



**ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΕΙΡΑΙΩΣ**

Μεταπτυχιακή
Διπλωματική Εργασία

ΑΝΑΠΤΥΞΗ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

**ΚΛΕΦΤΑΚΗΣ ΣΠΥΡΙΔΩΝ
(ΜΕ 1621)**

**ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ
ΜΑΡΙΝΟΣ ΘΕΜΙΣΤΟΚΛΕΟΥΣ**

Πανεπιστήμιο Πειραιώς
Τμήμα Ψηφιακών Συστημάτων
Προηγμένα Πληροφοριακά Συστήματα



Μεταπτυχιακή Διπλωματική Εργασία

ΑΝΑΠΤΥΞΗ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΤΟΥ

Κλεφτάκη Σπυρίδωνα (ΜΕ 1621)

Επιβλέπων Καθηγητής: Μαρίνος Θεμιστοκλέους

Φεβρουάριος 2018

Εξεταστική Επιτροπή:

Καθηγητής Μαρίνος Θεμιστοκλέους

Καθηγητής Δημοσθένης Κυριαζής

Ευχαριστίες

Ευχαριστώ θερμά τον Καθηγητή του τμήματος και επιβλέποντα της παρούσας διπλωματικής εργασίας, Μαρίνο Θεμιστοκλέους, για την πολύτιμη συμβολή του στη διεκπεραίωση της εργασίας καθώς και για την καθοδήγηση, τη συνεχή υποστήριξη και βοήθειά του καθόλη τη διάρκειά της.

Θα ήθελα, επίσης, να ευχαριστήσω τον συνάδελφό μου Γιώργο για τις πολύτιμες συμβουλές του σε προηγμένα θέματα προγραμματισμού.

Τέλος, ευχαριστώ τους δικούς μου ανθρώπους, την οικογένειά μου και κυρίως τη σύζυγό μου Χαρά για την υποστήριξη που μου παρείχε και κυρίως για την υπομονή και την κατανόηση της για τον περιορισμένο χρόνο που αφιέρωσα σε αυτή και το νέο μας μέλος, ώστε να ολοκληρώσω με επιτυχία το μεταπτυχιακό πρόγραμμα σπουδών.

Περίληψη

Με την παρούσα εργασία επιχειρείται μία παράθεση των σημαντικότερων μεθοδολογιών ανάπτυξης πληροφοριακών συστημάτων, όπως αυτές τεκμηριώνονται στη διεθνή βιβλιογραφία.

Αρχικά αναλύονται κάποιες βασικές έννοιες, οι οποίες αποτελούν τη βάση κατανόησης όλων των μεθοδολογιών ανάπτυξης ενός πληροφοριακού συστήματος. Στη συνέχεια, η μελέτη πραγματεύεται τα παραδοσιακά μοντέλα ανάπτυξης πληροφοριακών συστημάτων δίνοντας λεπτομερείς περιγραφές για τις σημαντικότερες, βιβλιογραφικά, εξ αυτών. Έπειτα, η μελέτη αναλύει και περιγράφει τις κυριότερες εκ των ευέλικτων μεθοδολογιών καταλήγοντας στην μεθοδολογία Scrum, ενώ παρατίθεται, μέσα από ένα δομημένο πλαίσιο συγκριτικής αξιολόγησης, η απόφαση επιλογής της καταλληλότερης μεθοδολογίας.

Η εργασία, στη συνέχεια, περιλαμβάνει μελέτη περίπτωσης, η οποία πραγματοποιείται σε πραγματικές παραγωγικές συνθήκες, σε οργανισμό του δημοσίου τομέα και συγκεκριμένα στον στρατό ξηράς και το αρμόδιο τμήμα ανάπτυξης πληροφοριακών συστημάτων. Η εφαρμογή της μεθοδολογίας Scrum γίνεται για πρώτη φορά στο συγκεκριμένο τμήμα και πραγματοποιείται αναλυτική περιγραφή των ευρημάτων της για το διάστημα που διήρκησε η παρούσα διπλωματική.

Τέλος, με την εξαγωγή συμπερασμάτων από την αξιολόγηση των ευρημάτων της μελέτης περίπτωσης, επιχειρείται μία τεκμηρίωση της αναγκαιότητας υιοθέτησης της υπόψιν μεθοδολογίας και των ευέλικτων μεθοδολογιών γενικότερα.

Abstract

This paper attempts to present the most important methodologies for the development of information systems, as documented in the international literature.

Firstly, some basic concepts are analyzed, which are the basis for understanding all the methodologies of developing an information system. The study then deals with traditional models of information systems development, giving detailed descriptions of the most important bibliographic ones. Then, the study analyzes and describes the most important of the agile methodologies resulting in the Scrum, followed by the decision to select the most appropriate methodology which is provided through a structured benchmarking framework. The paper then includes a case study, which takes place under real production conditions, in a public-sector organization, specifically the army and the responsible IT development department. The application of Scrum methodology is for the first time in this department and a detailed description of its findings for the duration of this diploma is carried out. Finally, drawing conclusions from the evaluation of the findings of the case study attempts to document the necessity of adopting the methodology and the agile methodologies in general.

Περιεχόμενα

Ευχαριστίες	3
Περίληψη	4
Abstract.....	5
Κατάλογος Εικόνων	10
1. Εισαγωγή	12
1.1 Ορισμός του Προβλήματος.....	12
1.2 Δομή της Μεταπτυχιακής Διπλωματικής Εργασίας.....	13
1.3. Μεθοδολογία Βιβλιογραφικής Ανασκόπησης.....	14
1.4. Συνεισφορά της Μεταπτυχιακής Διπλωματικής Εργασίας	15
2. Βασικές Έννοιες	16
2.1. Έργο (Project)	16
2.2. Μεθοδολογία.....	16
2.3. Κύκλος Ζωής	17
3. «Παραδοσιακά» Μοντέλα Ανάπτυξης Λογισμικού	19
3.1. Μοντέλο Καταρρακτοειδούς Ανάπτυξης ΠΣ.....	19
3.2. Δομημένη Ανάλυση και Σχεδιασμός Συστημάτων (SSADM)	23
3.3. V – Μοντέλο.....	24
3.4. Μεθοδολογία Λειτουργικής Επαύξησης.....	25
3.5. Μοντέλο Δημιουργίας Πρωτοτύπου	27
3.6. Σπειροειδές Μοντέλο	29
3.7. Συνοψίζοντας.....	31
4. Ευέλικτες (Agile) Μεθοδολογίες Ανάπτυξης Λογισμικού.....	32
4.1. Γενικά Χαρακτηριστικά.....	32

4.1.1.	Αξίες – Αρχές – Ιδιότητες.....	34
4.1.2.	Διασφάλιση Ποιότητας.....	37
4.2.	Μοντέλα Agile Μεθοδολογιών	38
4.2.1.	Ακραίος Προγραμματισμός (Extreme Programming)	38
4.2.2.	Ανάπτυξη με Βάση τις Δοκιμές (Test Driven Development)	42
4.2.3.	Ανάπτυξη με Βάση τα Χαρακτηριστικά (Feature Driven Development).....	44
4.2.4.	Μεθοδολογία Ανάπτυξη Δυναμικών Συστημάτων (Dynamic Systems Development Method)	46
4.2.5.	DevOps	49
4.2.6.	Μεθοδολογία Scrum.....	50
4.3.	Επιλογή ή Μη των Ευέλικτων Μεθοδολογιών.....	52
5.	Επιλογή Μεθοδολογίας Ανάπτυξης Συστημάτων	56
6.	Μεθοδολογία Scrum	59
6.1.	Φάσεις της Μεθοδολογίας Scrum	59
6.2.	Ρόλοι και Αρμοδιότητες στη Scrum.....	61
6.2.1.	Ο ιδιοκτήτης του προϊόντος (Product Owner).....	62
6.2.2.	Η ομάδα (The Team) ή ομάδα ανάπτυξης.....	62
6.2.3.	Ο διαχειριστής ή ειδικός του Scrum (The Scrum Master)	64
	Η υπηρεσία του Scrum Master προς τον Product Owner.....	64
	Η υπηρεσία του Scrum Master προς την Ομάδα Ανάπτυξης	65
	Η υπηρεσία του Scrum Master προς τον Οργανισμό	65
6.3.	Διαδικασία της Μεθοδολογίας.....	66
6.3.1.	Product Backlog	66
6.3.2.	Sprint.....	68
6.3.3.	Σχεδιασμός του Sprint (Sprint Planning)	70

Θέμα 1: Τι μπορεί να γίνει σε αυτό το Sprint;	70
Θέμα 2: Πώς θα εκτελεστούν οι επιλεγμένες εργασίες;	71
6.3.4. Στόχος του Sprint	73
6.3.5. Διαχείριση του Sprint.....	73
6.3.6. Ακύρωση του Sprint	74
6.3.7. Καθημερινή συνεδρίαση Scrum (Daily Scrum).....	75
6.3.7.1. Παρακολούθηση της προόδου προς τον Στόχο του Sprint ..	76
6.3.7.2. Επαύξηση	78
6.3.8. Επισκόπηση του Sprint (Sprint Review).....	78
6.3.9. Αναδρομή στο Sprint (Sprint Retrospective)	80
6.3.10. Ορισμός του «Έτοιμου» (Definition of Done)	81
6.3.11. Ταχύτητα (Velocity)	82
6.4. Διαφάνεια Αντικειμένων της Scrum.....	84
7. Σύγκριση Μεθοδολογίας Scrum με «Παραδοσιακές» Μεθοδολογίες.....	85
7.1. Πλεονεκτήματα της Scrum συγκριτικά με την «Καταρρακτοειδή» Μεθοδολογία.....	92
7.2. Μειονεκτήματα της Scrum συγκριτικά με την «Καταρρακτοειδή» Μεθοδολογία.....	94
8. Μελέτη Περίπτωσης	96
8.1. Εισαγωγή.....	96
8.2. Ερευνητική Μεθοδολογία	96
8.3. Στοιχεία του Έργου.....	97
8.4. Ανάθεση Ρόλων.....	98
8.5. Παρακολούθηση Μεθοδολογίας	99
8.6. Εφαρμογή Μεθοδολογίας Scrum.....	99
8.6.1. Βήμα 1 – Όραμα.....	100

8.6.2.	Βήμα 2 – Product Backlog.....	101
8.6.3.	Βήμα 3 – Προγραμματισμός εκδόσεων	102
8.6.4.	Βήματα 4,5,6,7 Εκτέλεση των Sprints	102
8.6.4.1.	Το πρώτο Sprint	102
	Σχεδιασμός του Sprint (Sprint Planning).....	102
	Καθημερινές συνεδριάσεις Scrum (Daily Scrum).....	103
	Επισκόπηση του Sprint.....	104
	Αναδρομή στο Sprint	106
	Συμπεράσματα εκτέλεσης πρώτου Sprint.....	107
8.6.4.2.	Τα επόμενα Sprints.....	107
8.6.5.	Συμπεράσματα Εφαρμογής της Μεθοδολογίας Scrum.....	109
9.	Συμπεράσματα	111
	Βιβλιογραφία.....	113
	Παράρτημα Α	A-1
	Παράρτημα Β	B-1
	Παράρτημα Γ	Γ-1

Κατάλογος Εικόνων

Εικόνα 1. Καταρρακτοειδές Μοντέλο (Βασιλακόπουλος Γ., 2018)	21
Εικόνα 2. Το V μοντέλο.....	25
Εικόνα 3. Μεθοδολογία Λειτουργικής Επαύξησης (Βασιλακόπουλος Γ., 2018)	26
Εικόνα 4. Μεθοδολογία Δημιουργίας Πρωτοτύπων (Βασιλακόπουλος Γ., 2018)	28
Εικόνα 5. Σπειροειδές μοντέλο.....	29
Εικόνα 6. Γενική Μεθοδολογία Ευέλικτης Ανάπτυξης Συστημάτων (Βασιλακόπουλος Γ., 2018).....	34
Εικόνα 7. Κύκλος ζωής μεθοδολογίας Ακραίου Προγραμματισμού (Beck K. , 2004).....	39
Εικόνα 8. Κύκλος ζωής της Ανάπτυξης με βάση τις δοκιμές (Excirial, 2009) .	43
Εικόνα 9. Μεθοδολογία ανάπτυξης με βάση τα χαρακτηριστικά (Palmer , S.R. and Felsing J.M., 2002)	45
Εικόνα 10. Διαδικασία ανάπτυξης δυναμικών συστημάτων (Stapleton, 1997)	48
Εικόνα 11. Οδηγός επιλογής Μεθοδολογίας (https://www.lucidchart.com)	58
Εικόνα 12. Φάσεις του Sprint της Scrum.....	61
Εικόνα 13. Τα βήματα της Scrum.....	66
Εικόνα 14. Ενδεικτικό παράδειγμα ενός Product Backlog (Deemer et al., 2010)	67
Εικόνα 15. Διαδικασίες της μεθοδολογίας Scrum.....	68
Εικόνα 16. Παράδειγμα Sprint Backlog (Deemer et al., 2010)	72
Εικόνα 17. Sprint Task Board (Hundermark, 2009)	74
Εικόνα 18. Παράδειγμα Sprint Burndown Chart [Deemer et al., 2010]	77
Εικόνα 19. Διαδικασία Scrum (Deemer et al., 2010).....	82

Εικόνα 20. Διαφορά Παραδοσιακών και Ευέλικτων Μεθοδολογιών (Owen, R., Koskela, L.J., Henrich, G., and Codinhoto, R., 2006)	85
Εικόνα 21. Βήματα - Στάδια της Scrum (Layton, 2012).....	99
Εικόνα 22. Sprint Backlog και ανάθεση εργασιών (Task Allocation)	103
Εικόνα 23. Πίνακας εποπτείας Sprint.....	104
Εικόνα 24. Σελίδα εισόδου (login) του Πληροφοριακού Συστήματος.....	105
Εικόνα 25. Σελίδα αναζήτησης νεοσυλλέκτου	105
Εικόνα 26. Σχήμα Βάσης Δεδομένων Πληροφοριακού Συστήματος	106
Εικόνα 27. Ταχύτητα των Sprints στις εκδόσεις του Πληροφοριακού Συστήματος Κατάταξης Νεοσυλλέκτων.....	108

1. Εισαγωγή

1.1 Ορισμός του Προβλήματος

Οι συνθήκες του σήμερα με την αυξανόμενη ανταγωνιστικότητα, την ταχύτητα των εξελίξεων, τόσο στην επιστημονική γνώση όσο και στην τεχνολογία, αλλά και η απαίτηση για παραγωγή και προώθηση στον πελάτη ποιοτικών υπηρεσιών και προϊόντων, έχουν δημιουργήσει την ανάγκη εφαρμογής συστημάτων διοίκησης που να αναγνωρίζουν τη φύση της επιχείρησης του 21^{ου} αιώνα και να προσεγγίζουν ολιστικά το σύνολο των παραμέτρων που δύναται να επιφέρουν ανταγωνιστικό πλεονέκτημα. Οι *Henderson* και *Venkatraman* θεωρούνται οι θεμελιωτές της έννοιας της εναρμόνισης μεταξύ ΠΣ και επιχειρηματικών δραστηριοτήτων του οργανισμού με τη δημοσίευση της εργασίας τους *“Alignment: Leveraging Information Technology for transforming organizations”* στο επιστημονικό περιοδικό *IBM Systems Journal*.

Οι σύγχρονες εταιρείες ανάπτυξης λογισμικού έρχονται αντιμέτωπες με προκλήσεις όπως η συνεχής αλλαγή των απαιτήσεων, η πίεση για γρήγορη παράδοση και η ανάγκη μείωσης κόστους λόγω ανταγωνισμού (Alleman, 2002). Σε μια προσπάθεια για την αντιμετώπιση αυτών των προκλήσεων, έχουν αναπτυχθεί μέθοδοι, γνωστές ως «Ευέλικτη διαχείριση έργου» και «Ευέλικτη ανάπτυξη λογισμικού» (Cohn, *Agile Estimating and Planning*, 2006). Η ευέλικτη προσέγγιση στοχεύει στην ταχύτερη παραγωγή ποιοτικών προϊόντων λογισμικού, καθώς επίσης και στη δημιουργία μεγαλύτερης αξίας και μέγιστης ικανοποίησης του πελάτη (Beck, K. et al., 2001). Ο σχεδιασμός και η παρακολούθηση της διαδικασίας ανάπτυξης λογισμικού με τη χρήση των ευέλικτων μεθοδολογιών είναι θεμελιώδη βήματα προς την εξασφάλιση της επιτυχίας (Chow, T., Cao D. B., 2008) και συμπεριλαμβάνουν την αποτελεσματική διαχείριση των πόρων και τη συνεχή παρακολούθηση της εξέλιξης χρησιμοποιώντας συγκεκριμένες μετρικές.

1.2 Δομή της Μεταπτυχιακής Διπλωματικής Εργασίας

Η παρούσα μελέτη οργανώνεται σε εννέα διακριτά κεφάλαια, καθώς και σε τρία παραρτήματα, συμπεριλαμβανομένου του εισαγωγικού κεφαλαίου, στο οποίο περιγράφεται ο ορισμός του προβλήματος, η μεθοδολογία βιβλιογραφικής ανασκόπησης, καθώς και η συνεισφορά της παρούσας διπλωματικής εργασίας.

Στο δεύτερο κεφάλαιο παρατίθενται κάποιες βασικές έννοιες χρήσιμες για την καλύτερη κατανόηση των μεθοδολογιών ανάπτυξης πληροφοριακών συστημάτων .

Στο τρίτο κεφάλαιο παρατίθενται τα λεγόμενα παραδοσιακά μοντέλα ανάπτυξης πληροφοριακών συστημάτων και περιγράφονται αναλυτικά τα σημαντικότερα βιβλιογραφικά εξ αυτών.

Το τέταρτο κεφάλαιο πραγματεύεται τις ευέλικτες μεθοδολογίες. Αφού γίνεται μία αναλυτική παρουσίαση των χαρακτηριστικών τους, στη συνέχεια παρουσιάζονται αναλυτικά οι σημαντικότερες εξ αυτών, ενώ παρατίθεται ένα συγκριτικό πλαίσιο αξιολόγησης για επιλογή ή μη των ευέλικτων μεθοδολογιών.

Στο πέμπτο κεφάλαιο παρουσιάζονται οι βασικοί κανόνες που διευκολύνουν τη λήψη απόφασης επιλογής μίας μεθοδολογίας.

Το έκτο κεφάλαιο πραγματεύεται, διεξοδικώς, τη μεθοδολογία Scrum, περιγράφοντας αναλυτικά τις φάσεις, τους ρόλους αλλά και τη διαδικασία που ακολουθείται κατά την εφαρμογή της.

Στο έβδομο κεφάλαιο δίδεται ένα δομημένο πλαίσιο σύγκρισης της μεθοδολογίας Scrum με την «καταρρακτοειδή» μεθοδολογία και παρατίθενται τόσο τα πλεονεκτήματα όσο και τα μειονεκτήματα της.

