι η ευκολία στη χρήση της εκάστοτε εφαρμογής. Ένα προϊόν λογισμικού θεωρείται «εύκολο» όταν εξοικονομεί χρόνο στο χρήστη [48].



Σχήμα 6.4: Χαρακτηριστικά που αξιολογήθηκαν ως λιγότερο σημαντικά.

Συνεπώς ένα προϊόν ή υπηρεσία μεγάλης πολυπλοκότητας χωρίς ευκολία στη χρήση θα μπορούσε να απωθήσει πολλούς από τους δυνητικούς χρήστες καθώς θα είναι χρονοβόρο. Στην περίπτωση της παραγγελίας φαγητού οι ερωτηθέντες επιθυμούν μια εφαρμογή/πλατφόρμα γρήγορης απόκρισης που να παρέχει εύκολη διαδικασία παραγγελίας. Αυτό σημαίνει πως επιθυμούν εύκολο περιβάλλον πλοήγησης, λίγα βήματα για την ολοκλήρωση μιας παραγγελίας και όχι χρονοβόρες εγγραφές. Άλλωστε έχει παρατηρηθεί ότι ένας από τους βασικούς λόγους που πολλοί εγκαταλείπουν το καλάθι των ηλεκτρονικών τους αγορών είναι η πολύπλοκη διαδικασία παραγγελίας λόγω πολλών βημάτων και απαίτησης συμπλήρωσης πολλών πεδίων¹¹. Συνήθως οι χρήστες είναι αρκετά ανυπόμονοι με αποτέλεσμα μια διαδικασία παραγγελίας πολυβηματική να τους κουράζει και να αποτρέπει την ολοκλήρωσή της.

Ένας επίσης λόγος που οδηγεί σε υψηλά επίπεδα εγκατάλειψης του ηλεκτρονικού καλαθιού είναι η απαίτηση από την εκάστοτε εφαρμογή ή πλατφόρμα για δημιουργία λογαριασμού. Από τη μια η συλλογή πληροφοριών που αφορούν στο χρήστη είναι μια ευνοϊκή μέθοδος για συλλογή πληροφοριών γύρω από αυτόν και τις προτιμήσεις του, παρ'ολ'αυτά ουκ ολίγοι είναι εκείνοι που θα αποφύγουν εντελώς ηλεκτρονικά καταστήματα που δεν επιτρέπουν την αγορά χωρίς προαπαιτούμενη εγγραφή. Ο λόγος που πολλοί θέλουν να αποφύγουν τις εγγραφές είναι η έλλειψη εμπιστοσύνης όπως ξεκάθαρα αναφέρθηκε σε προηγούμενη παράγραφο. Έτσι είναι λογικό να βρίσκονται στις πρωταρχικές τους προτιμήσεις την ασφάλεια των προσωπικών δεδομένων καθώς και την παροχή ασφαλούς πληρωμής.

Το ζήτημα της εμπιστοσύνης σε σχέση με την ασφάλεια των προσωπικών δεδομένων σε ενέργειες στο διαδίκτυο είναι ένα θέμα μακροσκελές το οποίο έχει απασχολήσει πολλές φορές. Οι πιο πολλοί χρήστες του διαδικτύου δηλώνουν διστακτικοί στο να παράσχουν τα προσωπικά τους δεδομένα λόγω φόβων όπως η κλοπή της ταυτότητάς τους, χρηματική κλοπή μέσω των πιστωτικών τους καρτών, ο κίνδυνος να πέσουν θύματα απάτης μέσω ηλεκτρονικού ταχυδρομείου. Όπως αναφέρεται όμως σε έρευνα του

¹¹<https://www.sitepoint.com/10-reasons-people-abandon-online-shopping-carts/>

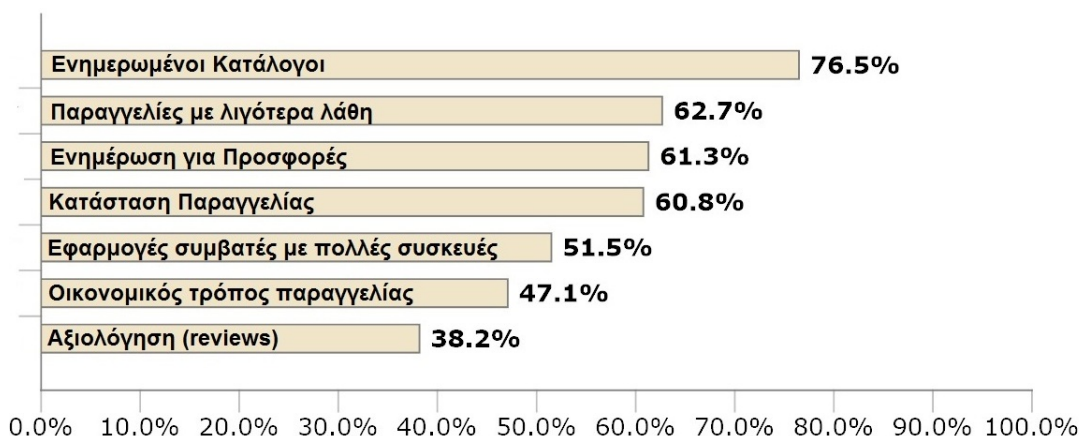
«ITProPortal» το 2013, το οξύμωρο είναι πως παρά τους φόβους τους οι καταναλωτές δείχνουν πρόθυμοι να μοιραστούν τα προσωπικά τους δεδομένα παρά την έλλειψη εμπιστοσύνης καθώς η συνήθεια του διαδικτύου και οι «ανέσεις» που αυτό τους παρέχει φαίνεται να υπερκαλύπτουν τις ανησυχίες τους. Αυτό βέβαια δεν έχει καθησυχάσει τους παρόχους ή τους οργανισμούς που δραστηριοποιούνται στο διαδίκτυο καθώς κατανοούν πως αρκεί ένα περιστατικό παραπόνου για μη παροχή επαρκούς ασφάλειας μπορεί να τους κοστίσει τόσο χρηματικά όσο και στη φήμη τους. Έτσι και τα Ελληνικά ηλεκτρονικά καταστήματα λαμβάνουν πολύ σοβαρά υπόψη τους την σχέση εμπιστοσύνης με τους πελάτες τους και αυτό φαίνεται από το γεγονός ότι τα τελευταία χρόνια παρατηρείται μια αύξηση της λήψης μέτρων σχετικών με ζητήματα ασφάλειας κατά τη διάρκεια ηλεκτρονικών αγορών [Σχήμα 6.5].



Σχήμα 6.5: Πηγή: “Καταγραφή και αξιολόγηση των Ελληνικών Ηλεκτρονικών Καταστημάτων - 2014”, Εργαστήριο Ηλεκτρονικού Εμπορίου (ELTRUN), Οικονομικό Πανεπιστήμιο Αθηνών.

Από τα παραπάνω προκύπτει ότι οι ανησυχίες των χρηστών είναι συγκεκριμένες και ίσως τους οδηγούν στην αποφυγή της ηλεκτρονικής παραγγελίας γενικότερα κατ’ επέκταση και φαγητού. Όμως μέσα από το τελευταίο τμήμα του ερωτηματολογίου, γίνεται εμφανές ότι πολλοί από τους ερωτηθέντες, παρά το γεγονός ότι ακόμα δεν έχουν ενστερνιστεί πλήρως την ηλεκτρονική παραγγελία φαγητού, δείχνουν να κατανοούν τα πλεονεκτήματα αυτής της μεθόδου παραγγελίας. Σύμφωνα με την (εικόνα 5) ένα πολύ μεγάλο ποσοστό (76.5%) θεωρεί ότι μέσα από τα διάφορα συστήματα ηλεκτρονικής παραγγελίας μπορούν να έχουν καλύτερη ενημέρωση

τόσο για τους τιμοκαταλόγους όσο και για τη διαθεσιμότητα. Ο συνηθισμένος τρόπος των φυλλαδίων υστερεί ιδιαίτερα στο κομμάτι αυτό ως είναι λογικό. Επιπροσθέτως 62,7% υποστήριξαν ότι μέσω των ηλεκτρονικών παραγγελιών περιορίζονται παρανοήσεις οι οποίες μπορεί να οφείλονται σε κακή συνεννόηση με το προσωπικό του καταστήματος είτε για οποιονδήποτε άλλο λόγο. Έτσι συμφωνούν ότι οι πιο πολλές παραγγελίες έχουν λιγότερα λάθη σε σχέση με όσες γίνονται μέσω τηλεφωνικής επικοινωνίας.



Σχήμα 6.6: Πλεονεκτήματα χρήσης εφαρμογών ή πλατφορμών ηλεκτρονικής παραγγελίας φαγητού σύμφωνα με τους ερωτηθέντες.

Ένα ακόμα θετικό γνώρισμα που υποστηρίζουν οι ερωτηθέντες είναι το γεγονός ότι μέσω των μέσω ηλεκτρονική παραγγελίας υπάρχει δυνατότητα να ενημερώνονται συχνότερα και άμεσα για τις τρέχουσες προσφορές. Έτσι οι παραγγελίες τους μπορούν να επηρεαστούν από την ενημέρωση αυτή. Επίσης όπως είδαμε και παραπάνω, οι καταναλωτές ενδιαφέρονται να έχουν κάποιο έλεγχο στην πορεία της παραγγελίας τους. Ταυτόχρονα αντιλαμβάνονται ότι τα μέσα ηλεκτρονικής παραγγελίας μπορούν να τους προσφέρουν τον έλεγχο αυτό είτε σε μικρό είτε σε μεγαλύτερο βαθμό η καθεμία. Γι' αυτό και βλέπουμε ένα μεγάλο ποσοστό να το συμπεριλαμβάνει στα κυρίαρχα θετικά στοιχεία (60,8%).

Σχεδόν οι μισοί εκ των συμμετεχόντων θεωρούν πλεονέκτημα το γεγονός ότι μπορούν να εκτελέσουν μια παραγγελία τους μέσα από κάθε συσκευή που χρησιμοποιούν. Δηλαδή τους ενδιαφέρει να μπορεί το εκάστοτε προϊόν λογισμικού που εξυπηρετεί την ηλεκτρονική παραγγελία να είναι συμβατό με όλες τις πιθανές συσκευές («έξυπνο» κινητό, προσωπικό υπολογιστή, ηλεκτρονική πλακέτα κ.τ.λ) . Έτσι η παραγγελία τους μπορεί να γίνει από οποιοδήποτε μέρος διαθέτει ευρυζωνική σύνδεση μέσω όποιας συσκευής έχουν διαθέσιμη. Δεν είναι εξαρτημένη από την τηλεφωνική κλήση, την οποία μάλιστα τη θεωρούν πιο ακριβό τρόπο παραγγελίας.

Τέλος σε συμφωνία με τα αποτελέσματα της έρευνάς μας η δυνατότητα κριτικής μέσω κάποιας προσφερόμενης διαδικασίας reviews, φαίνεται να είναι στα τελευταία επίπεδα στις προτιμήσεις των ερωτηθέντων [Σχήμα 6.6].