Το όγδοο κεφάλαιο αποτελεί το πρακτικό μέρος της διπλωματικής εργασίας. Αποτελεί τη μελέτη περίπτωσης εφαρμογής της μεθοδολογίας Scrum σε πραγματικό περιβάλλον και συγκεκριμένα στο τμήμα ανάπτυξης πληροφοριακών συστημάτων του στρατού ξηράς. Σκοπός είναι η παρατήρηση και αξιολόγηση των ευρημάτων, κατά την πρακτική εφαρμογή της

μεθοδολογίας για πρώτη φορά στο συγκεκριμένο τμήμα για τη διάρκεια της διπλωματικής εργασίας.

Στο ένατο και τελευταίο κεφάλαιο παρουσιάζονται τα γενικά συμπεράσματα της παρούσας μελέτης.

Στο τέλος της διπλωματικής εργασίας παρατίθενται τρία παραρτήματα. Στο πρώτο εξ αυτών παρουσιάζεται η Μεθοδολογία Μαλακών Συστημάτων που πραγματώθηκε για την κατανόηση της προβληματικής κατάστασης που ώθησε στην ανάγκη ενός νέου πληροφοριακού συστήματος, του εξεταζόμενου στη μελέτη περίπτωσης. Στο δεύτερο παράρτημα δίδεται ο κύκλος ζωής του υπόψιν πληροφοριακού συστήματος αν αυτό πραγματωνόταν με χρήση της παραδοσιακής «καταρρακτοειδούς» μεθοδολογίας. Τέλος, στο τρίτο παράρτημα παρουσιάζονται οι κυριότερες των τεχνολογιών που χρησιμοποιήθηκαν για την πραγμάτωση του πληροφοριακού συστήματος, καθώς και μια σύντομη περιγραφή του.

1.3. Μεθοδολογία Βιβλιογραφικής Ανασκόπησης

Η βιβλιογραφική ανασκόπηση θα ακολουθήσει την παραδοσιακή μορφή βιβλιογραφικών ανασκοπήσεων, την αποκαλούμενη αφηγηματική (narrative). Αποτελεί την κριτική συμπερίληψη επιλεγμένων εργασιών. Η αναζήτηση στο γνωστικό χώρο της μελέτης πραγματοποιείται αρχικά στις βιβλιογραφικές βάσεις scholar.google.com, academia.edu και ieeeexplore.ieee.org καθώς και στη μηχανή αναζήτησης της google. Επίσης μελετήθηκαν διεξοδικά βιβλία (David Avison & Guy Fitzgerald, 2006), (Βασιλακόπουλος Γ., 2018) καθώς και άρθρα και παραπομπές από επιλεγμένους διαδικτυακούς κόμβους.

Αξίζει να σημειωθεί ότι το γνωστικό αντικείμενο που αφορά στα εναλλακτικά μοντέλα και στις μεθοδολογίες ανάπτυξης πληροφοριακών συστημάτων, κινείται σε κάποιο βαθμό στους χώρους της πρακτικής. Ως συνέπεια, βασικές πηγές αναφοράς στην παρούσα μελέτη, προέρχονται από επιστημονικά βιβλία, από αναφορές επιλεγμένων δικτυακών κόμβων, blogs, forums, οδηγούς, καθώς και από επιστημονικά άρθρα και μελέτες που έχουν δημοσιευθεί από έγκυρους ακαδημαϊκούς και επιστημονικούς φορείς.

1.4. Συνεισφορά της Μεταπτυχιακής Διπλωματικής Εργασίας

Στόχος της παρούσας εργασίας είναι η λεπτομερής μελέτη της Scrum μεθοδολογίας και η αξιολόγηση της συγκριτικά με τα παραδοσιακά μοντέλα ανάπτυξης πληροφοριακών συστημάτων, προκειμένου να ερευνηθεί σε βάθος η καταλληλότητά της ή μη σε σύγχρονα επιχειρησιακά περιβάλλοντα ανάπτυξης λογισμικού. Σκοπός είναι επίσης, η μελέτη και αξιολόγησή της μεθοδολογίας αυτής περεταίρω, καθώς και η ανάλυση των πρακτικών που χρησιμοποιεί, ώστε να αποτελέσει αφορμή για υιοθέτησή της στα πλαίσια του κέντρου ανάπτυξης πληροφοριακών συστημάτων του στρατού ξηράς, αλλά και γενικότερα κάθε αντίστοιχου οργανισμού.

Στη μελέτη περίπτωσης, όπου εξετάζεται η ανάπτυξη πληροφοριακού συστήματος με τη μεθοδολογία Scrum, επισημαίνονται τα πλεονεκτήματα και μειονεκτήματά της σε πραγματικό περιβάλλον, οπότε προσφέρεται μια συγκριτική αξιολόγηση και ανάλυση της.

2. Βασικές Έννοιες

Στο παρόν κεφάλαιο αναφέρονται οι βασικές έννοιες που είναι απαραίτητες για την κατανόηση των μεθοδολογιών ανάπτυξης των πληροφοριακών συστημάτων.

2.1. Έργο (Project)

Ο ορισμός του έργου, σύμφωνα με τη διεθνή βιβλιογραφία, προτείνεται με αρκετούς τρόπους. Σύμφωνα με έναν από αυτούς και συγκεκριμένα με το εγχειρίδιο του Project Management Institute, PMI, «Έργο είναι ένα προσωρινό εγχείρημα που στοχεύει στην δημιουργία ενός μοναδικού προϊόντος ή υπηρεσίας» (PMBOK, 2004). Επίσης, «η λέξη “έργο (project)” μεταφέρει την ιδέα της επίτευξης ενός συγκεκριμένου στόχου μέσω οργανωμένης σκόπιμης δραστηριότητας» (Βασιλακόπουλος Γ., 2018). Τα έργα διαδραματίζουν σημαντικό ρόλο και μπορούν να καθορίσουν την επιτυχία μίας εταιρίας ή επιχείρησης. Αποτελούν διαδικασίες των οποίων το αποτέλεσμα είναι ένα καινούριο ή τροποποιημένο προϊόν ή υπηρεσία. Τα έργα συμβάλλουν στην αύξηση της παραγωγικότητας και συνεπώς των κερδών της εταιρίας, στην καλύτερη ποιότητα και την ικανοποίηση των πελατών και τον εμπλουτισμό του περιβάλλοντος εργασίας (Kerzner, 2001).

Κάθε έργο, όπως προαναφέραμε είναι προσωρινό, με αρχή και τέλος, και συγκεκριμένο, με την έννοια της μοναδικότητας και της σκοπιμότητας. Φυσικά και κάθε έργο διαφοροποιείται ως προς τον χρόνο υλοποίησής του, το μέγεθος, αλλά και το κόστος.

2.2. Μεθοδολογία

Σύμφωνα με τους Avison και Fitzgerald (David Avison & Guy Fitzgerald, 2006), ο όρος «Μεθοδολογία» δεν υπάρχει συμφωνία ως προς το τι σημαίνει στην παγκόσμια βιβλιογραφία και έτσι χρησιμοποιείται σε μεγάλο βαθμό με ασάφεια. Αυτή η ασαφής χρήση δεν σημαίνει, φυσικά, ότι δεν υπάρχουν ορισμοί, αλλά ότι δεν υπάρχουν ορισμοί που να είναι παγκοσμίως αποδεκτοί. Σε ένα γενικό επίπεδο, «μεθοδολογία» θεωρείται μία σειρά από προτεινόμενα βήματα και διαδικασίες που ακολουθούνται κατά την ανάπτυξη ενός πληροφοριακού

συστήματος, ωστόσο ο ορισμός αυτός δε θεωρείται επαρκής και δεν είναι ευρέως αποδεκτός. Σε μία προσπάθεια ορισμού του όρου «Μεθοδολογία» επεκτείνοντας, ουσιαστικά, τον ορισμό που είχε προτείνει η BCS (British Computer Society) το 1983, οι Avison και Fitzgerald αναφέρουν ότι:

«Μία μεθοδολογία ανάπτυξης συστημάτων είναι το προτεινόμενο μέσο για να επιτευχθεί η ανάπτυξη, ή μέρος της ανάπτυξης, ενός πληροφοριακού συστήματος, και βασίζεται σε ένα σύνολο λογικών εξηγήσεων και μιας βασικής φιλοσοφίας η οποία υποστηρίζει, δικαιολογεί και κάνει κατανοητή τη χρήση του προτεινόμενου μέσου για ένα συγκεκριμένο περιβάλλον. Τα προτεινόμενα μέσα συνήθως περιλαμβάνουν τον ορισμό των φάσεων, των διαδικασιών, των καθηκόντων, των κανόνων, τεχνικών, των κατευθυντηρίων γραμμών, της τεκμηρίωσης και των εργαλείων. Μπορούν, επίσης, να περιλαμβάνουν υποδείξεις σχετικά με τη διαχείριση και οργάνωση της προσέγγισης και εκπαίδευσης των συμμετεχόντων.»

2.3. Κύκλος Ζωής

Συχνά, οι τυποποιημένες διεργασίες ή μεθοδολογίες ανάπτυξης πληροφοριακών συστημάτων (ΠΣ) ονομάζονται “κύκλοι ζωής ανάπτυξης συστημάτων - ΚΖΑΣ (*systems development life cycles - SDLC*)”. Κατά γενική αντίληψη, κάθε κύκλος ζωής είναι ένα χρονικό πλαίσιο που εκτείνεται από την ανάπτυξη ενός νέου συστήματος μέχρι την τελική απόσυρσή του. Είναι μια διεργασία που αρχίζει από μια ιδέα, προχωρά στη συγκεκριμενοποίησή της και στη μετουσίωσή της σε πραγματικό σύστημα και τελειώνει με τον τερματισμό της, αφού έχει διανύσει όλα τα ενδιάμεσα στάδια κατά τα οποία έχει δοθεί ιδιαίτερη έμφαση στους ελέγχους βιωσιμότητας, χρησιμότητας και χρηστικότητας του συστήματος (Βασιλακόπουλος Γ., 2018).

Παρόλο που κάθε κύκλος ζωής διαθέτει τα παραπάνω χαρακτηριστικά, η έννοια του κύκλου ζωής δεν είναι μονοσήμαντη. Τούτο, έχει ως αποτέλεσμα να υπάρχουν πολλά διαφορετικά μοντέλα ή προσεγγίσεις που ανήκουν στην οικογένεια του κύκλου ζωής ανάπτυξης συστημάτων. Επίσης, κάθε έργο πληροφορικής έχει διαφορετικά χαρακτηριστικά και συνεπώς διαφορετικό κύκλο ζωής. Όλες οι εκδοχές, ωστόσο, των κύκλων ζωής έχουν ως κοινή βάση

επίλυσης προβλημάτων τη διάσπαση (*decomposition*) του όλου προβλήματος σε υποπροβλήματα, την ανάλυση (*analysis*) κάθε υποπροβλήματος και τη σύνθεση (*synthesis*) των λύσεων των υποπροβλημάτων, προκειμένου να επιτευχθεί η ολική λύση (δηλαδή, έχουν ως βάση την κοινή μηχανιστική λογική).

3. «Παραδοσιακά» Μοντέλα Ανάπτυξης Λογισμικού

«Η καθιέρωση και χρήση ορθών αρχών μηχανικής, ώστε να κατασκευάσουμε οικονομικά λογισμικό που είναι αξιόπιστο και λειτουργεί αποτελεσματικά σε πραγματικά μηχανήματα» (Bauer, F. et al., 1968), ήταν ένας εκ των πρώτων ορισμών για την Τεχνολογία Λογισμικού που δόθηκε κατά την πρώτη διάσκεψη για την ανάπτυξη λογισμικού το 1968. Ομοίως, ο (Schach, 1990), όρισε την Τεχνολογία Λογισμικού, ως «μία διαδικασία που έχει ως στόχο την παραγωγή ποιοτικού λογισμικού, που παραδίδεται στην ώρα του, στα πλαίσια του προϋπολογισμού, και το οποίο πληροί τις απαιτήσεις του». Παρακάτω αναλύονται οι βασικές μεθοδολογίες ανάπτυξης λογισμικού ξεκινώντας από τις «παραδοσιακές» και καταλήγοντας στις πλέον σύγχρονες.

3.1. Μοντέλο Καταρρακτοειδούς Ανάπτυξης ΠΣ

Το θεμελιώδες, ίσως, μοντέλο για την ανάπτυξη λογισμικού αποτελεί το μοντέλο του «Καταρράκτη» ή «Μεθοδολογία καταρρακτοειδούς ανάπτυξης πληροφοριακών συστημάτων (Waterfall Systems Development - WSD)». Το μοντέλο αυτό υπαγορεύει μια διεργασία ανάπτυξης ΠΣ κατά την οποία εκτελούνται σειριακά διάφορες, διακριτές μεταξύ τους, φάσεις. Οι φάσεις αυτές είναι επιγραμματικά οι κάτωθι:

- Ανάλυση και Προσδιορισμός Απαιτήσεων (*Requirements Analysis & Specification*),
- Σχεδιασμός Συστήματος (*System Design*),
- Πραγμάτωση και Έλεγχος Ενοτήτων (*Implementation & Unit Testing*),
- Ολοκλήρωση και Έλεγχος Συστήματος (*System Integration & Testing*), και
- Εγκατάσταση, Λειτουργία & Συντήρηση (*Deployment, Operation & Maintenance*).

Στην πρώτη φάση, γίνεται λεπτομερής προσδιορισμός των λειτουργικών και τεχνικών απαιτήσεων, των απαιτήσεων ασφάλειας και των απαιτήσεων δεδομένων του συστήματος, των επιδόσεών του (π.χ. αριθμός ταυτόχρονων χρηστών, ανοχή παύσης λειτουργίας, ταχύτητας λειτουργίας, ποιότητας

δεδομένων), των εισροών και εκροών δεδομένων και πληροφοριών, των επεξεργασιών και των αποτελεσμάτων. Σ' αυτή τη φάση γίνεται ανάλυση της διαδικασίας και προσδιορίζονται οι λειτουργικές και πληροφοριακές απαιτήσεις του συστήματος, καθώς και οι απαιτήσεις ασφάλειας πληροφοριών.

Εν συνεχεία, στη δεύτερη φάση, αυτή του *σχεδιασμού του συστήματος*, καθορίζεται η αρχιτεκτονική του συστήματος, διακρίνοντας τις απαιτήσεις του αναφορικά με το υλικό (*hardware*), το δίκτυο (*network*), το λογισμικό συστήματος (*system software*) και το λογισμικό εφαρμογών (*application software*).

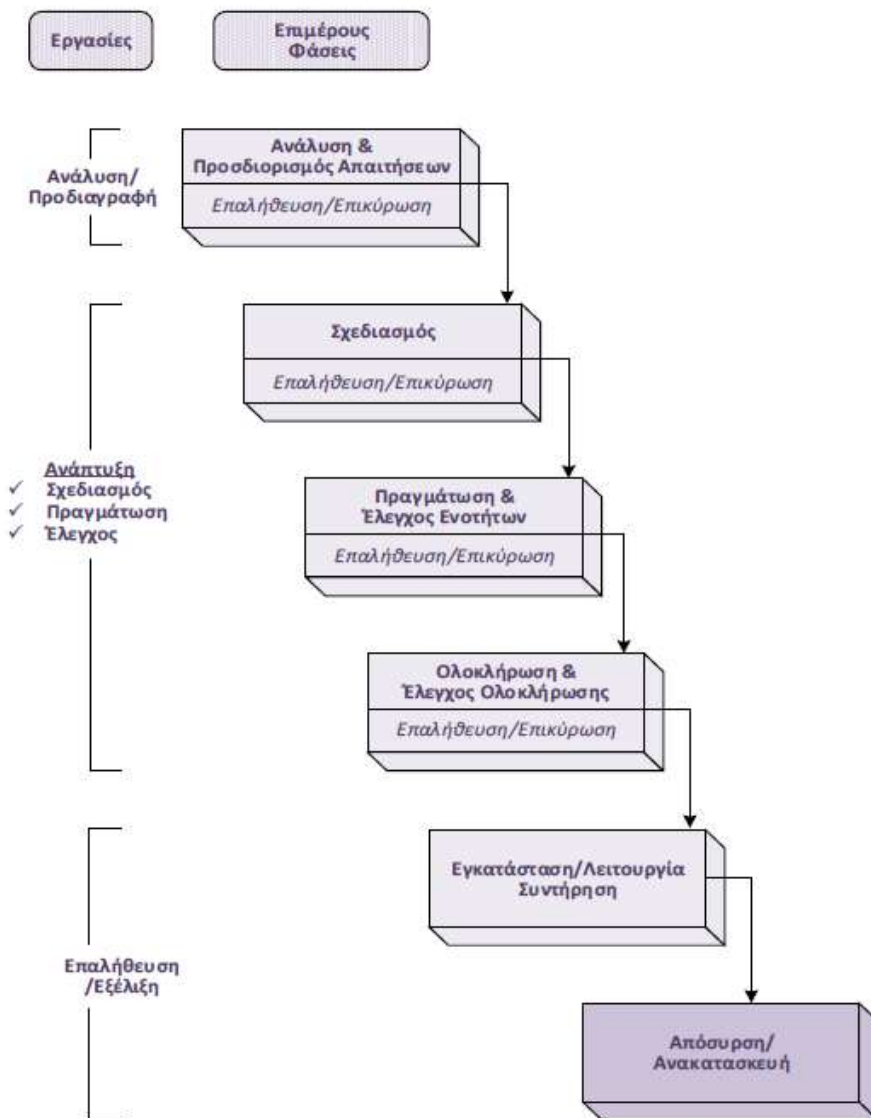
Στην τρίτη φάση της *πραγμάτωσης και ελέγχου ενοτήτων*, κατασκευάζεται το σύνολο των προγραμμάτων της εφαρμογής ή των εφαρμογών, σύμφωνα με τις προδιαγραφές που έχουν ορισθεί. Κάθε επιμέρους εφαρμογή ελέγχεται με δοκιμές μονάδων (Unit tests) ως προς την τήρηση των προδιαγραφών που έχουν προκαθορισθεί.

Στην προτελευταία, τέταρτη φάση της μεθοδολογίας ολοκληρώνονται και συνενώνονται, πιθανώς, όλες οι εφαρμογές μεταξύ τους, ώστε να παραδοθεί ένα ολοκληρωμένο σύστημα (συνεκτικό και διαλειτουργικό), το οποίο στην συνέχεια ελέγχεται αναφορικά με την τήρηση των προδιαγραφών που έχουν τεθεί, καθώς και τα τεχνικά και λειτουργικά του χαρακτηριστικά. Το σύστημα είναι, πλέον, έτοιμο να παραδοθεί στον χρήστη και να ξεκινήσει η τελευταία φάση της μεθοδολογίας.

Η τελευταία φάση προβλέπει την *εγκατάσταση, λειτουργία και συντήρηση* του συστήματος στο παραγωγικό περιβάλλον και τίθεται σε δοκιμαστική λειτουργία. Με την περάτωση όλων των προπαρασκευαστικών ενεργειών, όπως η μετάπτωση δεδομένων, το σύστημα τίθεται σε παραγωγική λειτουργία. Αμέσως μετά, ενεργοποιείται η διαδικασία της συντήρησης, η οποία δε σταματά ποτέ ουσιαστικά.

Το καταρρακτοειδές μοντέλο είναι, ίσως, το πιο διαδεδομένο για περιπτώσεις όπου οι απαιτήσεις του συστήματος είναι εξαρχής γνωστές και σαφείς (καλά δομημένη προβληματική κατάσταση) και, επιπλέον, δε μεταβάλλονται κατά τη διεργασία ανάπτυξής του ΠΣ. Τούτο διότι, σε σύγκριση με άλλα μοντέλα, δεν

επιτρέπει την επανάληψη φάσεων, αλλά επιβάλλει τη σειριακή εκτέλεσή τους για την ανάπτυξη του συστήματος. Με άλλα λόγια, προκειμένου να ξεκινήσει η εκτέλεση μίας φάσης, θα πρέπει να έχει περατωθεί η εκτέλεση της προηγούμενης, να έχουν γίνει, δηλαδή, αποδεκτά τα παραδοτέα της φάσης, έπειτα από αξιολόγηση, πριν προχωρήσουμε στην επόμενη.



Εικόνα 1. Καταρακτοειδές Μοντέλο (Βασιλακόπουλος Γ., 2018)

Το κύριο χαρακτηριστικό της μεθοδολογίας είναι η διεξαγωγή αυστηρότατου ελέγχου καθ' όλη τη διάρκεια του έργου, ο οποίος πραγματοποιείται με τη

χρήση λεπτομερούς γραπτής τεκμηρίωσης (τα παραδοτέα), καθώς επίσης και από επίσημες αναθεωρήσεις που εγκρίνονται από τον τελικό πελάτη.

Συμπερασματικά, υπάρχουν κάποιες θεμελιώδεις αρχές πάνω στις οποίες στηρίζεται το παραδοσιακό καταρρακτοειδές μοντέλο, σύμφωνα με τον Βασιλακόπουλο. Αυτές είναι οι ακόλουθες:

- Η διεργασία ανάπτυξης του συστήματος είναι καθαρά σειριακή, δηλαδή εκτελείται μια ακολουθία από σαφώς καθορισμένα και διαδοχικά βήματα.
- Κάθε βήμα καταλήγει στη δημιουργία ενός, τουλάχιστον, παραδοτέου (έγγραφο ή κώδικας) το οποίο κοστολογείται.
- Κάθε παραδοτέο συνιστά το υπόβαθρο για την εκτέλεση του επομένου βήματος και το υπόβαθρο για την ολική κοστολόγηση του έργου.
- Η πληρότητα και ορθότητα κάθε παραδοτέου μπορεί να ελεγχθεί μέσω μιας συγκεκριμένης και προκαθορισμένης διεργασίας ελέγχου (δοκιμών).

Όπως είναι φανερό, από τις παραπάνω αρχές, το μοντέλο στοχεύει στην εξασφάλιση της ποιότητας, της αξιοπιστίας και της μελλοντικής συντήρησης του λογισμικού προσβλέποντας ταυτόχρονα και στην καλύτερη δυνατή εκμετάλλευση των πόρων. Ωστόσο, αυτά τα θετικά του χαρακτηριστικά, λειτουργούν παράλληλα και ως αδυναμίες του. Το μοντέλο είναι άκαμπτο και κοστίζει αρκετά, καθώς η ανάπτυξη του λογισμικού προχωρά με αργούς ρυθμούς, ακριβώς εξαιτίας των μεγάλων, αυστηρών και λεπτομερών παραδοτέων και ελέγχων. Επιπροσθέτως, βασίζεται σε μεγάλο βαθμό στην ορθή και λεπτομερή καταγραφή των απαιτήσεων. Συχνά, ο πελάτης αδυνατεί να καθορίσει σαφώς τις απαιτήσεις του συστήματος από την αρχή της ανάπτυξής του. Συνεπώς, ελλοχεύει ο κίνδυνος εντοπισμού ασαφειών και μη προβλεπόμενων αναγκών κατά τη φάση της πραγμάτωσης του συστήματος. Η απόπειρα διόρθωσης τέτοιων προβλημάτων, που εμφανίζονται δηλαδή αργά στον κύκλο ζωής της ανάπτυξης του έργου, συνεπάγεται μεγάλο κόστος, καθώς η ενσωμάτωση των αλλαγών είναι εξαιρετικά δύσκολη. Αυτή η “ακαμψία” του βασικού καταρρακτοειδούς μοντέλου είναι αντικείμενο συχνής αρνητικής κριτικής και προκαλεί τη στροφή των κατασκευαστών σε άλλα πιο “ευέλικτα”

μοντέλα ανάπτυξης ΠΣ. Τέλος, όπως προαναφέρθηκε, υπάρχει πολλή γραπτή τεκμηρίωση για την οποία δαπανείται μεγάλος σχετικά χρόνος.

3.2. Δομημένη Ανάλυση και Σχεδιασμός Συστημάτων (SSADM)

Η μεθοδολογία της Δομημένης Ανάλυσης και Σχεδιασμού Συστημάτων (*Structured Systems Analysis and Design Methodology – SSADM*) είναι μία ιδιαίτερα δομημένη μεθοδολογία, η οποία ακολουθεί κατά βάση το καταρρακτοειδές μοντέλο. Η SSADM αποτελείται από επτά (7) στάδια μέσα σε ένα πλαίσιο πέντε (5) ενοτήτων με το δικό τους σύνολο στόχων, χρονοδιαγραμμάτων, ελέγχων και παρακολούθησης διαδικασιών (David Avison & Guy Fitzgerald, 2006).

- Μελέτη σκοπιμότητας (ή εφικτότητας)
 - Σκοπιμότητα
- Ανάλυση προδιαγραφών
 - Διερεύνηση υπάρχοντος περιβάλλοντος
 - Επιλογές επιχειρησιακών συστημάτων
- Προδιαγραφές απαιτήσεων
 - Καθορισμός απαιτήσεων
- Προδιαγραφές λογικού συστήματος
 - Τεχνικές επιλογές συστήματος
 - Λογικός σχεδιασμός
- Φυσικός σχεδιασμός
 - Φυσικός σχεδιασμός

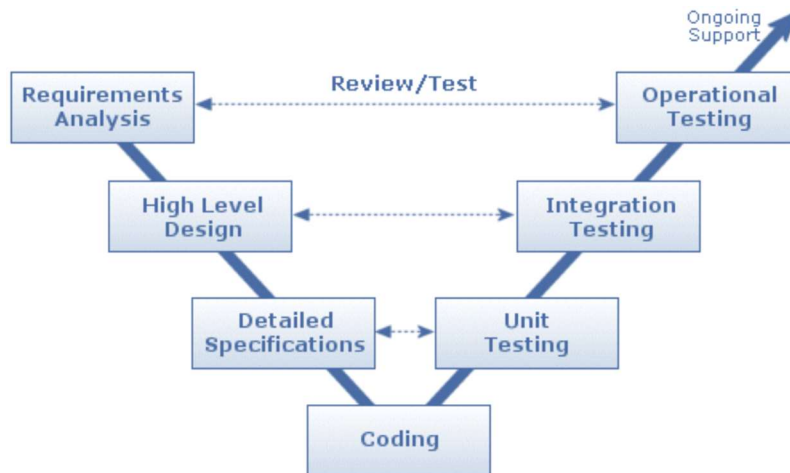
Η συγκεκριμένη μεθοδολογία αποτελεί μία βάση σύγκρισης και αξιολόγησης για άλλες μεθοδολογίες. Η ΔΑΣΣ εκτελεί κάθε βήμα της διεργασίας ανάπτυξης ενός συστήματος στη βάση του προηγούμενου βήματος χωρίς καμία παρέκκλιση. Τούτο, σε αντιστοιχία με το καταρρακτοειδές μοντέλο, έχει ορισμένα πλεονεκτήματα και μειονεκτήματα. Το κυριότερο εκ των πλεονεκτημάτων είναι ότι λόγω αυτής της μη παρέκκλισης που αναφέρθηκε προηγουμένως, ουσιαστικά πρέπει να προσδιοριστούν με πληρότητα και σαφήνεια εξ αρχής οι απαιτήσεις του συστήματος, πριν τη φάση του σχεδιασμού και της πραγμάτωσης. Αποδεικνύεται, ωστόσο, στην πράξη ότι κάτι τέτοιο είναι

εξαιρετικά δύσκολο. Ως κυριότερο μειονέκτημα της μεθοδολογίας θα πρέπει να αναφέρουμε τη δυσκαμψία και την αυστηρότητα που καθιστούν αδύνατη την ενσωμάτωση τροποποιήσεων ή πρόσθετων απαιτήσεων που μπορεί να προκύψουν κατά τη διάρκεια ή αργότερα της διεργασίας ανάπτυξης. Τούτο, διότι δεν προβλέπεται η δυνατότητα επανάληψης προηγούμενων φάσεων και ως λύση προβλέπεται μόνο η εξ αρχής επανάληψη της όλης διεργασίας, καθιστώντας τη μεθοδολογία μη επιδεκτική σε τυχόν αλλαγές.

Η ΔΑΣΣ (SSADM) είναι παραδοσιακά συνδεδεμένη με καλά καθορισμένα, μεγάλης κλίμακας αναπτυξιακά έργα πληροφοριακών συστημάτων που απαιτούν βαθιά τεκμηρίωση, για χρήση σε ιδιαίτερα μεγάλους γραφειοκρατικούς οργανισμούς (David Avison & Guy Fitzgerald, 2006).

3.3. V – Μοντέλο

Το μοντέλο V υιοθετεί αρκετά εκ των χαρακτηριστικών του καταρρακτοειδούς μοντέλου. Πήρε το όνομά του από το γράμμα «V» του λατινικού αλφαβήτου, καθώς η όλη διαδικασία ακολουθεί μία καθοδική κλίση και κατόπιν μια ανοδική. Η μεθοδολογία V προβλέπει την πραγμάτωση κάθε βήματος (φάσης) με βάση την τεκμηρίωση του προηγούμενου βήματος (φάσης), προκειμένου να ελέγχονται και να εγκρίνονται τα παραδοτέα κάθε φάσης της διεργασίας πριν αυτή δρομολογηθεί προς την επόμενη φάση, όπως ακριβώς και στην καταρρακτοειδή. Η ουσιαστική διαφορά μεταξύ τους είναι ότι, αντί να συνεχίζεται αυτή η καθοδική κίνηση από μια φάση στην άλλη υπάρχει μια παράλληλη δομή που κινείται ανοδικά προσδίδοντας στο μοντέλο το διακριτό σχήμα V. Η ανοδική πορεία περιγράφει τα βήματα δοκιμών (ελέγχων) που έπονται της κωδικοποίησης, αρχίζοντας με τις δοκιμές μονάδων (unit testing) και τελειώνοντας με τις δοκιμές αποδοχής (acceptance testing), το τελευταίο βήμα πριν την παράδοση/παραλαβή του συστήματος.



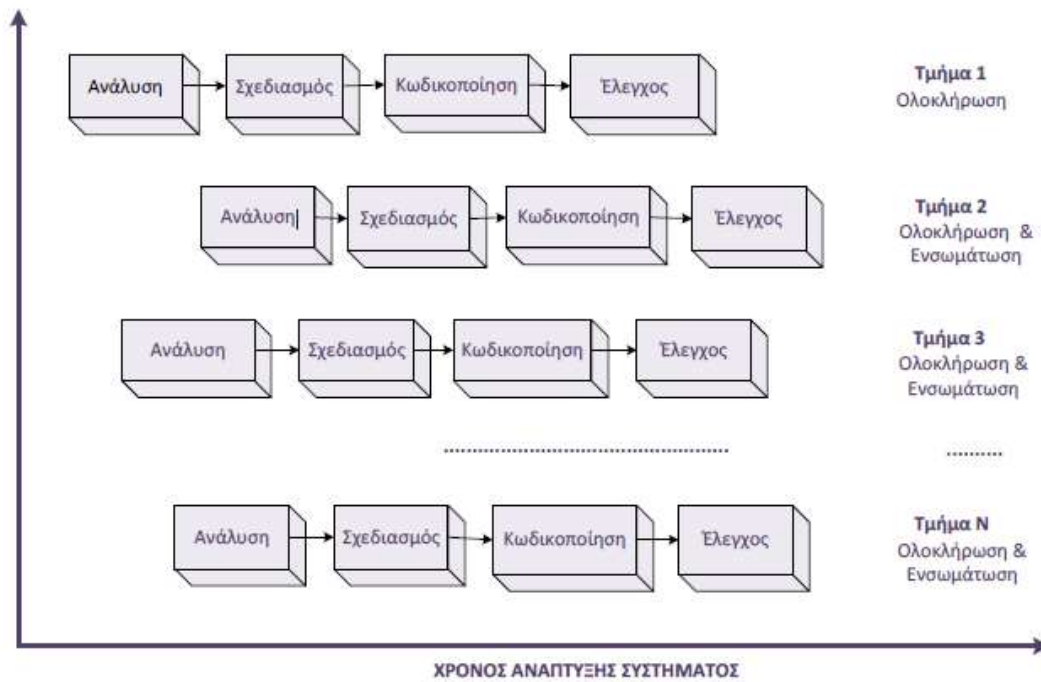
Εικόνα 2. Το V μοντέλο

Είναι φανερό και σε αυτή την μεθοδολογία, ότι τα τυχόν λάθη αναγνωρίζονται αργά στον κύκλο ζωής του έργου, με την έννοια της αυστηρότητας της διεργασίας και της παροχής ελάχιστου χώρου για ευέλικτη προσαρμογή. Ωστόσο, το μεγαλύτερο πλεονέκτημα για τη χρήση του μοντέλου σε σύγκριση με το κλασικό καταρακτοειδές μοντέλο, είναι το γεγονός ότι επειδή η μεθοδολογία V προσβλέπει στη διόρθωση των σφαλμάτων του συστήματος αμέσως μόλις εντοπιστούν (πριν την εγκατάσταση και λειτουργία του συστήματος), μειώνεται το κόστος συντήρησης.

3.4. Μεθοδολογία Λειτουργικής Επαύξεσης

Έχουν προταθεί διάφορα μοντέλα που συνδυάζουν γραμμικές και επαναληπτικές μεθοδολογίες ανάπτυξης λογισμικού με αρχικό στόχο κάθε ενός από αυτά να μειωθεί όσο το δυνατόν περισσότερο ο υπαρκτός κίνδυνος αποτυχίας του συστήματος, με τη διαίρεση του σε μικρότερα μέρη και την παροχή ευελιξίας στην αλλαγή κατά τον κύκλο ανάπτυξης του έργου (Pfleeger, 2003). Η μεθοδολογία λειτουργικής επαύξεσης (Functional Incrementation) στοχεύει στη θεραπεία, κατά το δυνατόν, των αδυναμιών του καταρακτοειδούς μοντέλου αναφορικά με τη σειριακή, χωρίς επαναλήψεις, διεργασία ανάπτυξης του ΠΣ (Βασιλακόπουλος Γ., 2018). Η μεθοδολογία προβλέπει την κατάτμηση του συστήματος σε υποσυστήματα που αντιστοιχούν σε λειτουργικές μονάδες. Οι εφαρμογές που αντιστοιχούν στα επί μέρους τμήματα αναπτύσσονται

ανεξάρτητα και παράλληλα. Όταν ολοκληρωθεί η ανάπτυξη της εφαρμογής που αντιστοιχεί σε κάποιο τμήμα, συνενώνεται με τις προηγουμένως αναπτυχθείσες εφαρμογές και ενσωματώνεται στο όλο σύστημα. Η όλη διαδικασία συνεχίζεται μέχρι να αναπτυχθεί και να ενσωματωθεί και το τελευταίο τμήμα στο σύστημα.



Εικόνα 3. Μεθοδολογία Λειτουργικής Επαύξεσης (Βασιλακόπουλος Γ., 2018)

Τα κύρια πλεονεκτήματα της μεθοδολογίας είναι ότι, αρχικώς, παρέχεται η δυνατότητα παράλληλης ανάπτυξης των επιμέρους εφαρμογών του συστήματος με αποτέλεσμα τη συντόμευση της διαδικασίας ανάπτυξης του. Επίσης, το σύστημα κατασκευάζεται σταδιακά και εμπλουτίζεται διαρκώς με νέες εφαρμογές που εντάσσονται στον κορμό του, που έχει ως αποτέλεσμα την πληρέστερη κατανόηση των απαιτήσεων για κάθε τμήμα από τις ομάδες ανάπτυξης.

Φυσικά, η μεθοδολογία λειτουργικής επαύξεσης έχει και αυτή μειονεκτήματα. Το πιο σημαντικό είναι η δυσκολία κατάτμησης ενός οργανισμού σε λειτουργικά τμήματα ή/και επιχειρησιακές διεργασίες, καθώς και ο ορισμός των αλληλεξαρτήσεων μεταξύ τους, ειδικά αν δεν υπάρχει εξοικείωση για κάτι τέτοιο. Ως αποτέλεσμα της μη ορθής διάσπασης είναι η μη σωστή κατάτμηση

και του ΠΣ σε επιμέρους εφαρμογές. Θα πρέπει να αναφέρουμε επίσης ότι και σε αυτή τη μεθοδολογία οι αρχικές απαιτήσεις θα πρέπει να είναι γνωστές στο σύνολο τους και να μη μεταβάλλονται σημαντικά κατά τη διάρκεια ανάπτυξης, δεδομένου ότι σε κάθε επιμέρους εφαρμογή (τμήμα) ακολουθείται η καταρακτοειδής μεθοδολογία.