6.3.2 Αποτελέσματα Πρώτης Διάδρασης

Συνοψίζοντας όλα τα παραπάνω συμπεραίνουμε πως, παρά το γεγονός ότι υπάρχει ήδη στη χώρα μας ένα μεγάλο ποσοστό που διαθέτει ευρυζωνική σύνδεση και παρέχεται μια πληθώρα πλατφορμών και εφαρμογών μέσω των οποίων μπορεί η παραγγελία φαγητού να εκτελεστεί ηλεκτρονικά, η τηλεφωνική κλήση δείχνει να κατέχει ακόμα την πρώτη επιλογή για τους έλληνες καταναλωτές. Βέβαια, είναι εμφανές πως η ηλεκτρονική παραγγελία παρουσιάζει μια ανοδική πορεία στις προτιμήσεις τους πράγμα το οποίο επιβεβαιώνεται από το γεγονός ότι η πλειοψηφία τους διαθέτει την απαραίτητη ενημέρωση γύρω από αυτήν και ότι ένα σημαντικό ποσοστό χρησιμοποιεί αυτό τον τρόπο παραγγελίας είτε sporadικά είτε σε πιο συχνή βάση. Έτσι μπορεί να ειπωθεί ότι η ηλεκτρονική παραγγελία φαγητού έχει κάνει ένα πρώτο σημαντικό βήμα για να εδραιωθεί σταδιακά στην ελληνική καθημερινότητα.

Όσον αφορά τις εφαρμογές που παρέχουν τη δυνατότητα ηλεκτρονικής παραγγελίας, οι καταναλωτές δείχνουν να κατανοούν τα πλεονεκτήματα τους έναντι των υπολοίπων μεθόδων, παρ'ολ'αυτά για να τις εντάξουν στην καθημερινότητά τους ζητούν να καλυφθούν συγκεκριμένες απαιτήσεις τους. Επιθυμούν εφαρμογές όπου θα χαρακτηρίζονται από απλότητα στη χρήση, με λίγα αναγκαία βήματα για ολοκλήρωση της παραγγελίας, ενώ ταυτόχρονα θα τους εξασφαλίζουν μια αίσθηση ασφάλειας στις πληρωμές αλλά και στην προστασία των προσωπικών τους δεδομένων. Αυτά, σε συνδυασμό με τη δυνατότητα έλεγχου από το χρήστη της πορείας της παραγγελίας του είναι τα χαρακτηριστικά εκείνα της εφαρμογής που θα τον δελέαζαν στο να επιλέξει να πραγματοποιήσει τις παραγγελίες του μέσω αυτής. Συνεπώς αυτά αποτελούν μια πολύ καλή βάση πάνω στην οποία μπορεί ο καθένας που οραματίζεται την δημιουργία μιας εφαρμογής ηλεκτρονικής παραγγελίας φαγητού.

6.4 Αξιοποίηση Εμπειρικής Ανάλυσης

6.4.1 Δεύτερη Φάση - Δεύτερη Διάδραση

Η συγκεκριμένη φάση βασίζεται στη λογική της επαναλαμβανόμενης διάδρασης με τους εν δυνάμει πελάτες, όπως ορίζεται στα πλαίσια των εύελικτων μεθόδων. Αποτελεί τη δεύτερη επανάληψη αλληλεπίδραση με τους τελικούς χρήστες μέσα από την οποία επιχειρείται αξιο-

ποίηση της δυναμικής τους. Έχοντας θέσει σαν βασικό μας γνώμονα τις απόψεις των ερωτηθέντων κατά την πρώτη επανάληψη της έρευνάς μας σχεδιάσαμε και υλοποιήσαμε μια εφαρμογή η οποία θα ικανοποιεί ένα σημαντικό τμήμα των επιθυμιών τους. Η υλοποίηση αποτελεί ουσιαστικά μια πρώτη βασική έκδοση της εφαρμογής η οποία περιλαμβάνει τα βασικά βήματα προκειμένου να εκτελεστεί επιτυχώς μια ηλεκτρονική παραγγελία φαγητού. Ακολουθώντας τη λογική των ευέλικτων μεθοδολογιών ορίσαμε το χρονικό διάστημα μέσα στο οποίο θα έπρεπε να είναι έτοιμο ένα λειτουργικό τμήμα κώδικα να ανέρχεται στις τρεις εβδομάδες. Μέσα στο χρονικό διάστημα αυτό η πρώτη έκδοση ήταν έτοιμη να παρουσιαστεί στους πιθανούς χρήστες και να τεθεί υπό αξιολόγηση.

Στόχος της ενέργειας αυτής ήταν να υπάρξει μια πρώτη επαφή του τελικού χρήστη με την εφαρμογή προκειμένου να ελεγχθεί ο αντίκτυπος που θα έχει σε αυτούς και ταυτόχρονα να γίνει συλλογή των απόψεών τους. Η συλλογή αυτής της πληροφορίας θα μπορούσε, σε αυτό το σύντομο χρονικό διάστημα να δείξει εάν η προσπάθειά μας βαίνει σε καλό δρόμο ή όχι. Επιπροσθέτως, σε περίπτωση θετικής ανταπόκρισης από τους χρήστες θα μπορούσαμε μέσα από την αλληλεπίδραση με αυτούς να εντοπίσουμε τυχόν αδυναμίες ή ελλείψεις αλλά και πιθανές προτάσεις ώστε να συνεχίσουμε την υλοποίηση έχοντας ακόμα πιο ξεκάθαρο πλάνο.

Προκειμένου να τους φέρουμε σε επαφή με την εφαρμογή και να συλλέξουμε την επιθυμητή πληροφορία χρησιμοποιήσαμε εκ νέου τη μέθοδο συλλογής πληροφορίας μέσω ερωτηματολογίου. Συγκεκριμένα συντάξαμε ένα ερωτηματολόγιο στο πρώτο τμήμα του οποίου παρουσιάζαμε μέσω σχεδιασμού πρωτοτύπων ένα πλήρες σενάριο εκτέλεσης παραγγελίας μέσω της εφαρμογής. Στη συνέχεια το ερωτηματολόγιο αποτελείτο από δημογραφικές ερωτήσεις αλλά και ερωτήσεις σχετικά με την εφαρμογή. Οι ερωτήσεις ήταν πολλαπλής επιλογής γραμμικής κλίμακας αλλά και ανοιχτού τύπου. Στόχευαν στην διερεύνηση της αρχικής εντύπωσης που αφήνει η εφαρμογή, αν δηλαδή φάνταζε ελκυστική και πρωτότυπη, στην αξιολόγησή της από τους χρήστες σχετικά με ποικίλα χαρακτηριστικά όπως η ευκολία χρήσης και άλλα, αλλά και στην κατάθεση απόψεων από τους πιθανούς χρήστες σχετικά με ενδεχόμενες βελτιώσεις και μελλοντικές επεκτάσεις. Το ερωτηματολόγιο διενεμήθη πιλοτικά αρχικά σε ένα σύνολο 15 ατόμων προκειμένου να ελεγχθεί αν έχει συνταχθεί σωστά για τη συλλογή των απαιτούμενων αποτελεσμάτων. Μετά τις απαραίτητες προσαρμογές διετέθει στο δείγμα του οποίου οι απαντήσεις χρησιμοποιήθηκαν για τα τελικά αποτελέσματα.

Το σύνολο των απαντηθέντων ερωτηματολογίων ανήλθε στα εξήντα τρία (63). Εκ των συμμετεχόντων το 60.3% ήταν άντρες ενώ το 39.7% ήταν γυναίκες. Το μεγαλύτερο ποσοστό αυτών ήταν εργαζόμενοι κάτοικοι αστικών κέντρων και ένα μικρότερο ήταν φοιτητές. Το 44.4% ανήκουν στην ηλικιακή ομάδα 18-27, το 38.1% στην 28-37, το 15.9% στην ομάδα 38-47 και το υπόλοιπο είναι άνω των 48.

Μια σειρά ερωτημάτων γραμμικής κλίμακας είχαν σαν στόχο να παρατηρηθεί, εφόσον προηγουμένως οι ερωτηθέντες είχαν δει το σενάριο εκτέλεσης παραγγελίας φαγητού μέσω της εφαρμογής, σε τι επίπεδο η υλοποίηση μας κάλυπτε τα διάφορα χαρακτηριστικά που στην πρώτη επανάληψη της διάδρασης με το πιθανό κοινό είχαν απαιτηθεί να υπάρχουν σε μια εφαρμογή. Συγκεκριμένα ζητήθηκε ο βαθμός ικανοποίησής τους σχετικά με την ευκολία χρήσης, την ασφάλεια πληρωμών και προσωπικών δεδομένων, την επάρκεια παροχής πληροφοριών για το εκάστοτε γεύμα και της δυνατότητας αξιολόγησης του προϊόντος καθώς και της παρακολούθησης της παραγγελίας. Συνάμα ζητήθηκε να απαντήσουν κατά πόσο θα ήθελαν να χρησιμοποιήσουν την εφαρμογή. Οι ερωτήσεις ανοικτού τύπου ζητούσαν προτάσεις σχετικά με χαρακτηριστικά που δεν έχουν συμπεριληφθεί και θα ήθελαν να υπάρχουν καθώς και αξιολόγηση πιθανών ελαττωμάτων της εφαρμογής.

Παρακάτω παρουσιάζεται ένα πλήρες σενάριο παραγγελίας μέσω της εφαρμογής το οποίο κλήθηκαν οι ερωτηθέντες να μελετήσουν πριν προβούν στις απαντήσεις τους. Ακολουθούν τα αποτελέσματα των απαντήσεων που συλλέχθηκαν από το ερωτηματολόγιο που περιγράψαμε παραπάνω και που παρατίθεται στο τέλος αυτής της εργασίας.

6.4.2 Ελάχιστο Βιώσιμο Προϊόν (MVP) - Εφαρμογή 2EAT

MΕΣΑ από τις απαντήσεις που συλλέξαμε στην πρώτη επανάληψη της δεύτερης φάσης της έρευνας δημιουργήθηκε μια εικόνα του ποια χαρακτηριστικά πρέπει να συνδυάζει μια εφαρμογή ή πλατφόρμα ηλεκτρονικής παραγγελίας φαγητού προκειμένου να καλύπτει τις ανάγκες και τις επιθυμίες ενός τελικού χρήστη σε έναν ικανοποιητικό βαθμό. Αξιοποιώντας αυτή την πληροφορία φτάνουμε στην παρούσα φάση όπου παρουσιάζεται μια πρώτη έκδοση της εφαρμογής "2eat" που δημιουργήσαμε στηριζόμενα στις επιταγές του εν δυνάμει τελικού κοινού όπως αυτές παρουσιάστηκαν παραπάνω. Έτσι λοιπόν παρακάτω παρουσιάζεται ένα πλήρες σενάριο χρήσης της εφαρμογής αυτής τόσο από την πλευρά του τελικού χρήστη (B2C) που επιθυμεί να παραγγείλει φαγητό αλλά και εκείνου που επιθυμεί να το διαθέσει (B2B).

Στη συγκεκριμένη εφαρμογή γίνεται αξιοποίηση της διεπαφής χρήστη του κοινωνικού μέσου δικτύωσης Twitter. Έτσι η διαδικασία και για τα δύο μέρη που επιθυμούν να κάνουν χρήση της είναι, πέραν από απλουστευμένη, και εξαιρετικά οικεία καθώς ο χρήστης, πιθανότατα, ήδη είναι εξοικειωμένος με την συγκεκριμένη πλατφόρμα. Ο λόγος που επιλέχθηκε η "μεσολάβηση" του Twitter στην υλοποίηση της εφαρμογής ήταν προκειμένου να απαλλαχθούν οι τελικοί χρήστες από την χρονοβόρα κατά τα λεγόμενά τους διαδικασία εγγραφής αλλά και για να ενισχυθεί η αίσθηση ασφάλειας σε επίπεδο προσωπικών δεδομένων καθώς ο συγκεκριμένος ενδιαμέσος είναι πιστοποιημένα ασφαλής. Όπως προέκυψε από την πρώτη

επανάληψη της έρευνας, η απλότητα στη χρήση αλλά και η παροχή ασφάλειας σε επίπεδο προσωπικών δεδομένων αποτελούσαν δύο από τα σημαντικότερα χαρακτηριστικά που απαιτεί ένας χρήστης να του παρέχονται προκειμένου να χρησιμοποιήσει μια εφαρμογή ή μια πλατφόρμα ηλεκτρονικής παραγγελίας φαγητού. Με την προσέγγιση αυτή γίνεται προσπάθεια προκειμένου τα αιτήματά τους να ικανοποιηθούν και κατ' επέκταση να είναι θετικοί στη χρήση της παρούσας εφαρμογής.