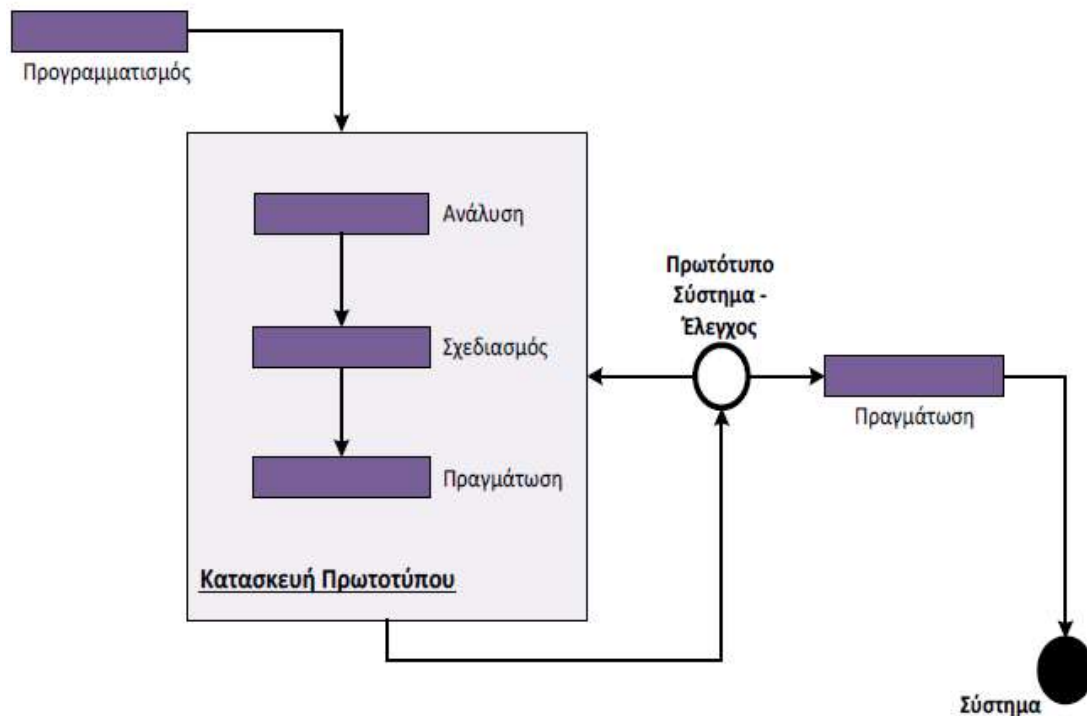
3.5. Μοντέλο Δημιουργίας Πρωτοτύπου

Η μεθοδολογία της δημιουργίας πρωτοτύπου (Prototyping model) είναι ένα μοντέλο ανάπτυξης λογισμικού, κατά το οποίο κατασκευάζονται σε τμήματα μη ολοκληρωμένες εκδόσεις του ΠΣ που πρόκειται να αναπτυχθεί. Κάθε μία από αυτές τις εκδόσεις ονομάζεται πρωτότυπο, ενώ περιέχει κάποια μόνο από τα χαρακτηριστικά και τις λειτουργίες του τελικού λογισμικού, ενώ είναι πιθανό το μοντέλο να είναι πολύ διαφορετικό από το τελικό προϊόν.

Η παραπάνω μεθοδολογία επιτρέπει στους τελικούς χρήστες να αξιολογήσουν τις προτάσεις της ομάδας ανάπτυξης, δοκιμάζοντας ένα πρωτότυπο του προϊόντος αντί της διαδικασίας της περιγραφής και των προδιαγραφών από μέρους τους. Με αυτό επιτυγχάνεται ο προσδιορισμός τυχόν ασαφειών στις απαιτήσεις του ΠΣ.

Κάθε πρωτότυπο είναι μία λειτουργική έκδοση του ΠΣ και θεωρείται προκαταρκτικό μοντέλο. Το μοντέλο αλλάζει συνεχώς και βελτιώνεται έως ότου τύχει της πλήρους αποδοχής από τους πελάτες. Με την τελική αποδοχή μετατρέπεται αυτομάτως σε τελικό ΠΣ. Η μέθοδος είναι μια επαναληπτική διαδικασία με καθορισμένα βήματα, όπου κάθε νέα έκδοση του πρωτοτύπου προσεγγίζει ακριβέστερα τις απαιτήσεις των τελικών χρηστών. Τα βήματα είναι τα εξής:

- Εντοπισμός και καταγραφή των βασικών απαιτήσεων του χρήστη.
- Γρήγορη ανάπτυξη λειτουργικού πρωτοτύπου.
- Δοκιμή του πρωτοτύπου (από τους τελικούς χρήστες).
- Διόρθωση και βελτίωση πρωτοτύπου. Επιστροφή στο προηγούμενο βήμα μέχρι τελικής ικανοποίησης των χρηστών.
- Μετατροπή πρωτοτύπου σε τελικό προϊόν.



Εικόνα 4. Μεθοδολογία Δημιουργίας Πρωτοτύπων (Βασιλακόπουλος Γ., 2018)

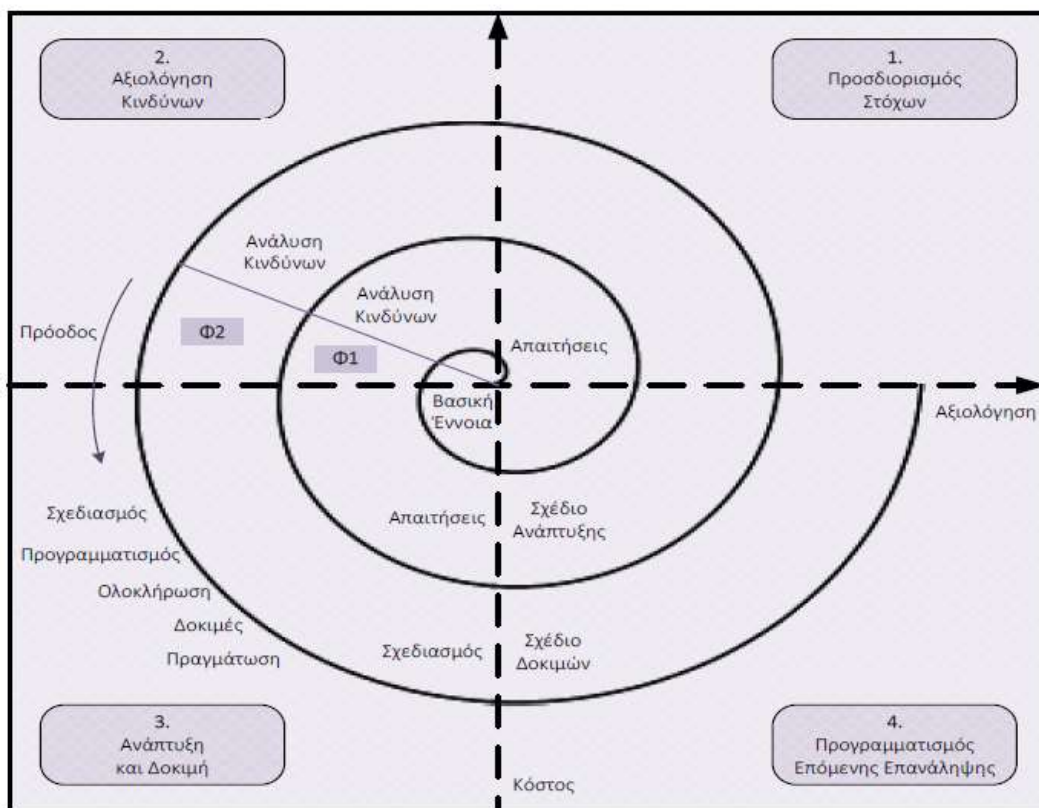
Τα κύρια πλεονεκτήματα της μεθοδολογίας είναι ότι ουσιαστικά δε διαγράφεται η καταβληθείσα προσπάθεια για την κατασκευή των πρωτοτύπων, καθώς επίσης και ότι επιταχύνεται η κατασκευή σε σύγκριση, κυρίως, με το καταρακτοειδές μοντέλο. Επίσης, με τη συμμετοχή των χρηστών από το αρχικό στάδιο ανακαλύπτονται σε αρκετά πρώιμο χρόνο προβλήματα και καθιστούν ευκολότερη την επίλυση τους, μειώνοντας έτσι τον κίνδυνο για αποτυχία του τελικού ΠΣ. Τέλος, η μεθοδολογία αυτή είναι κατάλληλη για μη καλά ορισμένες απαιτήσεις.

Τα κύρια μειονεκτήματά της είναι ουσιαστικά ότι, λόγω των συνεχών αλλαγών, δεν είναι δυνατή η παραγωγή υλικού τεκμηρίωσης, ενώ ταυτόχρονα δυσχεραίνεται και η συντήρηση του συστήματος. Επίσης, ελλοχεύει ο κίνδυνος, από τυχόν ανεπάρκεια στην ανάλυση του υφισταμένου προβλήματος, να οδηγηθούμε σε κάλυψη μόνο ορισμένων επιφανειακών αναγκών του συστήματος με αποτέλεσμα να πραγματοποιηθεί ένα ημιτελές και όχι εύχρηστο ΠΣ.

Τέλος, οι συνεχείς επαναλήψεις κοστίζουν και ο προϋπολογισμός του έργου δε δύναται να καθοριστεί εξ' αρχής.

3.6. Σπειροειδές Μοντέλο

Το 1988 ο Barry Boehm παρουσίασε το σπειροειδές μοντέλο. Η μεθοδολογία αυτή δεν ήταν η πρώτη που εφάρμοζε την επαναληπτική ανάπτυξη αλλά ήταν η πρώτη που εξηγούσε γιατί ήταν τόσο σημαντική. Η βασική διαφορά του σπειροειδούς μοντέλου από το επαυξητικό είναι ότι πλέον ορίζεται “ο κύκλος (η επανάληψη)” ως η βάση για την προσθήκη αναβαθμισμένων λειτουργικών χαρακτηριστικών του συστήματος. Επίσης, η καινοτομία που προσέφερε το σπειροειδές μοντέλο είναι ότι πριν από κάθε κύκλο (επανάληψη) διενεργείται ανάλυση κινδύνων αναφορικά με την επικείμενη εκτέλεση του κύκλου αυτού. Το όνομα της μεθοδολογίας κληρονομείται από τη γραφική του απεικόνιση, όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 5. Σπειροειδές μοντέλο

Τα βασικά χαρακτηριστικά του μοντέλου σύμφωνα με τον δημιουργό του (Boehm B, 1998) είναι τα εξής: Αρχικώς, το ότι δίνεται έμφαση στην αξιολόγηση του κινδύνου και στην προσπάθεια να ελαχιστοποιηθεί. Τούτο προκύπτει από την κατάτμηση του έργου σε υποέργα, όπως ακριβώς και στη μεθοδολογία πρωτοτύπων. Για τον λόγο αυτό, η ανάπτυξη χωρίζεται σε πολλούς κύκλους σε κάθε έναν από τους οποίους προστίθενται νέα λειτουργικά χαρακτηριστικά στο σύστημα. Κάθε κύκλος αρχίζει με τον προσδιορισμό των στόχων και των απαιτήσεων του τρέχοντος σταδίου ανάπτυξης του συστήματος και με μια ανάλυση των εναλλακτικών λύσεων και των περιορισμών.

Η μεθοδολογία προσβλέπει στη διαρκή βελτίωση του πρωτοτύπου, καθώς μειώνονται οι κίνδυνοι (και, πιθανώς, εμφανίζονται άλλοι). Καθώς το πρωτότυπο γίνεται αποδεκτό, και οι κίνδυνοι μειώνονται σε αποδεκτά επίπεδα, εκτελείται το επόμενο βήμα σύμφωνα με το βασικό καταρρακτοειδές μοντέλο με τη γνωστή αλληλουχία των φάσεων. Μόλις περατωθεί αυτός ο κύκλος, αρχίζει ο επόμενος, καθώς δημιουργείται μια νέα, επαυξημένη έκδοση του συστήματος. Επαναλαμβάνεται ότι η αξιολόγηση κινδύνων συνιστά μια σημαντική διαφορά από το επαυξητικό μοντέλο.

Οι βασικές φάσεις σε αυτό το μοντέλο είναι οι κάτωθι :

- *Προσδιορισμός στόχων*, λύσεων καθώς και περιορισμών της επανάληψης. Επίσης, στη φάση αυτή καθορίζονται οι εργασίες κάθε επανάληψης και ο εντοπισμός των κινδύνων της διαδικασίας.
- *Διαχείριση κινδύνων*, που αφορά στη λεπτομερή ανάλυση των ενδεχόμενων κινδύνων, στην αξιολόγηση κάθε εναλλακτικής λύσης και στη λήψη απόφασης για συνέχιση της μιας κατεύθυνσης, ενώ λαμβάνεται και η απόφαση για κατασκευή ή όχι πρωτοτύπων, κ.τ.λ.
- *Εκτέλεση διεργασιών ανάπτυξης του συστήματος*, που αναφέρεται στην εκτέλεση των βημάτων που έχουν αποφασιστεί από την προηγούμενη φάση για το τμήμα του συστήματος που αφορά η τρέχουσα επανάληψη της σπειροειδούς διεργασίας. Στη συνέχεια, αφού γίνει η επικύρωση των αποτελεσμάτων του τμήματος λογισμικού που αναπτύχθηκε, προγραμματίζεται η συνέχιση της ανάπτυξης.

Το μεγαλύτερο πλεονέκτημα της μεθοδολογίας είναι ότι δίνεται μεγάλη έμφαση στην ανάλυση των κινδύνων, ιδιότητα που την καθιστά ιδανική για μεγάλα έργα με σημαντική κρισιμότητα. Τούτο, επιφέρει επιτάχυνση της ανάπτυξης του συστήματος καθώς πραγματοποιείται διαρκής μείωση των κινδύνων του έργου. Αντιθέτως, δεν είναι το ίδιο αποτελεσματική για μικρότερα έργα, διότι η διαδικασία της εκτίμησης των κινδύνων μπορεί να επιφέρει σημαντική αύξηση στο κόστος κατασκευής σε τέτοιο βαθμό που η κατασκευή του συστήματος μπορεί να αποδειχθεί αναποτελεσματική ανεξάρτητα των κινδύνων. Τέλος, πρέπει να τονισθεί ότι για την επιτυχία της μεθοδολογίας απαιτείται εξαιρετικά ικανή ομάδα στην ανάλυση και διαχείριση των κινδύνων προκειμένου να εμφανιστούν θετικά αποτελέσματα.

3.7. Συνοψίζοντας

Πολλές διαφορετικές μεθοδολογίες έχουν προταθεί στηριζόμενες στο «παραδοσιακό» καταρακτοειδές μοντέλο. Όλες οι προσεγγίσεις, οι οποίες χαρακτηρίζονται και «βαριές», είναι προσανατολισμένες στη βαριά τεκμηρίωση και στις αυστηρά προκαθορισμένες διαδικασίες. Με το πέρασμα του χρόνου και κυρίως με την εφαρμογή τους «στη πράξη» έχουν κατηγορηθεί ως μη παραγωγικές και γραφειοκρατικές. Τούτο, διότι όπως αναφέρθηκε και στις προηγούμενες παραγράφους, αποτελούνται από πολλές φάσεις, μεγάλης διάρκειας, αφού προσπαθούν ουσιαστικά να μαντέψουν και να προλάβουν κάθε πιθανό πρόβλημα που μπορεί να προκύψει κατά την ανάπτυξη του ΠΣ. Αυτή είναι ουσιαστικά και η μεγαλύτερή τους αδυναμία που τις καθιστά ανελαστικές.

4. Ευέλικτες (Agile) Μεθοδολογίες Ανάπτυξης Λογισμικού

Στο παρόν κεφάλαιο αναφέρονται τα γενικά χαρακτηριστικά των ευέλικτων μεθοδολογιών. Εν συνεχεία, περιγράφονται οι σημαντικότερες εξ αυτών και τέλος παρουσιάζονται τα βασικά κριτήρια για επιλογή τους ή μη προς ανάπτυξη ενός πληροφοριακού συστήματος.

4.1. Γενικά Χαρακτηριστικά

Ο στόχος της γρήγορης ανάπτυξης εφαρμογών απασχολεί έντονα τις επιχειρήσεις τα τελευταία χρόνια. Οι αλλαγές στις επιχειρηματικές ανάγκες έχουν ως αποτέλεσμα την ανάγκη να αναπτυχθούν πληροφοριακά συστήματα πιο γρήγορα απ' ό,τι ίσως στο παρελθόν. Το γενικό επιχειρηματικό περιβάλλον γίνεται όλο και πιο ανταγωνιστικό, περισσότερο επικεντρωμένο στον πελάτη και απευθύνεται σε διεθνές επίπεδο.

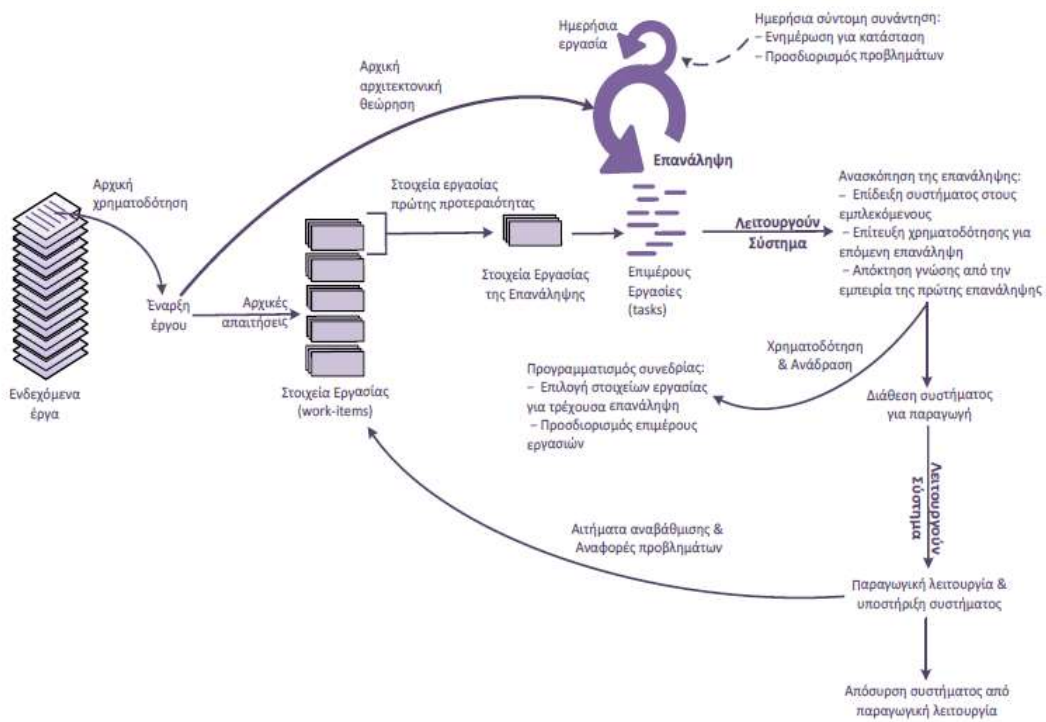
Η ταχεία ανάπτυξη συστημάτων – ΤΑΣ (Rapid Application Development – RAD) προσδιορίστηκε ως μια προσέγγιση στη δημιουργία και τη συντήρηση των συστημάτων που βασίζονται στους υπολογιστές, που συνδυάζει την αποδοτική χρήση εργαλείων και τεχνικών για παράδοση έργων σε στενές χρονικές κλίμακες (David Avison & Guy Fitzgerald, 2006). Η διαδικασία της ανάπτυξης του συστήματος υπαγορεύει τη γρήγορη ανάπτυξη κάποιου μέρους του συστήματος και την παράδοσή του στους χρήστες προς υποβολή παρατηρήσεων προκειμένου να κατασκευαστεί η επόμενη έκδοση του συστήματος που θα τις ικανοποιεί. Κάποιες εκ των παραλλαγών της ΤΑΣ προσπαθούν να προσαρμόζονται σε πιθανές τροποποιήσεις των επιχειρησιακών διεργασιών με εκτέλεση όλων των φάσεων της ανάπτυξης του συστήματος σχεδόν ταυτόχρονα. Μία ομάδα μεθοδολογιών που βασίστηκε σε αυτήν την αρχή και εδραιώθηκε στις μέρες μας είναι οι μεθοδολογίες ευέλικτης ανάπτυξης συστημάτων (Agile Systems Development Methodologies).