6.4.3 B2B Πελάτες

Η διαδικασία προκειμένου ένα εστιατόριο να διαθέσει μέσω της εφαρμογής το γεύμα που επιθυμεί είναι απλή και απαιτεί μόνο ένα βήμα. Χρειάζεται μόνο μια ανάρτηση (tweet) η οποία θα εμπεριέχει την ονομασία του γεύματος, την τιμή του καθώς και μια εικόνα αυτού. Η ανάρτηση θα πρέπει να συνοδεύεται από το hash-tag #2eat προκειμένου να είναι ανιχνεύσιμη κατά την αναζήτηση των πελατών [Σχήμα 6.7]. Με τον τρόπο αυτό ο τελικός πελάτης που θα επιλέξει το συγκεκριμένο γεύμα έχει την απαραίτητη αρχικά πληροφόρηση σχετικά με το προϊόν όπως και μια εικόνα αυτού.



Σχήμα 6.7: Το εστιατόριο κάνει tweet το γεύμα.

Η εφαρμογή ανιχνεύει το #2eat hash-tag και παράγει αυτόματη απάντηση στην ανάρτηση, η οποία περιγράφει τα θρεπτικά συστατικά του συγκεκριμένου γεύματος [Σχήμα 6.8]. Η πληροφορία αυτή αναχτάνται από την ανοιχτή βάση δεδομένων του Calorie King μέσω αναζήτησης ελεύθερου κειμένου. Έτσι η πληροφορία για το γεύμα εμπλουτίζεται και συνεπώς ικανοποιείται η επιθυμία του πελάτη για επαρκή πληροφόρηση σχετικά με το γεύμα που θα καταναλώσει. Μετά την ολοκλήρωση αυτής της μονοβηματικής ενέργειας το σύστημα είναι πλέον έτοιμο να δεχτεί παραγγελίες.

Η παραπάνω διαδικασία δίνει τη δυνατότητα στο εκάστοτε εστιατόριο να ξεφύγει από τη λογική του πλήρους καταλόγου ο οποίος πιθανότατα δεν θα είναι προσφάτως ενημερωμένος, όπως βλέπουμε σε υπάρχουσες εφαρμογές. Μέσω της εφαρμογής μπορεί να διαθέσει τα γεύματα

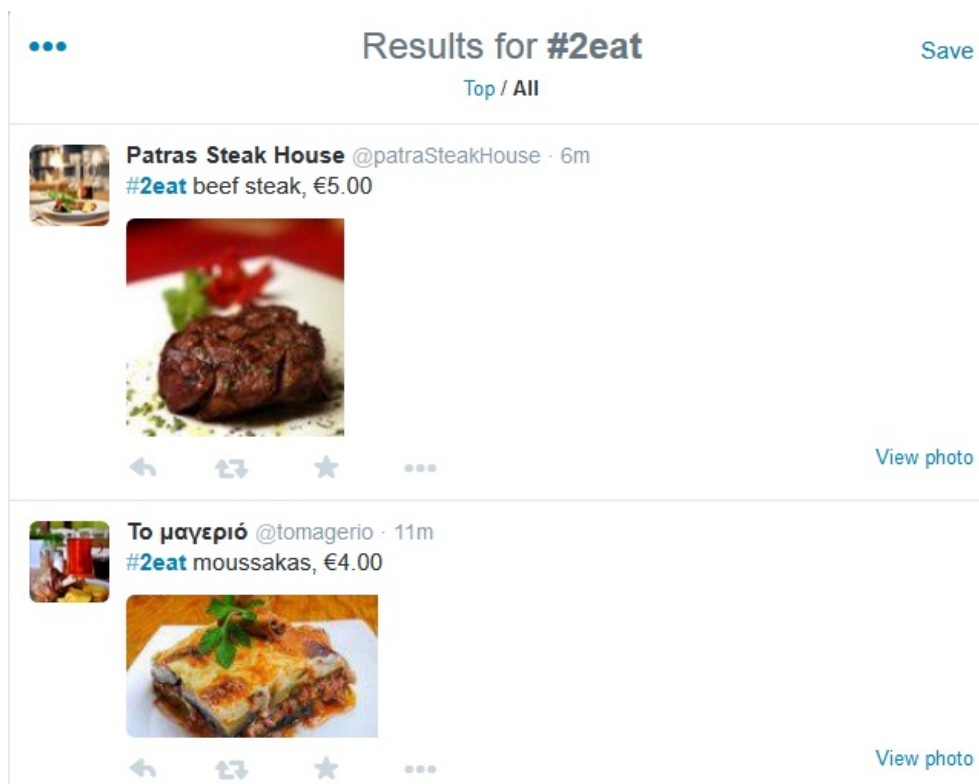
που έχει διαθέσιμα την τρέχουσα στιγμή ενώ ταυτόχρονα του δίνεται και η δυνατότητα να αφαιρέσει με μια απλή διαγραφή όποια από αυτά έχουν εξαντληθεί. Συνεπώς έχει την ευελιξία να κάνει διαχείριση των προϊόντων του βάσει της διαθεσιμότητας τους άμεσα, χωρίς απαίτηση να μεσολαβήσει κάποιος προκειμένου να την ενημερώσει. Κατ' επέκταση αποφεύγει περιπτώσεις όπως το να χρειαστεί να καλέσει κάποιον πελάτη ο οποίος παρήγγειλε κάποιο γεύμα που πλέον δεν είναι διαθέσιμο και να τον ενημερώσει σχετικά. Επιπροσθέτως μπορεί να προωθήσει και τα προϊόντα εκείνα που έχει σε αφθονία την συγκεκριμένη ημέρα και σε γενικές γραμμές να αποφασίζει και να λειτουργεί ανεξάρτητα χωρίς να χρειάζεται τις υπηρεσίες κάποιου τρίτου που θα είναι γνώστης των απαραίτητων τεχνολογιών προκειμένου να προβεί στις απαραίτητες αλλαγές.



Σχήμα 6.8: Αυτόματη απάντηση με τα διατροφικά χαρακτηριστικά του γεύματος που προκύπτουν από αναζήτηση με *free-text* αναζήτηση.

6.4.4 B2C Πελάτες

Όντας το σύστημα έτοιμο πλέον να δεχτεί παραγγελίες οι πελάτες μπορούν να δουν όλα τα διαθέσιμα γεύματα από όλα τα εστιατόρια κάνοντας μια απλή αναζήτηση στο Twitter για το #2eat hash-tag [Σχήμα 6.9]. Αυτομάτως εμφανίζονται όλες οι διαθέσιμες επιλογές. Έτσι η αναζήτηση είναι πιο εύκολη καθώς δεν απαιτείται απ' αυτούς να ανατρέξουν σε καταλόγους ενός μόνο συγκεκριμένου εστιατορίου. Έχουν μια πλήρη εικόνα του τι είναι διαθέσιμο από ποικίλους παρόχους κάνοντας μόνο μια απλή αναζήτηση. Με τον τρόπο αυτό καλύπτεται και η απαίτησή τους για ποικιλία προϊόντων και εστιατορίων.



Σχήμα 6.9: Αποτελέσματα αναζήτησης για το 2eat.

Όταν πλέον έχει επιλέξει ποιο είναι το γεύμα που θέλει να καταναλώσει μπορεί να προβεί στην παραγγελία αυτού απλά απαντώντας στην ανάρτηση που το αφορά. Η απάντηση αυτή θα πρέπει να περιλαμβάνει τα #2eat #order hash-tags ακολουθούμενα από το όνομα του γεύματος και την ποσότητα που επιθυμεί να παραγγείλει [Σχήμα 6.10] .



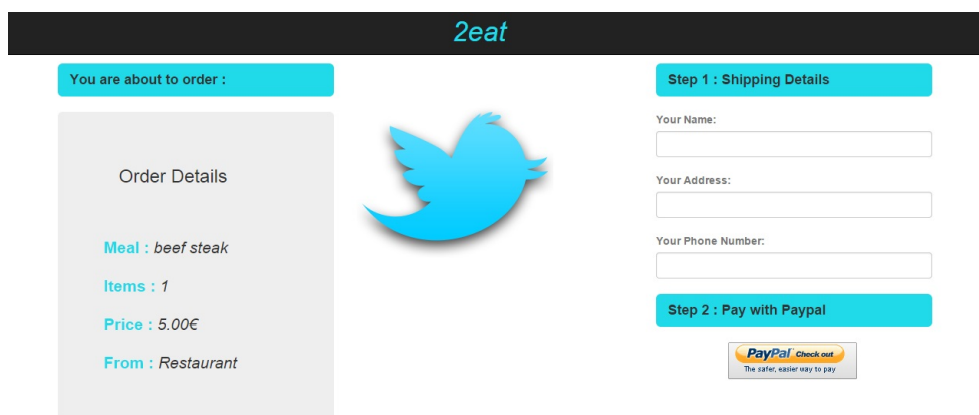
Σχήμα 6.10: Ο πελάτης τοποθετεί παραγγελία με ένα απλό tweet.

Όταν σταλεί η απάντηση η εφαρμογή αυτομάτως ανιχνεύει #2eat #order hash-tags, και όταν η παραγγελία αποσταλεί απαντάει με ένα προσωπικό σύνδεσμο που παροτρύνει τον πελάτη να τον ακολουθήσει προκειμένου να συμπληρώσει στοιχεία διανομής και να πραγματοποιήσει την πληρωμή προκειμένου να ολοκληρωθεί η παραγγελία του [Σχήμα 6.11].



Σχήμα 6.11: Αυτόματο tweet στον πελάτη που αναγράφει το σύνδεσμο που ακολουθεί για να πληρώσει.

Όπως φαίνεται και στο Σχήμα 6.12 η προσωπική σελίδα στην οποία κατευθύνεται ο χρήστης μόλις ακολουθήσει το σύνδεσμο, περιλαμβάνονται λεπτομέρειες σχετικά με την παραγγελία ενώ ταυτόχρονα παρουσιάζονται και τα δύο βήματα που πρέπει να κάνει ο προκειμένου η παραγγελία να ολοκληρωθεί. Στο πρώτο βήμα καλείται να συμπληρώσει τα απολύτως απαραίτητα στοιχεία για τη διανομή του γεύματος και στο δεύτερο βήμα θα πρέπει να εξοφλήσει την παραγγελία του χρησιμοποιώντας το PayPal.



The screenshot displays the 2eat website interface. At the top, the 2eat logo is visible. Below it, the page is divided into three main sections. On the left, under the heading 'You are about to order:', there is a box titled 'Order Details' containing the following information: 'Meal : beef steak', 'Items : 1', 'Price : 5.00€', and 'From : Restaurant'. In the center, there is a large blue Twitter bird logo. On the right, there are two steps for completing the order. 'Step 1 : Shipping Details' includes input fields for 'Your Name:', 'Your Address:', and 'Your Phone Number:'. Below this, 'Step 2 : Pay with Paypal' features a 'PayPal Check out' button with the tagline 'The safer, easier way to pay'.