Το 2001, για πρώτη φορά γίνεται η ιδρυτική διακήρυξη Agile Manifesto (Agile Alliance, 2001) με την ονομασία «ευέλικτες μέθοδοι» (agile methods) . Υπόσχεται, σε αντίδραση, ενάντια στην παραδοσιακή καταρρακτοειδή

μεθοδολογία, ανταπόκριση στις αλλαγές, παραγωγικότερες πρακτικές και λιγότερη γραφειοκρατία. Η ονομασία «ευέλικτες» αναφέρεται ουσιαστικά στην προσαρμοστικότητα της στις αλλαγές στη διαδικασία ενός έργου. Στη βιβλιογραφία υπάρχουν πολλοί ορισμοί της ευέλικτης μεθοδολογίας. Σύμφωνα με μία εξ αυτών, ευέλικτη ονομάζεται μια μεθοδολογία ανάπτυξης συστημάτων λογισμικού η οποία είναι:

- ανθρωποκεντρική και επικοινωνιοστρεφής, εστιάζει, δηλαδή στην επικοινωνία μεταξύ των εμπλεκόμενων στην ανάπτυξη του συστήματος ανθρώπων (π.χ. κατασκευαστών και χρηστών),
- ευέλικτη (είναι έτοιμη για προσαρμογή σε προβλέψιμες αλλαγές του συστήματος σε οποιονδήποτε χρόνο κατά τη διάρκεια της ανάπτυξης ή λειτουργίας του),
- ταχεία (ενθαρρύνει τη γρήγορη και επαναληπτική ανάπτυξη του συστήματος σε μικρές εκδόσεις),
- εύλογη (εστιάζει στη μείωση του χρόνου και του κόστους κατασκευής του συστήματος με παράλληλη βελτίωση της ποιότητάς του),
- αποκριτική (ανταποκρίνεται κατάλληλα σε αναμενόμενες και μη αναμενόμενες αλλαγές του συστήματος), και
- μαθησιακή (εστιάζει στη διαρκή βελτίωση του συστήματος, και κατά τη διαδικασία ανάπτυξής του και κατόπιν αυτής).

Οι ευέλικτες μεθοδολογίες, γενικώς, προωθούν μια πειθαρχημένη διεργασία διοίκησης έργων ανάπτυξης συστημάτων η οποία ενθαρρύνει τον προσανατολισμό στην ομαδική εργασία, τη διαρκή προσαρμογή και επιθεώρηση, την αυτό-οργάνωση και λογοδοσία, την εναρμόνιση με τις ανάγκες των χρηστών και τους στόχους και τη στρατηγική του οργανισμού και τέλος την υιοθέτηση ενός συνόλου καλών πρακτικών μηχανευτικής συστημάτων που επιτρέπουν τη γρήγορη ανάπτυξη λογισμικού υψηλής ποιότητας.



Εικόνα 6. Γενική Μεθοδολογία Ευέλικτης Ανάπτυξης Συστημάτων (Βασιλακόπουλος Γ., 2018)

Οι σημαντικότερες ευέλικτες μεθοδολογίες είναι οι εξής: το δυναμικό μοντέλο ανάπτυξη συστημάτων (Dynamic Systems Development Model – DSDM) (Stapleton, 1997), η Scrum (Schwaber K. a., 2001) και η μεθοδολογία του ακραίου προγραμματισμού (Extreme Programming – XP) (Beck K, 1999) ενώ έχουν προταθεί και αρκετές άλλες όπως η Feature Driven Design (FDD) (Coad, P. Lefebvre, E. & De Luca, J., 1999), οι κρυσταλλικές μέθοδοι (Crystal Methods), Adaptive Software Development (Highsmith, 2000) κ.α. Οι πιο διαδεδομένες, ωστόσο, είναι ο ακραίος προγραμματισμός και η μεθοδολογία Scrum.

4.1.1. Αξίες – Αρχές – Ιδιότητες

Το μανιφέστο ευελιξίας (agile manifesto), στο οποίο αναφερθήκαμε παραπάνω, περιγράφει τέσσερις σημαντικές αξίες που σήμερα έχουν τόσο νόημα όσο είχαν τότε που περιγράφηκαν. Οι αξίες αυτές είναι οι ακόλουθες:

- I. Έμφαση στα άτομα και τις αλληλεπιδράσεις τους αντί στις διαδικασίες και τα εργαλεία ανάπτυξη συστημάτων.

- II. Έμφαση στην ανάπτυξη λογισμικού εργασίας αντί στην ανάπτυξη εμπειριστατωμένου τεκμηριωτικού υλικού.
- III. Έμφαση στην εγκαθίδρυση κλίματος συνεργασίας με τους πελάτες αντί στις στιβαρές διαπραγματεύσεις συμβολαίων με τους πελάτες.
- IV. Έμφαση στην αντιμετώπιση της αντίστασης στην αλλαγή (*resistance to change*) αντί στον αυστηρό προγραμματισμό (*planning*) των ενεργειών.

Έτσι, ενώ αναγνωρίζεται ότι ενυπάρχει αξία στα στοιχεία που βρίσκονται στο δεξιό μέρος των άνω προτάσεων (μετά τη λέξη “αντί”), δίδεται περισσότερη βαρύτητα, ή μεγαλύτερη έμφαση, στην αξία που ενυπάρχει στα στοιχεία που βρίσκονται στο αριστερό μέρος των προτάσεων (πριν τη λέξη “αντί”). (Βασιλακόπουλος Γ., 2018).

Το μανιφέστο, επίσης, περιλαμβάνει δώδεκα βασικά σημεία που αποτελούν τις αρχές του agile manifesto:

- I. Πρέπει να υπάρχει ικανοποίηση των πελατών μέσω της κατασκευής γρήγορων και συνεχών επαυξήσεων (*increments*).
- II. Πρέπει να εγκατασταθεί η πρώτη επαύξηση εντός μερικών εβδομάδων και ολόκληρο το σύστημα εντός μερικών μηνών.
- III. Πρέπει οι ομάδες πελατών και κατασκευαστών να εργάζονται από κοινού καθημερινά και καθ’ όλη τη διάρκεια του έργου.
- IV. Πρέπει οι ομάδες πελατών και κατασκευαστών να πραγματοποιούν συναντήσεις πρόσωπο-με-πρόσωπο.
- V. Πρέπει να είναι καλοδεχούμενες τυχόν απαιτήσεις που να προτείνονται από τους χρήστες ακόμη και στα τελευταία στάδια του έργου.
- VI. Πρέπει να υπάρχει εμπιστοσύνη και σεβασμός μεταξύ των μελών της ομάδας κατασκευαστών.
- VII. Πρέπει να μετράται η ταχύτητα εκτέλεσης του έργου μετά την παράδοση κάθε επαύξησης.
- VIII. Πρέπει να δίδεται έμφαση στον καλό σχεδιασμό για να αυξηθεί η ευελιξία.
- IX. Πρέπει να υπάρχει αυτό-οργάνωση για την καλύτερη αρχιτεκτονική και τον καλύτερο σχεδιασμό του συστήματος.

- X. Πρέπει να υπάρχει προσαρμογή και συντονισμός σύμφωνα με την κατάσταση και τις συνθήκες.
- XI. Πρέπει να ακολουθείται η αρχή της απλότητας (*keep it simple stupid - KISS*) καθ' όλη τη διεργασία ανάπτυξης του συστήματος.
- XII. Πρέπει να υπάρχει συνεπής εργασία από την αρχή μέχρι το τέλος των έργων ευέλικτης ανάπτυξης.

Είναι φανερό από τα παραπάνω ότι ουσιαστικά οι ευέλικτες μεθοδολογίες βασίζονται στις εξής ιδιότητες και είναι:

- Επαναληπτικές (*iterative*), αφού αρχικά παραδίδεται μια λειτουργική έκδοση του συστήματος και κατόπιν των υποδείξεων των χρηστών γίνονται αλλαγές στη λειτουργία κάθε υποσυστήματος σε κάθε νέα έκδοση.
- Αυξητικές (*incremental*), αφού το σύστημα, όπως ορίζεται στις απαιτήσεις, χωρίζεται σε υποσυστήματα με βάση τη λειτουργία τους και νέες λειτουργίες προστίθενται σε κάθε νέα έκδοση.
- Προκύπτουσες (*emergent*), αφού οι απαιτήσεις αλλά και η τεχνολογία που θα χρησιμοποιηθεί, προκύπτουν κατά τη διάρκεια του κύκλου ανάπτυξης.

Οι ευέλικτες μεθοδολογίες είναι προσαρμοσμένες στον άνθρωπο και λιγότερο στην τεκμηρίωση. Επίσης θεωρούνται προτιμότερες όταν οι ομάδες εργασίας είναι μικρές και τούτο διότι όσο μεγαλώνει ο αριθμός των ατόμων της ομάδας, δυσκολεύει ο συντονισμός και η επικοινωνία μεταξύ των μελών.

Η ίδια η ομάδα, ωστόσο, για την επιτυχία της μεθοδολογίας θα πρέπει να είναι αποτελεσματική ως προς την ποιότητα των ατόμων αλλά και τον τρόπο συνεργασίας και επικοινωνίας (Highsmith, J. and Cockburn, A., 2004).

Το μεγάλο μειονέκτημα που έχουν οι «παραδοσιακές» μεθοδολογίες είναι η αιτία της κυριαρχίας των ευέλικτων μεθοδολογιών, ιδιαίτερα στα έργα χαμηλού ρίσκου.

4.1.2. Διασφάλιση Ποιότητας

Οι ευέλικτες μεθοδολογίες για να υιοθετηθούν θα πρέπει σαφώς να απαντούν στα εξής ερωτήματα: Ποια είναι η ποιότητα του παραγόμενου λογισμικού ; Ποια είναι τα στοιχεία εκείνα που τεκμηριώνουν την ποιότητα ;

Όπως αναφέρθηκε και παραπάνω, η βασική ιδέα πίσω από τις ευέλικτες μεθοδολογίες είναι ότι το συνολικό έργο χωρίζεται σε μικρότερες χρονικές επαναληπτικές περιόδους, από μία έως τέσσερις εβδομάδες. Κάθε επανάληψη (κύκλος) προσομοιάζει με ένα μικρό έργο ανάπτυξης συστήματος και περιλαμβάνει όλες τις επιμέρους εργασίες που είναι αναγκαίες για την κατασκευή μιας ελάχιστης επαύξησης της λειτουργίας και λειτουργικότητας του συστήματος, δηλαδή τον προγραμματισμό (planning), την ανάλυση απαιτήσεων (requirements analysis), τον σχεδιασμό (design), την κωδικοποίηση (coding), τον έλεγχο ή τις δοκιμές (testing), και την τεκμηρίωση (documentation).

Το πιο σημαντικό χαρακτηριστικό των ευέλικτων μεθοδολογιών είναι ότι καθώς το έργο αναπτύσσεται, μεταβάλλονται κι επαναξιολογούνται οι αρχικές απαιτήσεις με βάση την ανάδραση από τον πελάτη (Alleman, 2002).

Η εμπλοκή του πελάτη παραμένει σταθερή σε όλη τη διάρκεια ζωής του έργου, σε αντίθεση με τις παραδοσιακές μεθοδολογίες, στις οποίες ο πελάτης εμπλέκεται κυρίως στην αρχή και στην ολοκλήρωση του έργου (Cohn, Agile Estimating and Planning, 2006).

Με βάση έρευνες που δημοσιεύονται, οι κυριότερες αιτίες για χρήση των ευέλικτων μεθοδολογιών είναι η ικανότητα της αφομοίωσης και προσαρμογής στις αλλαγές, η βελτίωση της ποιότητας, η ενίσχυση κινήτρων, η συχνότερη παράδοση λογισμικού, η μείωση ρίσκων και η αύξηση κόστους αποτελεσματικότητας.

Σήμερα, πολλοί από τους γίγαντες της βιομηχανίας λογισμικού χρησιμοποιούν ευέλικτες μεθόδους ανάπτυξης λογισμικού, συμπεριλαμβανομένων των Yahoo, Microsoft, Oracle, Sun Microsystems, HP, APL, IBM, Motorola, Xerox, Federal Reserve Bank, Capital One (Bhardwaj, 2011), αλλά και των Google, Apple και Philips (2017).

4.2. Μοντέλα Agile Μεθοδολογιών

4.2.1. Ακραίος Προγραμματισμός (Extreme Programming)

Ο ακραίος προγραμματισμός επινοήθηκε από τον Kent Beck στα τέλη της δεκαετίας του 1990 και αρχές της δεκαετίας του 2000. Ωστόσο, μετά από δοκιμές στην πράξη (Anderson, 1998), ο ακραίος προγραμματισμός καταγράφηκε για πρώτη φορά από τον Beck, περιγράφοντας τα βασικά χαρακτηριστικά και πρακτικές της μεθόδου, στο βιβλίο «Extreme Programming Explained» (Beck K. , 2004). Ο Jeffries (2001) ορίζει την XP ως την *«πειθαρχία της ανάπτυξης του λογισμικού στηριζόμενη στην απλότητα, την επικοινωνία, την ανάδραση και το κουράγιο»* .

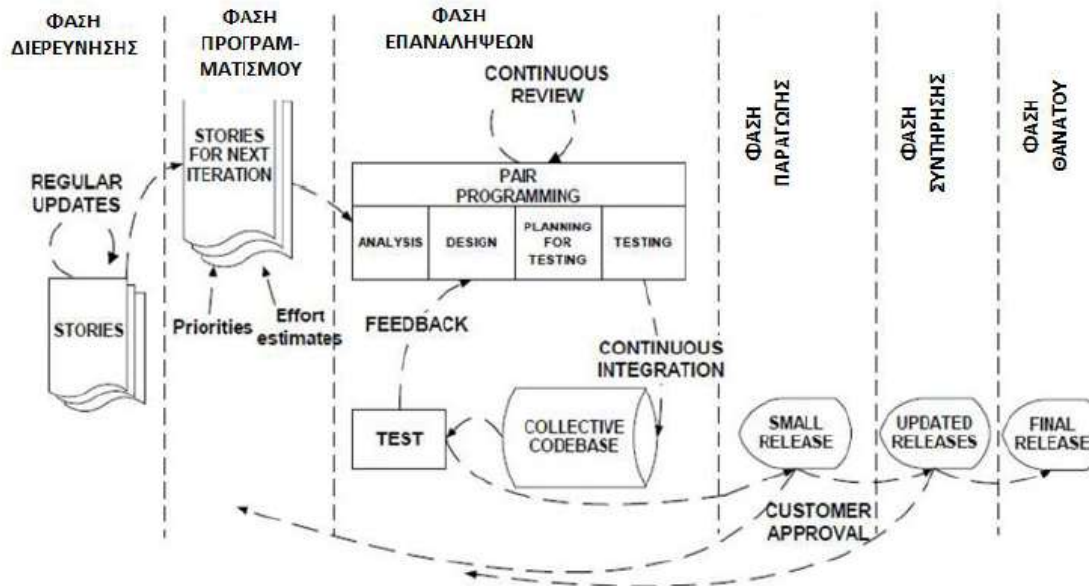
Ο ακραίος προγραμματισμός (Extreme Programming, XP), εντάσσεται στις ευέλικτες μεθοδολογίες, αφού προσπαθεί να βελτιώσει την ποιότητα του λογισμικού και επιτρέπει τις συχνές αλλαγές στις απαιτήσεις των χρηστών. Αποσκοπεί στην παροχή της δυνατότητας κατασκευής λογισμικού ακόμη και εντός ενός ασταθούς περιβάλλοντος επιτρέποντας σημαντική ευελιξία στη διεργασία μοντελοποίησης του συστήματος. Στον ακραίο προγραμματισμό η επικοινωνία μεταξύ κατασκευαστών και χρηστών είναι άτυπη και πραγματοποιείται σε καθημερινή βάση, «ανοικτή και τίμια επικοινωνία» τη χαρακτηρίζει ο Avison (David Avison & Guy Fitzgerald, 2006).

Η μεθοδολογία XP χαρακτηρίζεται από μικρούς κύκλους ανάπτυξης, επαναληπτικές εκδόσεις, τη συνεχή ανατροφοδότηση, τη συνεχή επικοινωνία και τον επαναστατικό σχεδιασμό (Beck K. , 2004). Κύριος στόχος της μεθοδολογίας είναι η μείωση του κόστους από αλλαγές των απαιτήσεων των χρηστών.

Ο κύκλος ζωής του XP περιλαμβάνει τις εξής φάσεις όπως αυτές παρουσιάζονται παρακάτω (Beck K. , 2004):

1. Διερεύνηση (Exploration)
2. Προγραμματισμός (Planning)
3. Επαναλήψεις (Iterations to release phase)
4. Παραγωγή (Productionizing)

5. Συντήρηση (Maintenance)
6. Θάνατο (Death)



Εικόνα 7. Κύκλος ζωής μεθοδολογίας Ακράιου Προγραμματισμού (Beck K. , 2004)

Συγκεκριμένα, η φάση της διερεύνησης περιλαμβάνει την καταγραφή των χαρακτηριστικών-λειτουργιών που πρόκειται να περιλαμβάνει το σύστημα. Οι πελάτες καταγράφουν σε «κάρτες ιστορίας» (story cards) τα χαρακτηριστικά γνωρίσματα που θα προστεθούν στο σύστημα, ενώ ταυτόχρονα η ομάδα ανάπτυξης εξοικειώνεται με τα εργαλεία, τις πρακτικές και την τεχνολογία που θα χρησιμοποιηθούν στο έργο και αναπτύσσεται τελικώς ένα πρωτότυπο του συστήματος.

Η φάση του προγραμματισμού καθορίζει εκείνες τις ιστορίες που θα τύχουν προτεραιότητας, καθώς και το τι θα περιέχει η πρώτη έκδοση του συστήματος. Γίνεται η αξιολόγηση των ιστοριών, ο υπολογισμός της προσπάθειας και του φόρτου εργασίας και κατασκευάζεται ένα χρονοδιάγραμμα εργασίας.