Σχήμα 6.12: Ο χρήστης παρέχει πληροφορίες παράδοσης και καλείται να πληρώσει.

Όταν η διαδικασία της αποπληρωμής ολοκληρωθεί τότε ο χρήστης λαμβάνει απάντηση από το σύστημα που του επιβεβαιώνει την παραγγελία του και τον ενημερώνει σε πόση ώρα το φαγητό του θα βρίσκεται στην πόρτα του. Επιπροσθέτως του παρέχει και πληροφορίες σχετικά με τη θερμιδική αξία του γεύματος που θα καταναλώσει και ποιά φυσική δραστηριότητα θα μπορούσε να ακολουθήσει προκειμένου να τις κάψει.

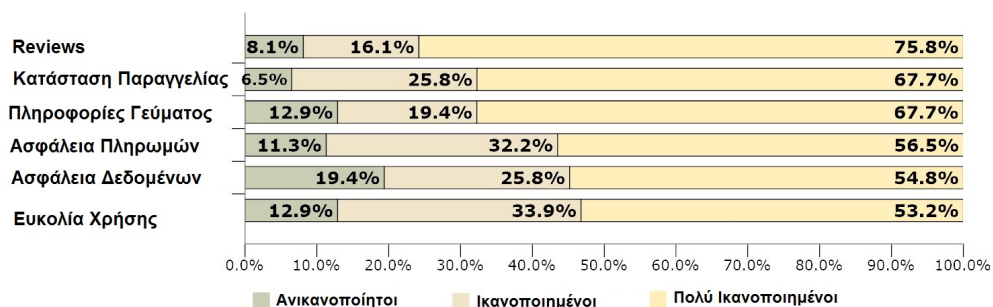


Σχήμα 6.13: Αυτόματη επιβεβαίωση παραγγελίας περιλαμβάνοντας χρόνο παράδοσης. Σχόλιο πελάτη.

Ο χρήστης δύναται, αν το επιθυμεί, όταν παραλάβει την παραγγελία του, να εκφράσει την άποψή του σχετικά με το γεύμα ή το εστιατόριο δίνοντας έτσι και στους υπόλοιπους χρήστες τη δυνατότητα να επωφεληθούν από αυτήν και να πράξουν αναλόγως [Σχήμα 6.13]. Με τον τρόπο αυτό υλοποιείται μια μέθοδος αξιολόγησης του εκάστοτε εστιατορίου από τους πελάτες που το προτίμησαν.

6.4.5 Αποτελέσματα Δεύτερης Διάδρασης

Εχοντας συλλέξει τις απαντήσεις του δείγματος τα αποτελέσματα που προέκυψαν σχετικά με τη λειτουργικότητα και την πρωτοτυπία της εφαρμογής ήταν ενθαρρυντικά. Στο [Σχήμα 6.13] παρουσιάζεται συγκεντρωτικά τα επίπεδα ικανοποίησης των συμμετεχόντων όσων αφορά τον τρόπο απόδοσης των διαφόρων χαρακτηριστικών που παρουσιάζονται.



Σχήμα 6.14: Αξιολόγηση χαρακτηριστικών Zeat σε σχέση με τα επίπεδα ικανοποίησης των πελατών.

Όπως προκύπτει από τις απαντήσεις όλα τα διαθέσιμα χαρακτηριστικά φαίνεται να ικανοποιούν ποσοστό πολύ μεγαλύτερο του 50%. Συγκεκριμένα ο τρόπος που υλοποιούνται, η δυνατότητα αξιολόγησης του προϊόντος, η κατάσταση της παραγγελίας και η πληροφόρηση επαρκεί σε μεγαλύτερο ή μικρότερο ποσοστό για την πλειοψηφία των ερωτηθέντων. Αρκετά έως πολύ ικανοποιημένοι δηλώνουν σχετικά με τον τρόπο που πραγματώνονται η ασφάλεια οικονομικών συναλλαγών καθώς και η ευκολία χρήσης ενώ τελευταία θέση καταλαμβάνει η ασφάλεια των προσωπικών δεδομένων. Στις ερωτήσεις ανοιχτού τύπου που τέθηκαν δίνεται η απάντηση στο γιατί ο τρόπος υλοποίησης του χαρακτηριστικού αυτού τους ανησυχεί. Εξετάζοντας με μια πιο γενική ματιά το παραπάνω διάγραμμα συμπεραίνουμε πως οι πιθανοί χρήστες δείχνουν να είναι πιο αυστηροί και πιο ανήσυχοι στα ζητήματα ασφάλειας συναλλαγών και προσωπικών δεδομένων, πράγμα που δεν εκπλήσσει καθώς αυτά είναι ζητήματα που τέθηκαν στην κορυφή των απαιτήσεων τους.

Σε γενικές γραμμές η πρώτη εκδοχή της εφαρμογής είχε θετική απήχηση. Η μεγαλύτερη μερίδα δήλωσε ότι φαίνεται ενδιαφέρουσα και πρωτότυπη. Στην ερώτηση εάν θα χρησιμοποιούσαν την εφαρμογή αυτή για να πραγματοποιήσουν μια παραγγελία το 55.5% δήλωσε πως θα την χρησιμοποιούσε σίγουρα, το 36.5% ότι πιθανότατα θα έκανε χρήση της ενώ το 7.9% δήλωσε πως δεν θα τη χρησιμοποιούσε.

Στις ερωτήσεις ανοιχτού τύπου σχετικά με το ποιά χαρακτηριστικά θα επιθυμούσαν να συμπεριλαμβάνονται στην εφαρμογή τα οποία δεν έχουν ήδη υλοποιηθεί και τι εντοπίζουν ως αδυναμία της υλοποίησης οι απαντήσεις τους συνοψίζονται ως εξής:

1. Πολλοί εκ των ερωτηθέντων ανέφεραν ότι θα επιθυμούσαν να υπάρχει δυνατότητα αποπληρωμής με αντικαταβολή. Είναι, σύμφωνα με τις απαντήσεις τους, πιο κοντά στην ελληνική λογική και πιθανότατα η μη ύπαρξη αυτής της επιλογής να αποκλείσει κάποιους από τη χρήση της εφαρμογής.

2. Λόγω του γεγονότος ότι η διαδικασία της παραγγελίας γίνεται μέσω της πλατφόρμας Twitter θεωρούν ότι είναι σε κοινή θέα οι παραγγελίες τους. Αυτό τους δημιουργεί μια ανασφάλεια και έτσι δικαιολογείται το γεγονός ότι δεν τους ικανοποιεί σε επίπεδο προστασίας των προσωπικών δεδομένων.
3. Θα ήταν ενδιαφέρον να επεκταθεί και σε άλλη πλατφόρμα κοινωνικής δικτύωσης, όπως παραδείγματος χάριν το Facebook, με το οποίο οι Έλληνες χρήστες είναι πιο εξοικειωμένοι.
4. Διαφέρει πολύ από την προϋπάρχουσα φιλοσοφία ηλεκτρονικής παραγγελίας φαγητού συνεπώς θα χρειαστεί χρόνος για την εξοικείωση.
5. Προτείνουν να συμπεριληφθεί η δυνατότητα να μπορεί να γίνει αναζήτηση βάσει επιπλέον επιλογών όπως το όνομα του εστιατορίου, το είδος του φαγητού, η περιοχή.

Οι απαντήσεις που συλλέχθηκαν σε αυτή τη φάση της αλληλεπίδρασης με τους εν δυνάμει χρήστες αποτελούν ουσιαστικά μια ανατροφοδότηση η οποία μπορεί να αξιοποιηθεί σε επόμενες εκδόσεις της εφαρμογής εφόσον φάνηκε η εφαρμογή να χαίρει θετικής αντίδρασης από την πλειοψηφία των ερωτηθέντων.

6.5 Τεχνολογίες που χρησιμοποιήθηκαν

Για την πραγμάτωση της εφαρμογής 2eat αξιοποιήθηκε το κοινωνικό δίκτυο Twitter το οποίο κάνει διαθέσιμα τα δεδομένα του σε μορφή API. Η εφαρμογή υλοποιήθηκε με τη χρήση της Spring Boot, μηχανής προτύπων Thymeleaf καθώς και της Java βιβλιοθήκης Twitter4J. Επίσης χρησιμοποιήθηκαν οι Java εκδόσεις των ExpressCheckout PayPal API και CalorieKing Food API.

Twitter API: Το API του Twitter αποτελείται από τρία μέρη: δύο REST APIs και το Streaming API. Αυτό που χρησιμοποιούμε στα πλαίσια της παρούσας εργασίας για τη συλλογή των μηνυμάτων από τους χρήστες (tweets) είναι το Streaming API το οποίο επιτρέπει σχεδόν σε πραγματικό χρόνο πρόσβαση σε διάφορα υποσύνολα των ενημερώσεων των χρηστών (tweets). Προκειμένου να υπάρξει επικοινωνία μεταξύ της εφαρμογής μας και του Twitter API χρησιμοποιείται η βιβλιοθήκη Twitter4j. Παρέχει κλάσεις (StreamListener, FilterQuery) και μεθόδους για την ανάκτηση και το φιλτράρισμα των tweets καθώς και απάντηση σε αυτά.

CalorieKing Food API: Αναπτύχθηκε από την CalorieKingDeveloper. Το API παρέχει πρόσβαση σε μία εκτενή βάση δεδομένων, στην οποία περιέχονται, κατηγοριοποιημένα, τα θρεπτικά και συστατικά στοιχεία πολλών διαφορετικών γευμάτων. Τα γεύματα είναι κατηγοριοποιημένα ανάλογα με τα συστατικά, τον τρόπο παρασκευής και διάθεσής τους. Τέλος, τα γεύματα και τα συστατικά που διατίθενται δεν περιορίζονται σε ένα εί-

δος κουζίνας, αλλά λαμβάνουν υπόψη τις γαστρονομικές συνήθειες πολλών λαών ανά τον κόσμο.

Οι υπόλοιπες τεχνολογίες έχουν ήδη παρουσιαστεί στο τέταρτο κεφάλαιο.

6.6 Συμπεράσματα - Συζήτηση

Στο παρόν κεφάλαιο πραγματοποιήθηκε ανάπτυξη web 2.0 εφαρμογής η οποία εξυπηρετεί την ηλεκτρονική παραγγελία φαγητού. Η ανάπτυξη αυτής βασίστηκε στις αρχές των ευέλικτων μεθόδων ανάπτυξης λογισμικού οι οποίες δίνουν έμφαση στην πελατοκεντρικότητα. Κυρίως στόχος του είναι η ικανοποίηση του τελικού πελάτη και η δημιουργία όσο το δυνατόν καλύτερης εμπειρίας χρήστη. Αυτό επιτυγχάνεται με τη συνεχή επαναληπτική συμμετοχή του χρήστη σε όλα τα στάδια ανάπτυξης του λογισμικού. Αυτό επιχειρήθηκε και στην ανάπτυξη της εφαρμογής 2eat. Ακολουθώντας τις αρχές των ευέλικτων μεθόδων όπως αυτές παρουσιάστηκαν σε προηγούμενο κεφάλαιο πραγματοποιήθηκε γρήγορη ανάπτυξη του ελάχιστου βιώσιμου προϊόντος μέσα από επαναληπτική διάδραση με πιθανούς τελικούς χρήστες.

Παρατηρήθηκε πως η προσέγγιση αυτή, φέρνοντας δηλαδή τους τελικούς χρήστες και τις ανάγκες στους στην πρώτη γραμμή μπορεί να οδηγήσει σε ανάπτυξη εφαρμογών που γρήγορα μπορούν να προσελκύσουν το ενδιαφέρον τους και να επιτύχει την επιθυμητή ικανοποίησή τους σε πρώιμα στάδια του προϊόντος. Οι επαναληπτική διάδραση μέσω συχνών παραδόσεων εκδόσεων λειτουργικού προϊόντος επιτρέπει την έγκαιρη ανακάλυψη πιθανών σφαλμάτων, τον περιορισμό υλοποίησης χαρακτηριστικών που δεν αποδίδουν ουσιαστική αξία βάσει των αναγκών του πελάτη αλλά και την πολύ γρήγορη αποκάλυψη μιας πιθανής αποτυχίας.

Η συνεχής συμμετοχή των πελατών σε αυτή την εφαρμογή αποκάλυψε τις ανάγκες και τις επιθυμίες τους ενώ η ταχεία παράδοση μια πρώτης έκδοσης αυτής οδήγησε στον άμεσο εντοπισμό των αδυναμιών αλλά και στην επιβεβαίωση των δυνατών της χαρακτηριστικών. Η συγκεκριμένη εφαρμογή έτυχε θετικής αποδοχής από τους εν δυνάμει χρήστες και χαρακτηρίστηκε ως πρωτότυπη με δυνατότητα επέκτασης σε άλλες πλατφόρμες κοινωνικής δικτύωσης, ενώ ταυτόχρονα φαίνεται να καλύπτει μια διαφορετική λογική ηλεκτρονικής παραγγελίας φαγητού. Η συλλογή και η αξιοποίηση της ανατροφοδότηση από τους χρήστες παρατηρήθηκε ότι οδηγεί σε περιορισμό της σπατάλης χρόνου και κόπου και στην παράδοση πιο συγχροτημένων web 2.0 εφαρμογών οι οποίες στοχεύουν κατευθείαν στην εξυπηρέτηση των ζητούμενων από τους πελάτες χωρίς να πλατειάζουν με παροχή αχρείαστων δυνατοτήτων. Μέσα από την προσπάθεια αυτή παρατηρήθηκε πως η εφαρμογή της πελατοκεντρικής προσέγγισης μπορεί να διευκολυνθεί με την αξιοποίηση τεχνολογιών που μπορούν να διευκο-

λύνουν την ανάπτυξη ενός βιώσιμου προϊόντος το οποίο θα διανεμηθεί πολύ γρήγορα σε πιθανούς πελάτες. Στην περίπτωση της 2eat εφαρμογής η χρήση διαφόρων web 2.0 APIs όπως το Twitter streaming API επέτρεψε την γρήγορη και αποτελεσματική πρωτοτυποποίηση.

Τέλος μέσα από αυτή τη διαδικασία ανάπτυξης με κέντρο τον πελάτη προέκυψαν σημαντικές παρατηρήσεις σχετικά με την σχέση των Ελλήνων με την ηλεκτρονική παραγγελία φαγητού. Όπως είδαμε, οι Έλληνες αν και αντιλαμβάνονται τη χρησιμότητα και την αξία των εφαρμογών ηλεκτρονικής παραγγελίας φαγητού χρειάζονται ακόμα χρόνο προκειμένου να απομακρύνουν από τη λογική και την καθημερινότητα τους την τοποθέτηση παραγγελιών μέσω τηλεφώνου. Η παροχή ασφάλειας σε επίπεδο προσωπικών δεδομένων και πληρωμών, η ευκολία χρήσης, η απλότητα και η επαρκής πληροφόρηση για τα προϊόντα φαγητού που επιθυμούν να παραγγείλουν αποτελούν τις βασικές τους απαιτήσεις προκειμένου να εγκαταλείψουν τους παραδοσιακούς τρόπους παραγγελίας φαγητού και να στραφούν προς την ηλεκτρονική μορφή της. Λαμβάνοντας υπόψη αυτές τις επιθυμίες και τις προτάσεις τους οι μελλοντικές εφαρμογές παραγγελίας φαγητού θα καταφέρουν να κερδίζουν έδαφος έναντι των υπαρχόντων παραδοσιακών μεθόδων.

Μέρος IV

ΣΥΜΠΕΡΑΣΜΑΤΑ

ΣΥΜΠΕΡΑΣΜΑΤΑ

7.1 Συμπεράσματα

Στην παρούσα εργασία παρατηρήθηκε ότι η ανάπτυξη web 2.0 εφαρμογών ακολουθώντας τις αρχές των παραδοσιακών μεθόδων δεν αποτελεί ιδιαιτέρως αποδοτική προσέγγιση. Η συμμετοχή των τελικών πιθανών χρηστών μόνο κατά τα αρχικά στάδια σχεδιασμού και στο τέλος όπου γίνεται η αξιολόγηση αυτής μπορεί να ενέχει πολλούς κινδύνους στην ανάπτυξη εφαρμογών που απευθύνονται σε ευρύ φάσμα χρηστών και όχι σε μεμονωμένο πελάτη. Οι σύγχρονοι ρυθμοί και οι συνεχείς μεταβολές των αναγκών και των επιθυμιών των τελικών χρηστών καθώς και ο συνεχής ανταγωνισμός επιβάλλουν συνεχή συμμετοχή του χρήστη ήδη από τα πρώιμα στάδια σύλληψης μια ιδέας έως και τη στιγμή της τελικής της παράδοσης. Οι ευέλικτες μέθοδοι που βρίσκονται στον αντίποδα αντιλαμβάνονται τη σημασία της πελατοκεντρικότητας, και πως η σωστή διαχείριση της πληροφορίας μπορεί να ξεκαθαρίσει από νωρίς τη δυναμική μιας ιδέας, να βοηθήσει στην ανάπτυξη πελατών πριν καν την κυκλοφορία της εκάστοτε εφαρμογής και να εξασφαλίσει ότι θα καλυφθούν οι απαιτήσεις των τελικών χρηστών. Όλα αυτά επιτυγχάνονται μέσα σε ένα πλαίσιο αντίληψης των ευμετάβλητων αναγκών και την ανάγκη για ταχεία ανάπτυξη και παράδοση λειτουργικού λογισμικού στο οποίο αρχικά θα καλύπτει βασικές απαιτήσεις των χρηστών.

Όπως είδαμε στο τέταρτο κεφάλαιο, η εφαρμογή EAT-IN η οποία σχεδιάστηκε με αρχές των παραδοσιακών μεθόδων, παρουσίασε προβλήματα τα οποία οδήγησαν σε δυσάρεσκη για κάποια χαρακτηριστικά ενώ

αποκαλύφθηκε μετά την πλήρη ολοκλήρωσή της πως σημαντικό τμήμα της δουλείας που είχε γίνει θα έμενε στην αφάνεια καθώς οι χρήστες δεν το εκτίμησαν όσο αρχικά θεωρήθηκε. Επίσης η βραδύτητα παράδοσης της εφαρμογής έδωσε την ευκαιρία στους χρήστες να βρουν άλλη λύση για το πρόβλημά τους. Αντίθετα η εφαρμογή Zeat με την επαναληπτική διαδικασία αναζήτησης και αξιοποίησης των τελικών χρηστών, με τη συνεχή εισαγωγή της άποψης και των αναγκών τους και με τη γρήγορη παράδοση μιας πρώτης απλής λειτουργικής της έκδοσης (MVP) έκανε το ενδιαφέρον των τελικών χρηστών να είναι πιο έντονο. Ταυτόχρονα η όποια αλλαγή στις απαιτήσεις τους μπορούσε γρήγορα να ενσωματωθεί δίνοντας στην εφαρμογή το χαρακτηριστικό της προσαρμοστικότητας και συνεπώς και της επικαιρότητας.

Συγκρίνοντας τη διαδικασία σχεδιασμού και υλοποίησης των δύο εφαρμογών παρατηρήθηκε πως η EAT-IN κόστισε παραπάνω χρονικά και από άποψη κόπου καθώς τμήματα αυτής δεν ακολουθούσαν εν τέλει τις επιθυμίες των πελατών-χρηστών τη συγκεκριμένη στιγμή. Αυτό μεταφράστηκε σε περίπου δυο χιλιάδες γραμμές κώδικα που έπρεπε μέρος του να τροποποιηθεί και κάποιο άλλο να αφαιρεθεί. Όμως η αλλαγή, παρά το γεγονός ότι κάναμε χρήση του ισχυρού Spring Framework, δεν ήταν τόσο εύκολη καθώς όριζε σημαντική αλλαγή στη συνολική δόμηση της εφαρμογής. Αντίθετα με την Zeat αυτά τα προβλήματα προλαμβάνονται και η σπατάλη χρόνου και κόπου περιορίζεται καθώς οι συνεχείς μικρές διανομές και η ανατροφοδότηση κάθε φορά από τους χρήστες επιτρέπει να αποβάλλονται γρήγορα τα περιττά και να γίνεται ο κώδικας πιο απλός και διαχειρίσιμος. Στην ευμεταβλητότητα που χαρακτηρίζει το κοινό των web 2.0 εφαρμογών οι ευέλικτες μέθοδοι ανταποκρίνονται καλύτερα καθώς καλοδέχονται τις αλλαγές προς ικανοποίηση του πελάτη ακόμα και σε προχωρημένο στάδιο δίνουν έμφαση στην πεποίθηση ότι κάθε επανάληψη προσθέτει αξία στο συνεχώς αναπτυσσόμενο προϊόν και στο ότι η απόφαση για το αν η αξία έχει προστεθεί λαμβάνεται από τους τελικούς χρήστες και τους πελάτες.

Από την παρούσα εργασία λοιπόν συμπεραίνουμε αρχικά πως η υιοθέτηση κάποιας μεθόδου παραγωγής λογισμικού για την ανάπτυξη web 2.0 εφαρμογών είναι σημαντική και πως ad-hoc διαδικασίες πιθανότατα να μην οδηγήσουν σε βιώσιμο προϊόν. Ακόμα πιο σημαντικό όμως είναι η επιλογή της σωστής μεθόδου. Οι ευέλικτες μέθοδοι σε συνδυασμό με έμφαση στην πελατοκεντρικότητα αποδείχθηκαν πιο αποδοτικές για την ανάπτυξη τέτοιων εφαρμογών. Η εισαγωγή των τελικών χρηστών επαναληπτικά σε όλα τα στάδια ανάπτυξης της εφαρμογής Zeat έδειξε πως μέσω αυτής της προσέγγισης μπορεί να αυξηθεί η ικανοποίηση των τελικών χρηστών ενώ ταυτόχρονα μπορούν να δοθούν ευκαιρίες ανάπτυξης οι οποίες μπορεί να πηγάζουν από ανεκπλήρωτες μέχρι τώρα επιθυμίες τους. Τέλος η συμβουλή των ευέλικτων μεθόδων για συνεχή αξιοποίηση της υπάρχουσας γνώσεως και των υπαρχόντων τεχνολογιών μπορεί να οδηγή-

σει σε ιδέες πρωτότυπες η οποίες σε σύντομο χρονικό διάστημα μπορούν να αξιολογηθούν.