Η φάση των επαναλήψεων προβλέπει τον διαχωρισμό του προγράμματος που έχει καθοριστεί στην προηγούμενη φάση σε διάφορες επαναλήψεις, κάθε μια από τις οποίες διαρκεί από μια έως τέσσερις εβδομάδες. Η πρώτη επανάληψη περιλαμβάνει τον καθορισμό της αρχιτεκτονικής του συστήματος και στη συνέχεια ακολουθεί η αναβάθμιση του συστήματος με την ενσωμάτωση των

ιστοριών, ενώ δοκιμάζεται το προϊόν στο τέλος κάθε επανάληψης. Η ολοκλήρωση της επανάληψης λήγει με την παραγωγή της νέας έκδοσης.

Στη φάση της παραγωγής διεξάγονται περισσότεροι έλεγχοι και δοκιμές ώστε να εξασφαλιστεί η απόδοση του συστήματος πριν αυτό παραδοθεί στον πελάτη. Η φάση επιτρέπει εκ νέου εντοπισμό αλλαγών και αποφασίζεται αν αυτές θα γίνουν στην παρούσα επανάληψη. Επίσης, καταγράφονται νέες, ενδεχομένως, ιδέες και εισηγήσεις, ώστε σε επόμενες εκδόσεις να μπορούν να υλοποιηθούν στη φάση της συντήρησης.

Τέλος, στη φάση του θανάτου θεωρείται ότι ο πελάτης δεν έχει οποιαδήποτε απορία και το σύστημα ικανοποιεί όλες τις απαιτήσεις του. Στο στάδιο αυτό καταγράφεται η απαραίτητη τεκμηρίωση (documentation) του λογισμικού και δεν είναι αποδεκτές οποιεσδήποτε αλλαγές στην αρχιτεκτονική, στον σχεδιασμό ή στον κώδικα του συστήματος. Επίσης, ο θάνατος μπορεί να επέλθει πριν ακόμα το σύστημα προλάβει να ολοκληρωθεί, εάν παρατηρηθεί πως δεν μπορεί να επιφέρει τα επιθυμητά αποτελέσματα .

Για τη μεθοδολογία XP έχει εκδοθεί ερμηνευτικός οδηγός που στην πρώτη έκδοσή του περιλαμβάνονται δώδεκα βασικές πρακτικές, οι οποίες ομαδοποιούνται στις ακόλουθες τέσσερις περιοχές (Βασιλακόπουλος Γ., 2018):

- Ανατροφοδότηση μικρής κλίμακας (δοκιμαστική ανάπτυξη συστήματος, προγραμματιστικό παίγνιο, συμμετοχή ολόκληρης της ομάδας, προγραμματισμός κατά ζεύγη),
- Διαρκής διεργασία ανάπτυξης του συστήματος κατά ομάδες (διαρκής ολοκλήρωση ενοτήτων συστήματος, βελτίωση σχεδιασμού συστήματος, μικρές εκδόσεις συστήματος),
- Κοινή κατανόηση συστήματος (απλός σχεδιασμός συστήματος, προσομοίωση συστήματος, συλλογική ιδιοκτησία κώδικα συστήματος, προγραμματιστικά πρότυπα ή συμβατικότητες), και
- Ποιότητα ζωής προγραμματιστών (σταθερός και αποδεκτός χρόνος εργασίας, όπως η εβδομάδα 40 ωρών).

Στη δεύτερη έκδοση του ερμηνευτικού οδηγού της μεθοδολογίας XP καταγράφεται ένα σύνολο από πορισματικές πρακτικές, επιπλέον των

βασικών, που προέρχονται από γενικά αποδεκτές βέλτιστες πρακτικές και ασκούνται στην ολότητά τους. Αυτές είναι:

- Η αλληλεπίδραση μεταξύ των κατασκευαστών και των πελατών θεωρείται καλή πρακτική. Έτσι, μια ομάδα XP πρέπει να περιλαμβάνει έναν εκπρόσωπο του πελάτη, ο οποίος θα είναι επιφορτισμένος με τον προσδιορισμό και την ιεράρχηση των εργασιών της ομάδας και ο οποίος θα μπορεί να απαντά σε ερωτήσεις άμεσα.
- Η διαρκής επιδίωξη της μάθησης θεωρείται καλή πρακτική. Αυτό σημαίνει, εν προκειμένω, ότι πρέπει να μειώνεται ο χρόνος της ανάπτυξης και της ανατροφοδότησης (μάθησης) με συχνούς και σύντομους ελέγχους.
- Η κατασκευή μικρών προγραμμάτων (μικρής έκτασης κώδικας) θεωρείται καλή πρακτική για αύξηση της πιθανότητας απρόσκοπτης λειτουργίας τους. Έτσι, οι προγραμματιστές XP συγγράφουν προγράμματα μόνο για την ικανοποίηση πραγματικών τρεχόντων αναγκών του έργου, και προσπαθούν πραγματικά να μειώσουν την πολυπλοκότητα και την επανάληψη στα προγράμματά τους.
- Η κατασκευή απλών προγραμμάτων (απλός κώδικας) θεωρείται καλή πρακτική για αύξηση της πιθανότητας απρόσκοπτης λειτουργίας τους. Έτσι, αποφεύγεται η εξαρχής συγγραφή πολύπλοκων προγραμμάτων ή ξαναγράφονται σε απλούστερη μορφή τα προγράμματα που αρχικά γράφηκαν απλά αλλά εξελίχθηκαν σε πολύπλοκα.
- Η διαρκής ανασκόπηση των προγραμμάτων θεωρείται καλή πρακτική. Έτσι, οι προγραμματιστές XP εργάζονται κατά ζεύγη έτσι ώστε τα προγράμματα να ανασκοπούνται, και πιθανώς να διορθώνονται, καθώς κατασκευάζονται.
- Η διενέργεια συνεχών ελέγχων (δοκιμών) των προγραμμάτων θεωρείται καλή πρακτική. Έτσι, στη μεθοδολογία XP οι δοκιμές καταγράφονται (σχεδιάζονται) πριν συγγραφούν τα προγράμματα. Τα προγράμματα θεωρούνται πλήρη όταν υποστούν επιτυχώς τη

δοκιμασία των δοκιμών (αλλά τότε χρειάζεται ανακατασκευή τους προκειμένου να αφαιρεθεί η ενδεχόμενη πολυπλοκότητα). Το σύστημα ελέγχεται περιοδικά ή αμέσως μετά την κατασκευή του, χρησιμοποιώντας όλους τους προκαθορισμένους αυτοματοποιημένους ελέγχους, προκειμένου να διασφαλιστεί ότι λειτουργεί αποτελεσματικά και αποδοτικά κατά τα αναμενόμενα.

Στα πλεονεκτήματα της μεθοδολογίας XP θα πρέπει να αναφέρουμε, αρχικώς, τη σημαντική μείωση ελλιπούς κατανόησης των πραγματικών απαιτήσεων των χρηστών, αφού η μεθοδολογία επιβάλλει τη στενή συνεργασία των κατασκευαστών με τους χρήστες. Επίσης, σημαντική είναι και η μείωση του κόστους ανάπτυξης του συστήματος, δεδομένου ότι όλη η ανθρωποπροσπάθεια κατευθύνεται στην ανάπτυξη και στις δοκιμές και όχι σε παραγωγή τεκμηριωτικού υλικού. Τέλος, η μεθοδολογία XP είναι κατάλληλη για μικρά συστήματα που απαιτούν μικρές ομάδες προγραμματιστών, αλλά και για μεγαλύτερα, τα οποία διασπώνται σε μικρότερα.

Ως κύριο μειονέκτημα της μεθοδολογίας XP θα πρέπει να αναφέρουμε ότι έχει ως μοναδικό παραδοτέο το αναπτυχθέν σύστημα, δηλαδή δεν υπάρχει συνοδευτικό τεκμηριωτικό υλικό για τις φάσεις της ανάλυσης και του σχεδιασμού του συστήματος. Τούτο, έχει αρνητικές επιπτώσεις στη συντήρησή του. Αυτό το μειονέκτημα είναι πολύ σημαντικό σε μεγάλου εύρους και μεγάλης πολυπλοκότητας συστήματα καθιστώντας τη μεθοδολογία αυτή σχεδόν ανεφάρμοστη.

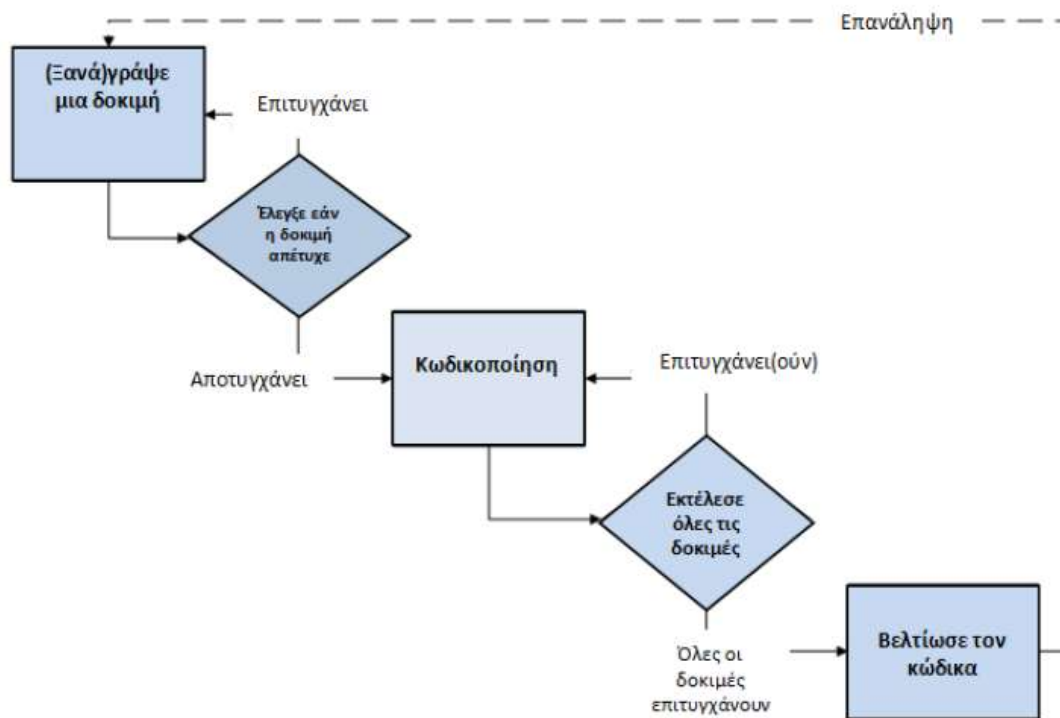
4.2.2. Ανάπτυξη με Βάση τις Δοκιμές (Test Driven Development)

Η ανάπτυξη με βάση τις δοκιμές είναι μια διαδικασία ανάπτυξης λογισμικού, που βασίζεται στην επανάληψη ενός πολύ σύντομου κύκλου ανάπτυξης. Κύριο χαρακτηριστικό είναι οι επαναλήψεις με βάση αυτοματοποιημένες δοκιμές (automated unit tests).

Η ανάπτυξη με βάση τις δοκιμές (Test Driven Development - TDD) προέρχεται από τις ευέλικτες μεθοδολογίες ανάπτυξης λογισμικού και η ανακάλυψή του οφείλεται στον Kent Beck, όταν το 2003 δημοσίευσε ένα βιβλίο με τίτλο «Test Driven Development: By Example» (Beck K. , 2003).

Ουσιαστικά, αντί της γραφής κώδικα πρώτα και τον έλεγχο εκ των υστέρων, η δοκιμή γίνεται πριν τη γραφή του λειτουργικού κώδικα (Beck K. , 2003). Τα χαρακτηριστικά του λογισμικού, κωδικοποιούνται ανεξάρτητα και ακολουθεί η δοκιμή προκειμένου να εξακριβωθεί εάν κάθε χαρακτηριστικό έχει πραγματοποιηθεί σωστά. Αν ο έλεγχος αποτύχει, ο κώδικας αλλάζει και προσαρμόζεται αναλόγως ώστε να δοκιμαστεί ξανά. Αν επιτύχει ο προγραμματιστής προχωράει στη φάση του “Refactoring”. Επιγραμματικά τα βήματα της διαδικασίας με βάση τον Beck είναι τα ακόλουθα:

- Πρόσθεσε μια καινούρια δοκιμή (Add a test)
- Εκτέλεσε όλες τις δοκιμές και έλεγξε αν κάποια αποτυγχάνει (Run all tests and see if the new one fails)
- Κωδικοποίηση (Write some code)
- Εκτέλεσε τις αυτοματοποιημένες δοκιμές και έλεγξε αν επιτυγχάνουν (Run the automated tests and see them succeed)
- Βελτίωση του κώδικα (Refactor code)
- Επανάληψη (Repeat)



Εικόνα 8. Κύκλος ζωής της Ανάπτυξης με βάση τις δοκιμές (Excirial, 2009)

4.2.3. Ανάπτυξη με Βάση τα Χαρακτηριστικά (Feature Driven Development)

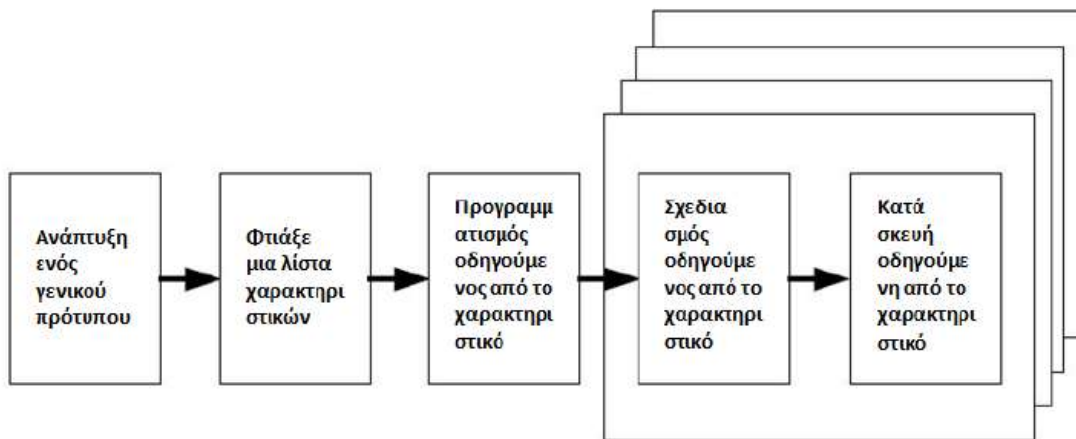
Η ανάπτυξη βάσει χαρακτηριστικών (Feature Driven Development, FDD), χρησιμοποιήθηκε για πρώτη φορά για την ανάπτυξη μιας πολύπλοκης τραπεζικής εφαρμογής στα τέλη της δεκαετίας του 1990 (Palmer , S.R. and Felsing J.M., 2002). Σε αντίθεση με τις υπόλοιπες μεθοδολογίες δεν ακολουθεί ολόκληρη τη διαδικασία ανάπτυξης λογισμικού, αλλά επικεντρώνεται στη φάση του σχεδιασμού και της υλοποίησης (Palmer , S.R. and Felsing J.M., 2002). Περιλαμβάνει πέντε (5) φάσεις με τις τρεις (3) πρώτες να εκτελούνται στην αρχή του έργου και τις δύο (2) τελευταίες να αποτελούν ουσιαστικά το επαναληπτικό μέρος της διαδικασίας. Χαρακτηριστικό της μεθοδολογίας είναι ότι εφαρμόζονται οι αρχές της ευέλικτης μεθοδολογίας ως προς τις γρήγορες προσαρμογές σε αλλαγές που παρουσιάζονται καθυστερημένα στις απαιτήσεις και τις επιχειρησιακές ανάγκες. Επίσης, περιλαμβάνει συχνές παραδόσεις και λεπτομερή παρακολούθηση της προόδου του έργου.

Τα βήματα της μεθοδολογίας είναι τα κάτωθι (Palmer , S.R. and Felsing J.M., 2002):

- Ανάπτυξη ενός γενικού προτύπου (Develop an Overall Model): Η φάση αυτή προβλέπει μια πρώτη γνωριμία με το πεδίο, το πλαίσιο και τις απαιτήσεις του συστήματος. Ενδέχεται να κατασκευαστούν περαιτέρω τεκμηριωμένες απαιτήσεις όπως οι περιπτώσεις χρήσης και οι λειτουργικές προδιαγραφές. Επίσης, πραγματοποιείται ένα λεπτομερές «πέραςμα» (walkthrough) του συστήματος και ο επικεφαλής αρχιτέκτονας (software architect) ενημερώνεται αναλυτικά σχετικά με το σύστημα. Το σύστημα διαιρείται σε περαιτέρω υποσυστήματα και διενεργούνται επιμέρους «περάσματα».
- Κατασκευή λίστας χαρακτηριστικών (Build a Features List): Παράγεται μια κατηγοριοποιημένη λίστα με τα χαρακτηριστικά για να υποστηρίξει τις απαιτήσεις.
- Προγραμματισμός οδηγούμενος από το χαρακτηριστικό (Plan by Feature): Η ομάδα ανάπτυξης ταξινομεί τα χαρακτηριστικά γνωρίσματα του συστήματος ανάλογα με τις προτεραιότητες και τις μεταξύ τους

εξαρτήσεις και αυτά ανατίθενται σε επικεφαλής προγραμματιστές. Επιπλέον, τα τμήματα που προσδιορίστηκαν στην πρώτη φάση ανατίθενται στους ιδιοκτήτες κατηγορίας (class owners) (μεμονωμένους προγραμματιστές). Επίσης, το χρονοδιάγραμμα και τα ορόσημα τίθενται για τα χαρακτηριστικά γνωρίσματα.

- Σχεδιασμός και Κατασκευή του συστήματος οδηγούμενα από το χαρακτηριστικό (Design & Build by Feature): Στο στάδιο αυτό επιλέγονται συγκεκριμένα χαρακτηριστικά γνωρίσματα για πραγμάτωση. Οι φάσεις αυτές είναι επαναληπτικές διαδικασίες αποτέλεσμα των οποίων είναι η πραγμάτωση των χαρακτηριστικών που επιλέγονται κάθε φορά. Μια ομάδα αναλαμβάνει να αναπτύξει μέσα από μια επανάληψη κάποιο χαρακτηριστικό γνώρισμα. Μπορεί να υπάρχουν πολλές ομάδες που δουλεύουν ταυτόχρονα στην πραγμάτωση διαφορετικών γνωρισμάτων. Μια επανάληψη διαρκεί από ορισμένες μέρες μέχρι το μέγιστο δύο βδομάδες. Όταν ολοκληρωθεί μια επανάληψη τότε το σύστημα αναβαθμίζεται με τις καινούριες λειτουργίες και τότε προκύπτει μια νέα έκδοση.



Εικόνα 9. Μεθοδολογία ανάπτυξης με βάση τα χαρακτηριστικά (Palmer , S.R. and Felsing J.M., 2002)

4.2.4. Μεθοδολογία Ανάπτυξη Δυναμικών Συστημάτων (Dynamic Systems Development Method)

Η μεθοδολογία ανάπτυξης δυναμικών συστημάτων (Dynamic Systems Development, DSD), πρωτοεμφανίστηκε στο Ηνωμένο Βασίλειο το 1994. Αποτελεί ένα μείγμα αλλά παράλληλα και επέκταση της γρήγορης προτυποποίησης και πρακτικών επαναληπτικής ανάπτυξης (Stapleton, 1997). Έχουν κυκλοφορήσει περισσότερες των τεσσάρων (4) εκδόσεων της συγκεκριμένης μεθοδολογίας μέχρι σήμερα.

Η DSDM, παρά το όνομα της, αποτελεί περισσότερο ένα πλαίσιο παρά μία μεθοδολογία και πολλή από την λεπτομέρεια του τρόπου που πραγματικά θα πρέπει να γίνουν τα πράγματα και τι θα περιλαμβάνουν τα ποικίλα προϊόντα εναπόκειται στην απόφαση του οργανισμού ή των ατόμων (Stapleton, 1997). Η μεθοδολογία είναι αξιοσημείωτη για το λόγο ότι ακολουθεί την υποδομή των παραδοσιακών μεθοδολογιών ανάπτυξης λογισμικού, ενώ παράλληλα ακολουθεί τις αρχές του ευέλικτου τρόπου ανάπτυξης (Fowler, 2005). Η βασική ιδέα πίσω από την DSD είναι να καθορίσει τον χρόνο και τους πόρους, και στη συνέχεια να προσαρμόσει το ποσό της λειτουργικότητας ανάλογα και όχι να καθορίσει το ποσό της λειτουργικότητας σε ένα προϊόν, και κατόπιν να προσαρμοστεί ανάλογα ο χρόνος και οι πόροι για την επίτευξη αυτής της λειτουργικότητας (DSMD Consortium, 2012). Η μεθοδολογία τονίζει τη σημασία του ανθρωπίνου παράγοντα στη διαδικασία και περιγράφεται ως προσέγγιση «επικεντρωμένη στο χρήστη» (David Avison & Guy Fitzgerald, 2006).

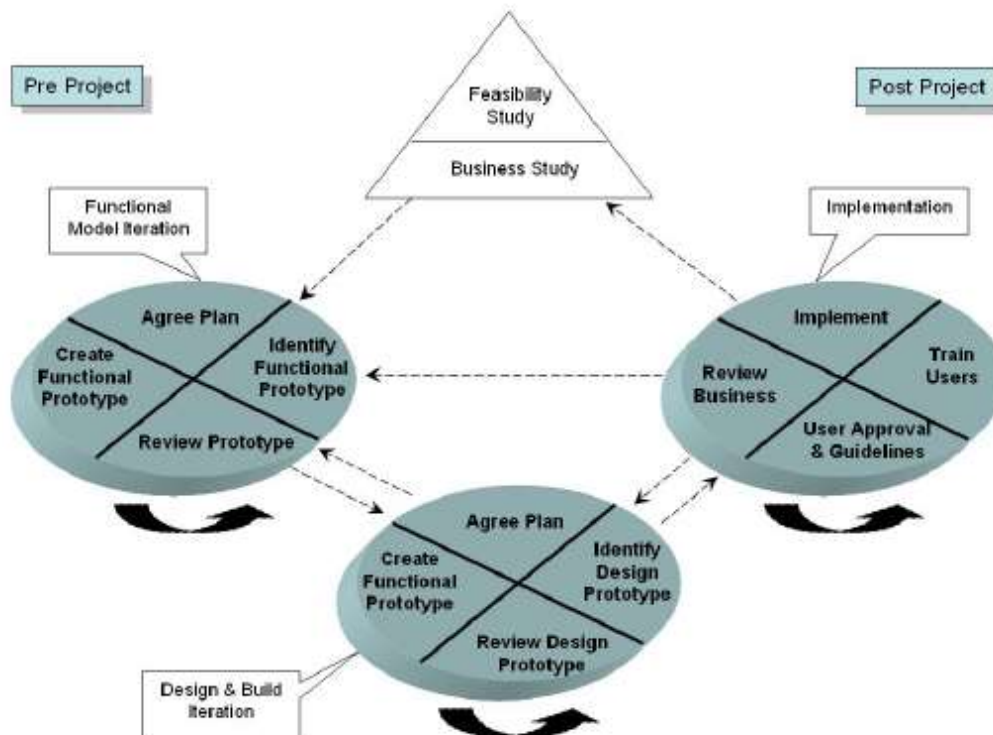
Οι σημαντικότερες εννέα αρχές που συμβάλουν στην επιτυχία της DSDM, είναι οι εξής (David Avison & Guy Fitzgerald, 2006):

1. Η ενεργή συμμετοχή του χρήστη είναι αναγκαία.
2. Οι ομάδες πρέπει να μπορούν να παίρνουν αποφάσεις. Οι τέσσερις προϋποθέσεις για απόφαση είναι: δικαίωμα απόφασης, πηγές, πληροφορίες και αξιοπιστία.
3. Η συχνή παράδοση προϊόντων είναι σημαντική.
4. Καλή φυσική κατάσταση για επιχειρηματικούς λόγους είναι σημαντικό κριτήριο για την αποδοχή των παραδοτέων.

5. Η επαναλαμβανόμενη και αυξανόμενη ανάπτυξη είναι απαραίτητη ώστε να συγκλίνουν σε μία έγκυρη επιχειρηματική λύση.
6. Όλες οι αλλαγές κατά τη διάρκεια ανάπτυξης είναι αναστρέψιμες.
7. Οι απαιτήσεις βασίζονται σε ένα υψηλό επίπεδο (π.χ. οι υψηλού επιπέδου επιχειρηματικές απαιτήσεις, εφόσον συμφωνηθούν δεν αλλάζουν). Αυτή είναι η βασική αρχή του έργου.
8. Οι έλεγχοι πρέπει να γίνονται σε όλο τον κύκλο ζωής.
9. Μία συνεργατική προσέγγιση μεταξύ των εμπλεκομένων είναι σημαντική.

Η DSDM αποτελείται από πέντε (5) φάσεις. Οι δύο (2) πρώτες φάσεις εκτελούνται σειριακά μόνο μια φορά. Οι υπόλοιπες τρεις (3) φάσεις, όπου και αναπτύσσεται το σύστημα, είναι, επαναληπτικές και αυξητικές. Ο χρόνος είναι το σημαντικότερο χαρακτηριστικό της μεθοδολογίας. Οι επαναλήψεις αντιμετωπίζονται σαν χρονικά παράθυρα. Ένα χρονικό παράθυρο έχει συγκεκριμένα όρια και η επανάληψη επιβάλλεται να ολοκληρωθεί μέσα στα χρονικά αυτά όρια.

Στην παρακάτω εικόνα απεικονίζεται το «*διάγραμμα των τριών πιτσών και του τυριού*» και στο οποίο φαίνονται οι φάσεις και τα κύρια παραδοτέα που πρέπει να παραχθούν σε κάθε φάση μαζί με τα διάφορα μονοπάτια κατά τη διαδικασία της εξέλιξης.



Εικόνα 10. Διαδικασία ανάπτυξης δυναμικών συστημάτων (Stapleton, 1997)

Οι πέντε κύριες φάσεις του κύκλου ζωής και σχεδίασης της DSDM είναι οι εξής:

1. Μελέτη εφικτότητας (Feasibility study): Στη φάση αυτή λαμβάνεται η απόφαση για το αν θα χρησιμοποιηθεί ή όχι η μεθοδολογία DSD. Τούτο, καθορίζεται από το είδος του έργου, οργανωτικά θέματα και κυρίως τον ανθρώπινο παράγοντα. Επιπλέον, παράγονται δύο προϊόντα εργασίας, μια έκθεση εφικτότητας και ένα γενικό σχέδιο για την ανάπτυξη.
2. Επιχειρηματική μελέτη (Business study): Η συνιστώμενη προσέγγιση σε αυτή τη φάση είναι να κατανοηθεί το επιχειρηματικό πεδίο του έργου. Τα βασικά αποτελέσματα αυτής της ενότητας είναι ο ορισμός της αρχιτεκτονικής του συστήματος και ένα πρωτότυπο σχέδιο αυτού.
3. Επανάληψη λειτουργικού μοντέλου (Functional model iteration): Είναι η πρώτη επαναληπτική φάση. Η φάση αυτή περιλαμβάνει την ανάλυση, την κωδικοποίηση και την πρωτοτυποποίηση. Τα αποτελέσματα που συλλέγονται από τη φάση αυτή χρησιμοποιούνται

ώστε να βελτιωθούν τα πρωτότυπα ανάλυσης. Το κύριο αποτέλεσμα της φάσης αυτής είναι ένα λειτουργικό πρωτότυπο.

4. Σχεδιασμός συστήματος και ανάπτυξη (Design and build iteration): Το σύστημα κτίζεται ουσιαστικά σε αυτή τη φάση. Ο σχεδιασμός και η λειτουργική αξιολόγηση των πρωτοτύπων γίνονται από τους χρήστες και η επιπλέον ανάπτυξη βασίζεται στα σχόλια των χρηστών.
5. Πραγμάτωση (Implementation): Σε αυτήν την τελική φάση, το σύστημα παραδίδεται στους χρήστες. Η κατάρτιση των χρηστών όσο αφορά τη χρήση του συστήματος, παρέχεται. Εγχειρίδια χρήστη και μια έκθεση ανασκόπησης του έργου είναι διαθέσιμα. Ωστόσο, η επαναληπτική και αυξητική φύση της DSD σημαίνει ότι η συντήρηση μπορεί να θεωρηθεί ως συνεχιζόμενη ανάπτυξη. Αντί να τελειώσει το έργο με ένα κύκλο, μπορεί να επιστρέψει σε κάποια από τις προηγούμενες φάσεις, έτσι ώστε το σύστημα να εμπλουτιστεί με τις απαιτήσεις που τυχόν υπολείπονται και να τελειοποιηθεί.

4.2.5. DevOps

Κατά τη διάσκεψη του οργανισμού των ευέλικτων μεθοδολογιών στο Τορόντο το 2008, ο Andrew Shafer και ο Patrick Debois εισήγαγαν για πρώτη φορά έναν νέο όρο στη συζήτησή τους για τις ευέλικτες μεθοδολογίες και πρακτικές. Ο όρος αυτός ήταν το “DevOps”. Από το 2009, ο όρος αυτός έχει προωθηθεί σταθερά και έχει τεθεί σε πιο ευρεία χρήση μέσω μιας σειράς «devopsdays» (συναντήσεις για το DevOps) που ξεκίνησε στο Βέλγιο και έχει πλέον εξαπλωθεί σε πολλές χώρες. Ο όρος DevOps έχει χρησιμοποιηθεί σε πολλά διαφορετικά περιβάλλοντα.

Ένας ορισμός που προτείνεται από τους Bass, Weber και Zhu είναι ο κάτωθι:

«Το DevOps είναι ένα σύνολο πρακτικών που αποσκοπούν στη μείωση του χρόνου μεταξύ της αλλαγής σε ένα σύστημα και της αλλαγής στην κανονική παραγωγή, εξασφαλίζοντας παράλληλα υψηλή ποιότητα».

Το DevOps είναι, ουσιαστικά, μια κουλτούρα και πρακτική εφαρμοσμένης μηχανικής λογισμικού που στοχεύει στην ενοποίηση της ανάπτυξης λογισμικού

(Dev) και της λειτουργίας του λογισμικού (Ops). Το κύριο χαρακτηριστικό του κινήματος DevOps είναι να υποστηρίξει την αυτοματοποίηση και την παρακολούθηση σε όλα τα στάδια της κατασκευής λογισμικού, από την ολοκλήρωση, τη δοκιμή, την έκδοση έως την ανάπτυξη και τη διαχείριση της υποδομής. Το DevOps στοχεύει σε συντομότερους κύκλους ανάπτυξης, στην αυξημένη συχνότητα ανάπτυξης, σε πιο αξιόπιστες εκδόσεις και σε πλήρη ευθυγράμμιση με τους επιχειρηματικούς στόχους.

Ορισμένα άρθρα της βιβλιογραφίας των DevOps υποθέτουν ή συνιστούν σημαντική συμμετοχή σε χρήση των DevOps εκτός του τμήματος πληροφορικής ενός οργανισμού και τονίζεται ότι «το DevOps είναι απλώς η ευέλικτη αρχή (agile principle) που ακολουθείται σε ολόκληρο τον οργανισμό».

Μια έρευνα που δημοσιεύθηκε τον Ιανουάριο του 2016 από την εταιρία SaSS Cloud Computing RightScale, σημειώνει ότι η υιοθέτηση DevOps αυξήθηκε από 66% το 2015 σε 74% το 2016. Μεταξύ των μεγαλύτερων οργανισμών, η υιοθέτηση DevOps είναι ακόμη υψηλότερη - 81%. Η υιοθέτηση των DevOps οφείλεται σε πολλούς παράγοντες - μεταξύ των οποίων:

- Χρήση ευέλικτων και άλλων διαδικασιών και μεθόδων ανάπτυξης.
- Η ζήτηση για αυξημένες εκδόσεις παραγωγής - από τους ενδιαφερόμενους φορείς εφαρμογής και επιχειρηματικών μονάδων.
- Μεγάλη διαθεσιμότητα virtual και cloud υποδομής - από εσωτερικούς και εξωτερικούς παρόχους.
- Αυξημένη χρήση της αυτοματοποίησης των κέντρων δεδομένων (data centers) και των εργαλείων παραμετροποίησης.
- Αυξημένη εστίαση στην αυτοματοποίηση δοκιμών και σε μεθόδους συνεχούς ενσωμάτωσης.
- Μεγάλος αριθμός διαθέσιμων στο κοινό βέλτιστων πρακτικών.

4.2.6. Μεθοδολογία Scrum

Η μεθοδολογία Scrum είναι μία μεθοδολογία ευέλικτης ανάπτυξης συστημάτων η οποία βασίζεται στη φιλοσοφική θεώρηση ότι «τα προβλήματα ανάπτυξης συστημάτων είναι εν πολλοίς αδόμητα και είναι δύσκολο να κατανοηθούν η να οριστούν στην απαιτούμενη έκταση και στο απαιτούμενο βάθος»

(Βασιλακόπουλος Γ., 2018). Είναι μία επαναληπτική και αυξητική διαδικασία ανάπτυξης λογισμικού. Η προσέγγιση αυτή αρχικά κατασκευάστηκε με στόχο την ανάπτυξη έργων λογισμικού. Ωστόσο, αργότερα αποδείχθηκε ότι μπορεί να χρησιμοποιηθεί για τη γενικότερη διαχείριση έργων.

Το όνομα της οφείλεται στη λειτουργική «ομάδα» που συγκρίνεται με αυτήν του ράγκμπι (εξ ου και η ονομασία), όπως, δηλαδή στο ράγκμπι, όλη η ομάδα σαν μια οντότητα προσπαθεί να διανύσει την απόσταση (δημιουργία του προϊόντος), περνώντας την «μπάλα» μπρος-πίσω. Η μεθοδολογία βελτιώνει αισθητά την ταχύτητα ανάπτυξης και τον βαθμό ευελιξίας στη διαδικασία οικοδόμησης ενός εμπορικού προϊόντος (Takeuchi, H. and Nonaka, I., 1986). Ονομάστηκε ολιστική προσέγγιση (holistic approach). Αργότερα, το 1990 οι DeGrace και Stahl έκαναν αναφορά σε αυτήν την προσέγγιση ως Scrum. Ένας όρος που επίσης αναφέρουν οι Takeuchi και Nonaka στο άρθρο τους. Στις αρχές της δεκαετίας του 1990, ο Ken Schwaber χρησιμοποίησε στην εταιρία του τη διαδικασία αυτή που ονομαζόταν Scrum. Ο Jeff Sutherland, με τον John Jeff Scumniotales και τον McKenna ανέπτυξαν μια παρόμοια προσέγγιση στο Easel Corporation, και ήταν οι πρώτοι που αναφέρθηκαν επίσημα σε αυτήν την μεθοδολογία χρησιμοποιώντας τον όρο Scrum (Sutherland, 2004). Το 2001 ο Sutherland εργάστηκε μαζί με τον Mike Beedle για να καταγράψουν τη μεθοδολογία αυτή στο βιβλίο «Agile Software Development with Scrum» (Schwaber, K. and Beedle, M., 2001).

Η μεθοδολογία Scrum είναι η πλέον δημοφιλής, ίσως, και ευρέως χρησιμοποιούμενη ευέλικτη μεθοδολογία, πιθανώς λόγω του γεγονότος ότι εφαρμόζεται σχετικά εύκολα και αντιμετωπίζει πολλά από τα διοικητικά θέματα που μάστιζαν την ανάπτυξη συστημάτων για πολλές δεκαετίες. Ουσιαστικά, η μεθοδολογία υιοθετεί μια εμπειρική προσέγγιση εστιάζοντας στην ανάπτυξη συστημάτων από ομάδες και στη μεγιστοποίηση των ικανοτήτων των ομάδων να ανταποκριθούν κατά ευέλικτο τρόπο στις όποιες προκλήσεις παρουσιαστούν. Η επιτυχία της μεθόδου οφείλεται στον τρόπο με τον οποίο προσεγγίζει και διαχειρίζεται την έννοια της διαδικασίας ανάπτυξης του λογισμικού.

Χρησιμοποιείται ευρέως από μεγάλες και μικρές εταιρίες, μεταξύ των οποίων οι Yahoo, Microsoft, Google, Lockheed Martin, Motorola, SAP, Cisco, GE, CapitalOne, US Federal Reserve και Apple. Πολλές ομάδες που χρησιμοποιούν τη μέθοδο αυτή, αναφέρουν σημαντικά οφέλη και σε ορισμένες περιπτώσεις ολοκληρωτική βελτίωση της παραγωγικότητας.

Στα επόμενα κεφάλαια περιγράφεται αναλυτικά η μεθοδολογία Scrum τόσο σε θεωρητικό επίπεδο όσο και σε εφαρμογή της στη πράξη με μελέτη περίπτωσης.