Αποδεικνύεται λοιπόν έμπρακτα πως ο συνδυασμός αρχών των μεθόδων Agile, Lean και Customer Development υπερέχουν έναντι των παραδοσιακών μεθόδων όταν πρόκειται για ανάπτυξη web 2.0 λογισμικού. Η ανάπτυξη λογισμικού βασισμένη σε αρχές που προκύπτει απο το συνδυασμό τους πράγματι οδηγεί στην ανάπτυξη web 2.0 εφαρμογών που απαιτούν λιγότερο χρόνο για να υλοποιηθούν, μικρότερο κόστος, να είναι περισσότερο πελατοκεντρικές και να εξασφαλίζουν μεγαλύτερο βαθμό ικανοποίησης των τελικών χρηστών.

Μέρος V

ΠΑΡΑΡΤΗΜΑ

ΚΕΦΑΛΑΙΟ

8

ΠΑΡΑΡΤΗΜΑ

8.1 Ερωτηματολόγια

Στη συνέχεια παρατίθενται τα ερωτήματα που τέθηκαν κατά τη διεξαγωγή προσωπικών συνεντεύξεων για την εξαγωγή προδιαγραφών στην περίπτωση της πλατφόρμας ηλεκτρονικής παραγγελίας φαγητού EAT-IN, καθώς και τα ερωτηματολόγια που χρησιμοποιήθηκαν για τη μέτρηση της ικανοποίησης των πελατών στην περίπτωση της εφαρμογής ηλεκτρονικής παραγγελίας φαγητού 2EAT.

8.1.1 Ερωτήσεις συνεντεύξεων (EAT-IN)

Ερώτηση 1: Στο εργασιακό σας περιβάλλον υπάρχει η δυνατότητα παροχής σίτισης;

Ερώτηση 2: Έχετε τη δυνατότητα διαλείμματος για φαγητό; Εάν ναι, πόσο διαρκεί; Είναι προκαθορισμένος ο χρόνος και η ώρα του διαλείμματός σας;

Ερώτηση 3: Πόσο συχνά παραγγέλνετε φαγητό στον εργασιακό σας χώρο;

Ερώτηση 4: Θεωρείτε την επιλογή φαγητού μια χρονοβόρα διαδικασία;

Ερώτηση 5: Με ποιους τρόπους παραγγέλνετε φαγητό στον εργασιακό σας χώρο;

Ερώτηση 6: Πόσο συχνά απευθύνεστε σε ηλεκτρονικές εφαρμογές για της παραγγελία του γεύματός σας;

Ερώτηση 7: Ποια θεωρείται πως είναι τα μειονεκτήματα των ηλεκτρονικών εφαρμογών παραγγελίας φαγητού;

Ερώτηση 8: Πόσο υγιεινά τρέφεστε;

Ερώτηση 9: Θεωρείτε πως το να σας προτείνει μια εφαρμογή κάποια γεύματα ανά ημέρα βάση των προτιμήσεών σας θα σας εξοικονομούσε χρόνο;

Ερώτηση 10: Θα σας ενδιέφερε ως χαρακτηριστικό μιας εφαρμογής το να σας παρέχει τη δυνατότητα εμφάνισης πλήρους εβδομαδιαίου μενού βάσει των προτιμήσεών σας;

Ερώτηση 11: Ποια χαρακτηριστικά θα θέλατε να έχει μία εφαρμογή ηλεκτρονικής παραγγελίας φαγητού;

Ερώτηση 12: Ποια είναι η ηλικία σας;

Ερώτηση 13: Πόσα χρόνια εργάζεστε στην επιχείρηση;

Ερώτηση 14: Ποια είναι η οικογενειακή σας κατάσταση;

8.1.2 Ερωτηματολόγια (2EAT) - Πρώτη Διάδραση

Η εταιρεία που εργάζεστε διαθέτει εγκαταστάσεις σίτισης;
Αν η απάντησή σας είναι "Όχι" προχωρήστε στην ερώτηση 8.

- ☐ Ναι
- ☐ Εν μέρει
- ☐ Όχι

Οι εγκαταστάσεις σίτισης που διαθέτει η εταιρεία σας είναι...

- ☐ Αυτόματος πωλητής με συσκευασμένα προϊόντα (πχ. κρύα σάντουιτς, τοστ κλπ.)
- ☐ Κυλικείο
- ☐ Εστιατόριο εντός της εταιρείας
- ☐ Άλλο...

Η εταιρεία σας σάς καλύπτει έξοδα σίτισης στα πλαίσια της απασχόλησής σας;

- ☐ Καθόλου
- ☐ Μικρό μέρος αυτών

☐ Μεγάλο μέρος αυτών

☐ Τα καλύπτει πλήρως

Έχετε δυνατότητα διαλείμματος για φαγητό στη δουλειά σας;

☐ Ναι, πάντα

☐ Ναι, κάποιες φορές

☐ Όχι, ποτέ

Είναι σημαντικό για εσάς να τρώτε κατά τη διάρκεια της δουλειάς σας;

Επιλέξτε έναν αριθμό από το 1 έως το 5 βάσει σημαντικότητας για εσάς.

1	2	3	4	5
---	---	---	---	---

Υπάρχει κάποιο εστιατόριο που να παρέχει καλύτερες τιμές στους εργαζόμενους της εταιρείας που δουλεύετε;

☐ Ναι, πάνω από ένα

☐ Ναι, υπάρχει ένα

☐ Όχι

☐ Δεν γνωρίζω

Πόσο συχνά μαγειρεύετε οι ίδιοι (ή κάποιος δικός σας) το φαγητό που τρώτε στην δουλειά;

Επιλέξτε τον αριθμό από το 1 έως το 5 που ανταποκρίνεται στη συχνότητα που μαγειρεύετε οι ίδιοι το φαγητό σας.

1	2	3	4	5
---	---	---	---	---

Το να μαγειρεύετε οι ίδιοι το φαγητό που τρώτε στη δουλειά σας το θεωρείτε...

Βαθμολογείστε το κάθε χαρακτηριστικό με το 1 να δηλώνει το χαμηλότερο βαθμό και το 3 τον υψηλότερο.

	1	2	3
Οικονομικό			
Υγιεινό			
Κουραστικό			
Χρονοβόρο			
Επιπλέον καθημερινό άγχος			

Πόσο συχνά το γεύμα που τρώτε στη δουλειά σας είναι κάποιο σνακ (πχ. κρύο σάντουιτς, τοστ κλπ.);

Επιλέξτε εάν αριθμό από το 1 έως το 5 αναλόγως τη συχνότητα.

1	2	3	4	5
---	---	---	---	---

Πόσο σας ικανοποιεί το γεύμα σας στη δουλειά να αποτελείται από σνακ;

Επιλέξτε εάν αριθμό από το 1 έως το 5 αναλόγως τον βαθμό ικανοποίησης.

1	2	3	4	5
---	---	---	---	---

Πόσο σας ενδιαφέρει να τρέφεστε υγιεινά στη δουλειά σας;

Επιλέξτε εάν αριθμό από το 1 έως το 5 αναλόγως του βαθμού ενδιαφέροντος.

1	2	3	4	5
---	---	---	---	---

Πόσο συχνά παραγγέλνετε απ' έξω το φαγητό που τρώτε στη δουλειά;

Επιλέξτε έναν αριθμό από το 1 έως το 5 βάσει της συχνότητας.

1	2	3	4	5
---	---	---	---	---

Το να παραγγέλνετε απ' έξω το φαγητό που τρώτε στη δουλειά το θεωρείτε...

Βαθμολογείτε με το 1 να δηλώνει το χαμηλότερο βαθμό και το 3 τον υψηλότερο

	1	2	3
Εξοικονόμηση χρόνου			
Ακριβή συνήθεια			
Όχι πολύ πιο ακριβό απ' το να μαγειρεύω ο ίδιος/η ίδια			
Ξεκούραστο			
Ανθυγιεινό			
Υγιεινό, αν κάνεις σωστές επιλογές			

Πόσο συχνά παραγγέλνετε στην καθημερινότητά σας φαγητό απ' έξω;

Επιλέξτε από το 1 έως το 5 τον αριθμό που ανταποκρίνεται στη συχνότητα που παραγγέλνετε απ' έξω

1	2	3	4	5
---	---	---	---	---

Το μέσο που προτιμάτε να πραγματοποιείτε τις παραγγελίες σας είναι:

- ☐ Τηλεφωνική επικοινωνία
- ☐ Εφαρμογή κινητού/internet

☐ Πηγαίνετε οι ίδιοι στο αντίστοιχο κατάστημα

☐ Κανένα, δεν παραγγέλνω απ' έξω

Παραγγέλνετε φαγητό μέσω internet και αν ναι, πόσο συχνά;

☐ Ναι, συνεχώς

☐ Συχνά

☐ Ναι, έχω παραγγείλει κάποιες φορές

☐ Όχι, δεν το προτιμώ

**Γνωρίζετε εφαρμογές που να αφορούν την ηλεκτρονική παραγγελιο-
ληψία φαγητού;**

☐ Ναι, αρκετές

☐ Ναι, πάνω από μία

☐ Όχι, δεν γνωρίζω καμία

Έχετε χρησιμοποιήσει κάποια από αυτές τις εφαρμογές;

☐ Ναι, πολλές φορές

☐ Ναι, αλλά λίγες φορές

☐ Όχι, ποτέ

Έχετε κάποια εφαρμογή τέτοιου τύπου στο κινητό σας ;

☐ Ναι

☐ Δεν θυμάμαι

☐ Όχι

**Χαρακτηριστικά που θα θέλατε να έχει μια εφαρμογή παραγγελίας
φαγητού**

*Βαθμολογείτε με το 1 να δηλώνει το χαμηλότερο βαθμό και το 3 τον
υψηλότερο.*

	1	2	3
Ευκολία στη χρήση			
Ασφάλεια συναλλαγών			
Ασφάλεια προσωπικών στοιχείων			
Ωραία εμφάνιση			
Ποικιλία επιλογών σε εστιατόρια και φαγητά			
Προσφορές και εκπτώσεις			
Παρουσίαση χαρακτηριστικών των φαγητών που θα παραγγείλω (πχ. εικόνα, συστατικά, θερμίδες κλπ.)			
Να μου προτείνει φαγητά βάσει των προτιμήσεών μου			
Να μου προτείνει ένα εβδομαδιαίο μενού			
Να μπορώ να προπαραγγείλω φαγητά για όλες τις μέρες της εβδομάδας πέραν της τρέχουσας			
Ενημέρωση χρόνου παράδοσης παραγγελίας			
Δυνατότητα ακύρωσης παραγγελίας			
Συμβατότητα με όλες τις συσκευές (tablet, smartphone, laptop)			
Δυνατότητα να κάνω κάποιο review			
Να μπορώ να παραγγείλω μόνο από κοντινά εστιατόρια			

Ποιά από τα παρακάτω θεωρείτε πλεονεκτήματα που προσφέρει η χρήση εφαρμογών παραγγελίας φαγητού;

- ☐ Δεν χρειάζεται να μαζεύω πια φυλλάδια τα οποία συχνά δεν είναι σωστά ενημερωμένα για τις τρέχουσες τιμές των προϊόντων
- ☐ Είναι οικονομικότερος τρόπος παραγγελίας καθώς δεν χρειάζεται να κάνω κλήση σε σταθερό ή κινητό τηλέφωνο
- ☐ Οι παραγγελίες μου γίνονται πιο γρήγορα και χωρίς λάθη που οφείλονται σε ασυνεννοησία με υπάλληλο του καταστήματος μέσω τηλεφώνου
- ☐ Περιορίζονται τα λάθη και υπάρχει καλύτερη διαχείριση όταν κάνω παραγγελία για πολλά άτομα
- ☐ Είμαι πάντα ενήμερος/η για τις προσφορές των καταστημάτων. Τηλεφωνικά, συχνά, ξεχνούν να μου αναφέρουν την ύπαρξη προσφορών.
- ☐ Μπορώ να διατυπώσω online την άποψή μου για καταστήματα και προϊόντα αλλά και να δω απόψεις άλλων χρηστών που με βοηθούν στην επιλογή μου
- ☐ Μπορώ να κάνω την παραγγελία μου από οποιαδήποτε συσκευή χρησιμοποιώ (smartphone, tablet, laptop)
- ☐ Ενημερώνομαι για την κατάσταση της παραγγελίας μου
- ☐ Δεν θεωρώ ότι έχουν κάποιο πλεονέκτημα
- ☐ Άλλο...

Τι θα χαρακτηρίζατε ως μειονέκτημα σε εφαρμογές ηλεκτρονικής παραγγελίας φαγητού που γνωρίζετε ή χρησιμοποιείτε;

Παρακαλούμε, γράψτε την άποψή σας, για εμάς είναι σημαντική!
(Απαντητικό πλαίσιο με ελεύθερο κείμενο)

Η ηλικία σας είναι:

- ☐ 18-27
- ☐ 28-37
- ☐ 38-47
- ☐ 48 και άνω

Το φύλο σας είναι:

- ☐ Άνδρας
- ☐ Γυναίκα

Σε ποια πόλη εργάζεστε;

Σε περίπτωση που η πόλη δεν βρίσκεται στη λίστα παρακαλώ επιλέξτε την πιο κοντινή σε εσάς από τις διαθέσιμες.

- ☐ Αθήνα
- ☐ Κόρινθος
- ☐ Γιαννιτσά
- ☐ Μυτιλήνη
- ☐ Χίος
- ☐ Σαλαμίνα
- ☐ Κέρκυρα
- ☐ Πύργος
- ☐ Κιλκίς
- ☐ Θήβα
- ☐ Τρίπολη
- ☐ Άρτα
- ☐ Λιβαδειά
- ☐ Κως
- ☐ Έδεσσα
- ☐ Ορεστιάδα
- ☐ Φλώρινα
- ☐ Σπάρτη
- ☐ Ναύπλιο

- ☐ Καστοριά
- ☐ Γρεβενά
- ☐ Πάτρα
- ☐ Θεσ/νίκη
- ☐ Λάρισα
- ☐ Ηράκλειο
- ☐ Βόλος
- ☐ Ιωάννινα
- ☐ Τρίκαλα
- ☐ Χαλκίδα
- ☐ Σέρρες
- ☐ Ξάνθη
- ☐ Κατερίνη
- ☐ Αγρίνιο
- ☐ Καλαμάτα
- ☐ Καβάλα
- ☐ Χανιά
- ☐ Λαμία
- ☐ Γρεβενά
- ☐ Κομοτηνή
- ☐ Ρόδος
- ☐ Δράμα
- ☐ Βέροια
- ☐ Κοζάνη
- ☐ Καρδίτσα
- ☐ Ρέθυμνο

Είστε εργαζόμενος στον...

- ☐ Ιδιωτικό τομέα
- ☐ Δημόσιο τομέα
- ☐ Άλλο...

Η ώρα που αποχωρείτε από το χώρο εργασίας σας είναι:

- ☐ Πριν τις 14:00
- ☐ Μετά τις 14:00 και πριν τις 17:00

- ☐ Μετά τις 17:00
- ☐ Άλλο...

8.1.3 Ερωτηματολόγια (2EAT) - Δεύτερη Διάδραση

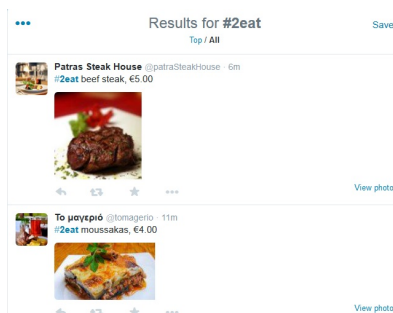
1. Το εστιατόριο κοινοποιεί (με tweet) το γεύμα που θέλει να διαθέσει μέσω της πλατφόρμας 2eat, χρησιμοποιώντας το hashtag "#2eat" ακολουθούμενο από την ονομασία, την τιμή και την εικόνα του γεύματος.



2. Η πλατφόρμα απαντά στην κοινοποίηση του εστιατορίου με πληροφορίες για την θερμιδική αξία του συγκεκριμένου φαγητού. Έτσι οι χρήστες μπορούν να δουν πληροφορίες για το γεύμα πριν το παραγγείλουν.



3. Κάθε χρήστης μπορεί να δει όλα τα διαθέσιμα γεύματα στην περιοχή του κάνοντας αναζήτηση για #2eat.



4. Ο χρήστης παραγγέλνει το γεύμα απαντώντας στην αρχική κοινοποίηση του εστιατορίου χρησιμοποιώντας το hashtag "#2eat #order"

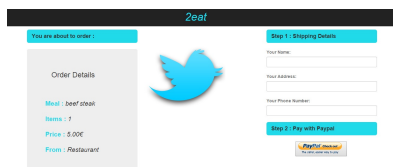
ακολουθούμενο απο την ονομασία του γεύματος και ένα hashtag με ποσότητα που θέλει να παραγγείλει.



5. Η πλατφόρμα απαντά στην παραγγελία του χρήστη με τον σύνδεσμο που πρέπει να ακολουθήσει για να ολοκληρώσει την παραγγελία του.



6. Όταν ο χρήστης ακολουθήσει το σύνδεσμο, οδηγείται στο περιβάλλον της πλατφόρμας όπου βλέπει τι πρόκειται να παραγγείλει, συμπληρώνει τα στοιχεία αποστολής και συνεχίζει στην πληρωμή μέσω PayPal.



7. Όταν ολοκληρωθεί η πληρωμή, η πλατφόρμα απαντά και πάλι στο χρήστη επιβεβαιώνοντας την παραγγελία του και ενημερώνοντάς τον για το χρόνο παράδοσης αυτής. Ταυτόχρονα του προτείνει μια φυσική δραστηριότητα που μπορεί να ακολουθήσει προκειμένου να κάψει τις θερμίδες του φαγητού που θα καταναλώσει. Ο χρήστης μπορεί να εκφράσει τη γνώμη του για το γεύμα που κατανάλωσε κάνοντας αντίστοιχα σχόλια. Έτσι οι υπόλοιποι χρήστες μπορούν να συμβουλευτούν την άποψή του.



Εφόσον έχετε παρατηρήσει το σενάριο που παρουσιάζεται στις παραπάνω εικόνες παρακαλώ αξιολογήστε τις παρακάτω προτάσεις. Βαθμολογήστε βάσει του πόσο συμφωνείτε. Χαμηλότερος βαθμός είναι το 1 και υψηλότερος το 3.

	1	2	3
Η διαδικασία είναι απλή και εύκολη			
Μου παρέχει ασφάλεια συναλλαγών			
Τα προσωπικά μου στοιχεία είναι ασφαλή εφόσον δεν απαιτείται εγγραφή			
Παρέχονται επαρκείς πληροφορίες για το γεύμα			
Ενημερώνομαι επαρκώς για την κατάσταση της παραγγελίας μου			
Θεωρώ ενδιαφέρουσα την πρόταση για τη δραστηριότητα που μπορώ να ακολουθήσω ώστε να διατηρήσω τη φυσική μου κατάσταση			
Εκφράζω τη γνώμη μου για το γεύμα που κατανάλωσα			
Μπορώ να βλέπω σχόλια άλλων χρηστών για τα διαθέσιμα γεύματα			

Θα χρησιμοποιούσατε την πλατφόρμα αυτή για να πραγματοποιήσετε ηλεκτρονική παραγγελία φαγητού;

Επιλέξτε από το 1 έως το 5 τον αριθμό που ανταποκρίνεται περισσότερο στην απάντησή σας.

1	2	3	4	5
---	---	---	---	---

Υπάρχουν χαρακτηριστικά που θα θέλατε να έχει η πλατφόρμα και δεν τα έχουμε συμπεριλάβει;

(Απαντητικό πλαίσιο με ελεύθερο κείμενο)

Εντοπίζετε ελαττώματα στην πλατφόρμα και, αν ναι, ποιά είναι αυτά;

(Απαντητικό πλαίσιο με ελεύθερο κείμενο)

Ποια η άποψή σας για την πλατφόρμα μας;

Η γνώμη σας είναι σημαντική για εμάς.
(Απαντητικό πλαίσιο με ελεύθερο κείμενο)

Το φύλο σας είναι:

- ☐ Άνδρας
- ☐ Γυναίκα

Η ηλικία σας είναι:

- ☐ 18-27
- ☐ 28-37
- ☐ 38-47
- ☐ 48 και άνω

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

2.1	Τα χαρακτηριστικά "κλειδιά" που κατευθύνουν την πελατοκεντρική ανάπτυξη λογισμικού	8
2.2	Η κυψέλη του Peter Morville's για την εμπειρία χρήστη. Κάθε κυψελίδα αντιπροσωπεύει επιμέρους έννοιες που συνθέτουν την εμπειρία χρήση (UX)	10
2.4	Το μοντέλο ανθρωποκεντρικού σχεδιασμού κατά ISO 9241-210:2010	14
3.1	Διακριτά στάδια του μοντέλου καταρράκτη	27
4.1	Σχήμα βάσης δεδομένων EAT-IN.	39
4.2	Αρχική σελίδα EAT-IN.	40
4.3	Διαδικασία αρχικής εγγραφής κοινή και για τα δύο είδη πελάτων.	41
4.4	Εγγραφή χρήστη.	41
4.5	Εγγραφή συνεργάτη.	41
4.6	Πάνελ συνεργάτη.	42
4.7	Διαγραφή ή διόρθωση διαθέσιμων γευμάτων.	43
4.8	Αλλαγή εικόνας γεύματος.	43
4.9	Προσθήκη νέου γεύματος.	44
4.10	Διαχείριση μενού.	45
4.11	Πίνακας παραγγελιών συνεργάτη	45
4.12	Εβδομαδιαίο μενού βάσει προτιμήσεων χρήστη.	46
4.13	Συγκεντρωτική εικόνα διαθέσιμων γευμάτων κατηγοριοποιημένα βάσει του τύπου τους.	47
4.14	Διαχείριση μενού.	48
4.15	οι διαθέσιμες πληροφορίες για το γεύμα	48

4.16 Καλάθι αγορών χρήστη.	49
4.17 Είσοδος στον λογαριασμό PayPal του χρήστη.	50
4.18 Ολοκλήρωση πληρωμής μέσω PayPal.	50
4.19 Ολοκλήρωση παραγγελίας χρήστη.	51
4.20 Διανομή παραγγελιών στους ανάλογους συνεργάτες.	52
4.21 Τμήματα της Spring (Spring modules).	56
4.22 Βήματα ολοκλήρωσης HTTP αιτήματος με Spring MVC.	57
4.23 Η Spring Boot σε σχέση με το χρήστη και τη Spring.	59
5.1 Διαδικασία Ανάπτυξης Πελατών - Πηγή [2]	83
6.1 Ποσοστά ερωτηθέντων που είναι ενήμεροι ή έχουν κατεβάσει εφαρμογές ηλεκτρονικής παραγγελίας φαγητού.	98
6.2 Χαρακτηριστικά που αξιολογήθηκαν ως υψηλής σημασίας.	100
6.3 Χαρακτηριστικά που αξιολογήθηκαν ως μέτριας σημασίας.	100
6.4 Χαρακτηριστικά που αξιολογήθηκαν ως λιγότερο σημαντικά.	101
6.5 Πηγή: “Καταγραφή και αξιολόγηση των Ελληνικών Ηλεκτρονικών Καταστημάτων - 2014”, Εργαστήριο Ηλεκτρονικού Εμπορίου (ELTRUN), Οικονομικό Πανεπιστήμιο Αθηνών.	102
6.6 Πλεονεκτήματα χρήσης εφαρμογών ή πλατφορμών ηλεκτρονικής παραγγελίας φαγητού σύμφωνα με τους ερωτηθέντες.	103
6.7 Το εστιατόριο κάνει tweet το γεύμα.	107
6.8 Αυτόματη απάντηση με τα διατροφικά χαρακτηριστικά του γεύματος που προκύπτουν από αναζήτηση με free-text αναζήτηση.	108
6.9 Αποτελέσματα αναζήτησης για το 2eat.	109
6.10 Ο πελάτης τοποθετεί παραγγελία με ένα απλό tweet.	110
6.11 Αυτόματο tweet στον πελάτη που αναγράφει το σύνδεσμο που ακολουθεί για να πληρώσει.	110
6.12 Ο χρήστης παρέχει πληροφορίες παράδοσης και καλείται να πληρώσει.	111
6.13 Αυτόματη επιβεβαίωση παραγγελίας περιλαμβάνοντας χρόνο παράδοσης. Σχόλιο πελάτη.	112
6.14 Αξιολόγηση χαρακτηριστικών 2eat σε σχέση με τα επίπεδα ικανοποίησης των πελατών.	113

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

5.1	Διαφορές Παραδοσιακών και Ευέλικτων Μεθόδων	85
5.2	Διαφορές Παραδοσιακών και Ευέλικτων Μεθόδων	87

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] P. Morville *Ambient Findability: What We Find Changes Who We Become*, October 3rd 2005, ISBN-10: 0596007655
- [2] Wright, P. C. Blythe, M. *User experience research as an inter-discipline: Towards a UX Manifesto*. In E.Law, A. Vermeeren, M. Hassenzahl, M. Blythe (Eds.), *Towards a UX Manifesto - Proceedings of a cost294-affiliated workshop on BHCI 2007*, 2007, pp. 65-70.
- [3] Gentile, Chiara, N. Spiller and G. Noci (2007), *How to Sustain the Customer Experience: An Overview of Experience Components that Cocreate Value with the Customer*, *European Management Journal*, 25 (5), p. 397
- [4] Ross Beard, *Customers Satisfaction Strategies*, November 2013,(ηλεκτ.βιβλ.) p.12 <http://blog.clientheartbeat.com/downloads/customer-satisfaction-strategies-ebook.pdf>
- [5] F.Buttle, *Customer Relationship Management: Concepts and Technologies*, 2009, p.48
- [6] C. Marsh, P. Sparrow and M. Hird *Is Customer Centricity A Movement Or Myth? Opening The Debate For HR* White Paper 10/03 December 2010, p.5
- [7] International Organization for Standardization (2009). *Ergonomics of human system interaction - Part 210: Human-centered design for interactive systems (formerly known as 13407)*. ISO FDIS 9241-210:2009.
- [8] Duyne, Douglas K.; Landay, James A.; Hong, Jason I., *The Design of Sites: Patterns for Creating Winning Web Sites (2nd Edition)*, Prentice

- Hall, 2006, ch.2, p.13
- [9] United States, Navy Mathematical Computing Advisory Panel (29 June 1956), Symposium on advanced programming methods for digital computers, [Washington, D.C.]: Office of Naval Research, Dept. of the Navy, OCLC 10794738
 - [10] Boehm, B. (2002). Get Ready for Agile Methods, with care. Computer, 35, 1. Pp. 64-69.
 - [11] I. Sommerville *Βασικές αρχές τεχνολογίας λογισμικού*, 2009,ISBN 978-960-461-220-8, ch 1.
 - [12] W. W. Royce, "Managing the development of large software systems: Concepts and techniques", IEEE ESCON, vol. 26, no. 8, pg. 1-9, 1970
 - [13] I. Sommerville *Βασικές αρχές τεχνολογίας λογισμικού*, 2009,ISBN 978-960-461-220-8, ch 2.
 - [14] Yogesh Deshpande, Martin Gaedke "Web Engineering: Developing Successful Web Applications In A Systematic Way", 14th International World Wide Web Conference, 10- 14 May, 2005, Chiba, Jap
 - [15] Δ.Πίχλιαβας, *Ανάλυση Τεχνικών Ανάπτυξης Καινοτόμων Προϊόντων Λογισμικού και Ανάπτυξη Μεθοδολογίας Εκμείευσης και Περιγραφής Απαιτήσεων Χρήστη*, Διπλωματική Εργασία Ε.Μ.Π., Αθήνα, Οκτώβριος 2015.
 - [16] Simon Salloum, *Choosing the right tools for the job*, 2014 <https://blog.inf.ed.ac.uk/sapm/2014/02/04/choosing-the-right-tools-for-the-job/>
 - [17] I. Sommerville *Βασικές αρχές τεχνολογίας λογισμικού*, 2009,ISBN 978-960-461-220-8, chapter
 - [18] M. A. Awad, *A Comparison between Agile and Traditional Software Development Methodologies*, 2005 <http://www.unf.edu/broggio/cen6940/ComparisonAgileTraditional.pdf>
 - [19] Prof. Abhijit Khasnis, *A Comparative Analysis on Heavyweight V/s Lightweight Strategies of Software Development*, Tirupati Institute of Management
 - [20] Sridhar Nerur, RadhaKanta Mahapatra, George Mangalaraj, *Challenges of Migrating to Agile Methodologies*, COMMUNICATIONS OF THE ACM, 2005/Vol. 48, No.5
 - [21] Powell, Thomas A., David L. Jones, and Dominique C. Cutts. *Web site engineering: beyond Web page design*. Prentice-Hall, Inc., 1998.
 - [22] N. Uikey A., U. Suman, *Lifecycle Model for Web-based Application*

- Development: Incorporating Agile and Plan-driven Methodology*, International Journal of Computer Applications (0975 – 8887) Volume 117 – No. 19, May 2015
- [23] Γ. Δρίγκας, *Πλατφόρμες Ανάπτυξης Διαδικτυακών Εφαρμογών σε Java: Spring MVC και STRUTS 2. Συγκριτική Μελέτη Ποιότητας και Τεχνικού Χρέους*, Εφαρμοσμένης Πληροφορικής, Πανεπιστήμιο Μακεδονίας, 2016.
- [24] A. Cockburn *Agile Software Development*, 2002, Addison-Wesley
- [25] M. Fowler, *The New Methodology*, <http://www.martinfowler.com/articles/newMethodology.html> Accessed on 08/08/2016
- [26] M. Fowler, *Is Design Dead?*”, *Software Development* <http://www.martinfowler.com/articles/designDead.html>
- [27] L. Williams and A. Cockburn, “Agile Software Development: It’s about Feedback and Change,” *IEEE Computer*, June 2003, pp. 39-43
- [28] P. Bowers, *Highpoints From the Agile Software Development Forum*, 2002
- [29] Beck, K. (2000). *Extreme Programming Explained: Embrace Change*. Reading, Massachusetts: Addison-Wesley.
- [30] I. G. Stamelos and P. D. Sfetsos, *Agile Software Development Quality Assurance*, IDEA GROUP PUBLISHING, Febr. 2007, ISBN: 978-1-59904-216-9.
- [31] Blank, S. G., *The Four Steps to the Epiphany: Successful Strategies for Products that Win*, KS Ranch, 2005, 1–15
- [32] M. Cohn, *User Stories Applied: For Agile Software Development*, 2004, Addison-Wesley Professional, pp 234-241
- [33] Yasuhiro Monden (1998), *Toyota Production System, An Integrated Approach to Just-In-Time*, Third edition, Norcross, GA: Engineering Management Press, ISBN 0-412-83930-X.
- [34] Mary Poppendieck; Tom Poppendieck (2003). *Lean Software Development: An Agile Toolkit*. Addison-Wesley Professional. ISBN 978-0-321-15078-3. Chapter 2
- [35] See Shigeo Shingo, *Study of ‘Toyota’ Production System*, Productivity Press, 1981, Chapter 2.3.
- [36] P. Nallasenapathi, Saksoft *Lean Development – A team approach to Software Application Development*, March 2006 p. 5-6
- [37] Using Agile Processes and Modeling To Build Enterprise Applications <http://www.oracle.com/technetwork/articles/grid/agile-modeling-094130.html>
- [38] E. Ries, *The Lean Startup: How Today’s Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*, Published September

13th 2011

- [39] S. Blank, *Why the Lean Start-Up Changes Everything*, Harvard Business Review, 2013
- [40] S.G. Blank, B. Dorf *The Startup Owner's Manual: The Step-By-Step Guide for Building a Great Company*, Published 2012, chapter 5, p. 20-29
- [41] Redouane , *Guidelines for Improving the Development of Web-Based Applications* Proceedings of the Fourth International Workshop on Web Site Evolution (WSE'02) 0-7695-1804-4/02 2002 IEEE,
- [42] T. O'Reilly, *What Is Web 2.0 - Design Patterns and Business Models for the Next Generation of Software*, Published 2005
- [43] Morris, James H. (August 30, 2006). *Software Product Management and the Endless Beta*". *Carnegie Mellon University*. Retrieved July 7, 2009.
- [44] Riehle, *A comparison of the value systems of Adaptive Software Development and Extreme Programming: How methodologies may learn from each other*, Proceedings of the First International Conference on Extreme Programming and Flexible Processes in Software Engineering, XP 2000, 21-23 Jun. 2000, Cagliari, Italy, pg. 35-50
- [45] Αβούρης, Ν., Κατσάνος, Χ., Τσέλιος, Ν., Μουστάκας, Κ., 2015. *Εισαγωγή στην αλληλεπίδραση ανθρώπου-υπολογιστή*. [ηλεκτρ. βιβλ.] Αθήνα:Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. <http://hdl.handle.net/11419/4213>, ch2, p.178-179
- [46] Ερευνα Χρήσης Τεχνολογιών Πληροφόρησης και Επικοινωνίας απο Νοικοκυριά και Ατομα – 2015 "ΒαθμόςΧρήσης Νέων Τεχνολογιών" <http://www.statistics.gr/documents/20181/51246a10-a5d9-44ae-9186-d17d55a496a0>
- [47] D. Gage, *The Venture Capital Secret: 3 Out of 4 Start-Ups Fail*, "The Wall Street Journal", "2012"
- [48] C. C. Chang, C. F. Yan, and J. S. Tseng. *Perceived convenience in an extended technology acceptance model: Mobile technology and english learning for college students*. *Australasian Journal of Educational Technology* , 28(5):812-813.
- [49] Mobile Vision, *Developer Economics 2013*, Published January 2013, <http://www.slideshare.net/andreasc/vision-mobile-developer-economics-2013>
- [50] Electronic Commerce Laboratory, Athens University of Economics and Busines, *Annual ELTRUN Report 2014* http://www.eltrun.gr/wp-content/uploads/2015/03/ELTRUN_ecommerce_survey_2015.pdf