4.3. Επιλογή ή Μη των Ευέλικτων Μεθοδολογιών

Είναι αλήθεια, ότι οι ευέλικτες μεθοδολογίες δεν αποτελούν πανάκεια για την ανάπτυξη όλων των ειδών συστημάτων. Έχουν και αυτές κάποια ασθενή σημεία τα οποία επιγραμματικά είναι:

- Έλλειψη εφαρμογής της πραγματικής ευέλικτης προσέγγισης. Ένα από τα μεγαλύτερα προβλήματα είναι ότι η πραγματική ευέλικτη ανάπτυξη εφαρμόζεται σπανίως.
- Ανάγκη πρόσθετης εργασίας. Λόγω της έλλειψης μακροπρόθεσμου προγραμματισμού και της μικρής έμφασης στην αρχιτεκτονική του συστήματος που επιτάσσει η ευέλικτη προσέγγιση, συχνά απαιτείται σημαντική πρόσθετη εργασία όταν συνδυάζονται οι διάφορες συνιστώσες του λογισμικού και καθίστανται αλληλεπιδραστικές.
- Ισχυρή προσκόλληση στις αρχές του μανιφέστου ευελιξίας. Αυτό αφορά κυρίως την ισχυρή αλληλεπίδραση με τους πελάτες που είναι και ένα από τα σημαντικότερα οφέλη της ευέλικτης προσέγγισης που, όμως, μπορεί να μετατραπεί σε ένα μειονέκτημα σε κάποια περιβάλλοντα. Για παράδειγμα αν ο πελάτης δεν μπορεί να αφιερώσει τον χρόνο που απαιτείται για τη διαδικασία ανάπτυξης, η χρήση της ευέλικτης προσέγγισης θα αποφέρει ελάχιστο όφελος σε σχέση με την καταρακτοειδή προσέγγιση.
- Η ευέλικτη ανάπτυξη δίδει μικρή έμφαση στον αρχιτεκτονικό προγραμματισμό.
- Δυσκολία συντονισμού για μεγάλα έργα και μεγάλους οργανισμούς.

- Η ευέλικτη μεθοδολογία εφαρμόζεται καλύτερα με ομάδες τα μέλη των οποίων βρίσκονται στον ίδιο τόπο.
- Μικρή έμφαση στον πυρήνα της διοίκησης έργων.

Ωστόσο, όπως είναι φυσικό οι ευέλικτες μεθοδολογίες έχουν και πολύ ισχυρά σημεία, τα οποία είναι επιγραμματικά:

- Αντιμετώπιση του πληθωρισμού των απαιτήσεων.
- Απόκτηση γρήγορων οφελών από τους χρήστες και τον οργανισμό.
- Βελτίωση της παραγωγικότητας της ανάπτυξης του συστήματος.
- Μείωση της ισχύος του παράγοντα “έλλειψη εργαζομένων”.
- Εξαφάνιση σφαλμάτων που ενυπάρχουν στο χρονοδιάγραμμα του έργου.
- Μη εμφάνιση αντικρουόμενων απαιτήσεων.

Για όλους τους παραπάνω λόγους, οι ευέλικτες μεθοδολογίες είναι κατάλληλες για τις εξής καταστάσεις (Βασιλακόπουλος Γ., 2018):

- Όταν το έργο είναι μικρομεσαίας κλίμακας και μικρής διάρκειας και άνθρωπο-προσπάθειας (π.χ. λιγότερο από έξι άνθρωπο-έτη).
- Όταν υπάρχει δέσμευση της ανώτερης διοίκησης του οργανισμού προκειμένου να διασφαλιστεί η ενεργός συμμετοχή των χρηστών.
- Όταν οι απαιτήσεις του συστήματος είναι άγνωστες ή ασαφείς κατά την έναρξη του έργου.
- Όταν δεν είναι δυνατή η χρήση απαιτήσεων από προγενέστερα αναπτυχθέντα συστήματα είτε διότι η προβληματική κατάσταση είναι νέα είτε διότι το σύστημα είναι πολύ καινοτομικό.
- Όταν τα μέλη της ομάδας ανάπτυξης του συστήματος διαθέτουν δεξιότητες και σε κοινωνικό και σε επιχειρησιακό επίπεδο.
- Όταν η σύνθεση της ομάδας ανάπτυξης είναι σταθερή και μπορεί να διατηρηθεί η συνέχεια του κορμού της ομάδας.
- Όταν οι κατασκευαστές διαθέτουν την ικανότητα χρήσης προηγμένων αυτοματοποιημένων εργαλείων.
- Όταν η τεχνική αρχιτεκτονική του συστήματος είναι προκαθορισμένη.

- Όταν οι βασικές τεχνικές συνιστώσες έχουν κατασκευαστεί και δοκιμαστεί.
- Όταν η ομάδα ανάπτυξης του συστήματος διαθέτει τη δυνατότητα να λαμβάνει αποφάσεις επί του σχεδιασμού του συστήματος σε καθημερινή βάση χωρίς την ανάγκη διαβούλευσης με τους ανωτέρους τους και οι αποφάσεις μπορεί να λαμβάνονται συλλογικά από μικρό αριθμό ατόμων.

Για τους ίδιους λόγους είναι ελάχιστα κατάλληλες στις εξής καταστάσεις:

- Όταν οι χρήστες που είναι μέλη της ομάδας ανάπτυξης δεν είναι εντελώς αφιερωμένοι στο έργο για τον βαθμό και το είδος της συμμετοχής που απαιτείται.
- Όταν πρόκειται για πολύ μεγάλα έργα υποδομής, ιδιαίτερα μεγάλα και κατανεμημένα ΠΣ και μεγάλες και κατανεμημένες βάσεις δεδομένων.
- Όταν πρόκειται για συστήματα πραγματικού χρόνου και για συστήματα κρίσιμης ασφάλειας δεδομένων.
- Όταν ο χρόνος και το κόστος του έργου είναι συγκεκριμένα και δεν είναι δυνατή οποιαδήποτε υπέρβαση αυτών, παρά το γεγονός ότι απαιτούνται περισσότερες επαναλήψεις όπως αποδεικνύεται κατά τη διάρκεια εκτέλεσης του έργου.
- Όταν δεν είναι δυνατή η αναθεώρηση του εύρους του συστήματος παρά το γεγονός ότι είναι εμφανής η αναγκαιότητα αναθεώρησης σε μεταγενέστερο χρόνο, ιδιαίτερα σε μεγάλα έργα ή σε συστήματα υψηλού βαθμού ολοκλήρωσης.
- Όταν πρόκειται για περίπλοκα υπολογιστικά συστήματα, όπου πολύπλοκα και ογκώδη δεδομένα πρέπει να αναλυθούν, να σχεδιασθούν και να αναδομηθούν εντός του εύρους του έργου.
- Όταν το έργο καλύπτει ένα ευρύ φάσμα δραστηριοτήτων και επιχειρησιακών διεργασιών του οργανισμού ενώ, ταυτόχρονα, οι επιχειρησιακοί στόχοι του έργου είναι ατελώς ορισμένοι (ασαφείς).
- Όταν πρόκειται για συστήματα στα οποία πρέπει να προσδιορισθούν πλήρως οι λειτουργικές απαιτήσεις πριν την ανάπτυξη των αντιστοίχων προγραμμάτων εφαρμογών.

- Όταν εμπλέκονται πολλά άτομα στη λήψη αποφάσεων για το έργο και οι αποφασίζοντες δεν είναι διαθέσιμοι όταν απαιτείται ή είναι γεωγραφικά διάσπαρτοι.
- Όταν η ομάδα ανάπτυξης είναι μεγάλη ή υπάρχουν πολλαπλές ομάδες έργου οι οποίες πρέπει να συγχρονιστούν και συντονιστούν.
- Όταν υπάρχει έλλειψη πόρων από τον οργανισμό ή δέσμευσης από τους χρήστες.
- Όταν δεν υπάρχει κάποιο άτομο στο ανώτατο δυνατό επίπεδο το οποίο να μπορεί να δώσει τις αναγκαίες εντολές για την πραγμάτωση του συστήματος και οι εντολές να εισακουστούν.
- Όταν πρόκειται να εισαχθούν πολλές επιμέρους τεχνολογίες στο πλαίσιο του έργου ή όταν η τεχνική αρχιτεκτονική είναι ασαφής και οι περισσότερες τεχνολογίες θα χρησιμοποιηθούν για πρώτη φορά σ' αυτό το έργο.
- Όταν είναι αναγκαία η ύπαρξη τεκμηρίωσης.

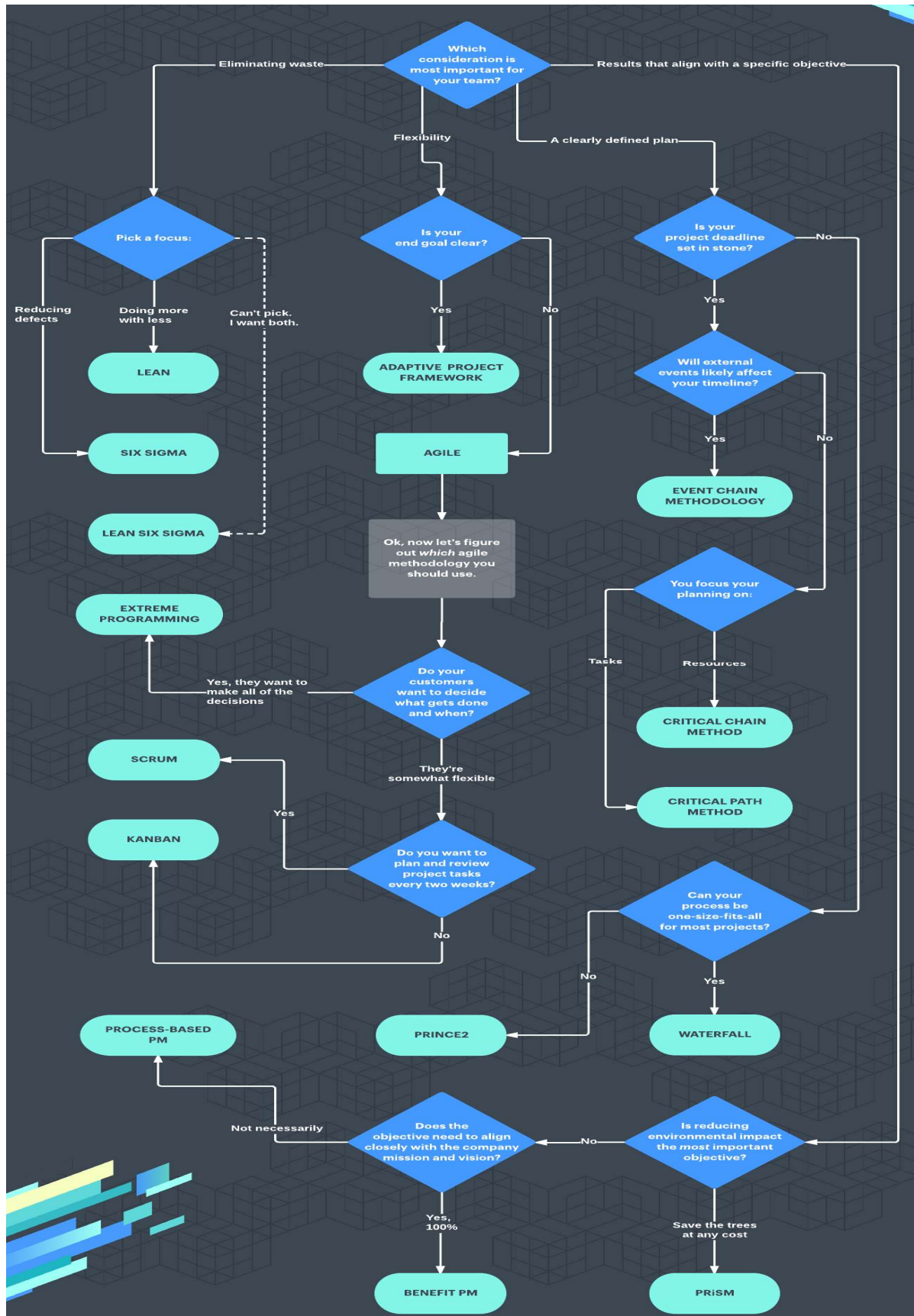
5. Επιλογή Μεθοδολογίας Ανάπτυξης Συστημάτων

Όπως γίνεται εύκολα αντιληπτό, από όσα αναφέρθηκαν διεξοδικώς στις προηγούμενες παραγράφους, υπάρχει μεγάλη ποικιλία μεθοδολογιών ανάπτυξης συστημάτων. Εκτός των μεθοδολογιών που αναφέρθηκαν παραπάνω, οι οποίες ουσιαστικά αποτελούν και τις πιο διαδεδομένες, υπάρχουν και πολλά άλλα μοντέλα, τα οποία είναι υβριδικά και σχεδιασμένα, έτσι ώστε να αντιμετωπίζουν συγκεκριμένες ανάγκες και να μειώνουν τις αδυναμίες των ατομικών μοντέλων. Τούτο, έχει ως αποτέλεσμα το ότι η διαδικασία επιλογής κατάλληλης μεθοδολογίας ανάπτυξης συστήματος μπορεί να αποδειχθεί αρκετά πολύπλοκη. Υπάρχουν, ωστόσο, κάποιοι βασικοί παράγοντες κάθε έργου που μπορούν να διευκολύνουν τη λήψη της σχετικής απόφασης. Οι πιο βασικοί εξ αυτών είναι:

- Οι απαιτήσεις του συστήματος: Αν οι απαιτήσεις είναι αυστηρά ορισμένες εξ αρχής και αμετάβλητες, η διοίκηση του έργου μπορεί να επιλέξει μια καταρακτοειδή προσέγγιση. Αν δεν ισχύει κάτι τέτοιο και αναμένεται να αλλάζουν διαρκώς ενδεχομένως να επιλέξει μια πιο ευέλικτη, επαναληπτική και επαυξητική προσέγγιση.
- Το χρονοδιάγραμμα της ανάπτυξης του συστήματος: Είναι προφανές ότι αν το χρονοδιάγραμμα είναι σφικτό τότε μια πιο αυστηρή προσέγγιση βήμα-προς-βήμα που βασίζεται σε εκτεταμένη τεκμηρίωση και δοκιμές σε μεταγενέστερα στάδια του έργου θα αποδεικνύονταν υπερβολικά αργή. Συνεπώς, θα αποκλείονταν τα καταρακτοειδή μοντέλα.
- Η διάσταση του έργου: Όσο μεγαλύτερο είναι το έργο, τόσο αυστηρότερο τείνει να είναι το μοντέλο ανάπτυξης που θα ακολουθηθεί.
- Ο τόπος εγκατάστασης της ομάδας έργου: Αν τα μέλη της ομάδας ανάπτυξης ή έργου είναι γεωγραφικά διάσπαρτα, η υιοθέτηση ενός καταρακτοειδούς μοντέλου είναι πιθανώς η βέλτιστη επιλογή λόγω της καθαρότητας των φάσεων και των εργασιών που περιλαμβάνονται σ' αυτές. Είναι, ωστόσο, γεγονός ότι σήμερα διατίθενται αυτοματοποιημένα εργαλεία συνεργασίας που καθιστούν την προφορική επικοινωνία μεταξύ των μελών των ομάδων πολύ εύκολη και συνεπώς την υιοθέτηση μιας ευέλικτης μεθοδολογίας πιο συχνή.

- Οι διαθέσιμοι πόροι: Έργα στα οποία εμπλέκονται συμφέροντα, χρηματοοικονομικές δυνάμεις, συνδικαλιστικές δυνάμεις, πολιτικοί οργανισμοί κτλ. και τα οποία απαιτούν τη χρήση ιδιαίτερων δεξιοτήτων/εμπειριών και τεχνολογιών είναι πιο εύκολο να εκτελεστούν χρησιμοποιώντας μοντέλα αυστηρού προγραμματισμού, όπως το καταρρακτοειδές. Είναι, επίσης, πιθανόν το θεσμικό πλαίσιο να θέτει περιορισμούς που να επιβάλλουν την υιοθέτηση του καταρρακτοειδούς μοντέλου.

Η επιλογή του καταλληλότερου μοντέλου για ένα έργο αποτελεί ένα κρίσιμο βήμα στη διεργασία ανάπτυξης του αντιστοίχου συστήματος. Μέχρι και σήμερα, και σίγουρα και στο μέλλον, οι μεθοδολογίες ανάπτυξης θα συνεχίζουν να ερευνώνται, να αναπτύσσονται και να αξιοποιούνται τόσο σε θεωρητικό υπόβαθρο, αλλά κυρίως στην πράξη.



Εικόνα 11. Οδηγός επιλογής Μεθοδολογίας (<https://www.lucidchart.com>)

6. Μεθοδολογία Scrum

Όπως αναφέρθηκε, προηγουμένως, η Scrum είναι μια επαναληπτική και αυξητική προσέγγιση για την ανάπτυξη έργων. Σκοπός της μεθοδολογίας είναι ο έλεγχος του κινδύνου, καθώς επιτρέπει την προσαρμογή στις αλλαγές που μπορεί να προκύψουν από τις μεταβολές των απαιτήσεων των πελατών στη διάρκεια του έργου. Μπορεί να χρησιμοποιηθεί για τη διαχείριση, βελτίωση και συντήρηση ενός υπάρχοντος ή ενός νέου συστήματος, ενώ παρέχει ένα πλαίσιο διαχείρισης γενικού σκοπού, με τα κατάλληλα βήματα και διαδικασίες, που μπορεί να εφαρμοστεί σε έργα με πειστικές ημερομηνίες παράδοσης, με σύνθετες απαιτήσεις και συγκεκριμένες ιδιαιτερότητες. Ο ακρογωνιαίος λίθος της μεθοδολογίας είναι η διαδικασία του εμπειρικού ελέγχου. Επίσης, προβλέπει διαφορετική δομή οργάνωσης και διοίκησης του έργου σε σχέση με τις υπόλοιπες μεθοδολογίες. Επιπροσθέτως, υποστηρίζει την παράδοση τμημάτων κώδικα του συστήματος που συμφωνείται, σε όσο το δυνατό μικρότερο χρονικό διάστημα κύκλου ανάπτυξης ανά κύκλο.

6.1. Φάσεις της Μεθοδολογίας Scrum

Σύμφωνα με τους Schwaber και Beedle (Schwaber, K. and Beedle, M., 2001), η μεθοδολογία Scrum απαρτίζεται από τρεις (3) σημαντικές φάσεις:

1. Αρχική διερεύνηση

Η φάση αυτή περιλαμβάνει δύο επιμέρους φάσεις, της ανάλυσης και του υψηλού επιπέδου σχεδιασμού.

- Φάση Ανάλυσης

Εδώ γίνεται ο καθορισμός των απαιτήσεων και η πιστοποίηση του συστήματος που πρόκειται να αναπτυχθεί. Συγκεκριμένα καθορίζονται οι προδιαγραφές του συστήματος και οι απαιτήσεις του χρήστη. Σημαντικό ρόλο σε αυτή τη φάση διαδραματίζει ο πελάτης, ο οποίος καλείται να ορίσει με λεπτομέρεια τι θέλει να κάνει το σύστημα. Σε συνεργασία με τον πελάτη δημιουργείται ένας κατάλογος ανεκτέλεστων προϊόντων (Product Backlog list), ο οποίος περιέχει όλες τις γνωστές,

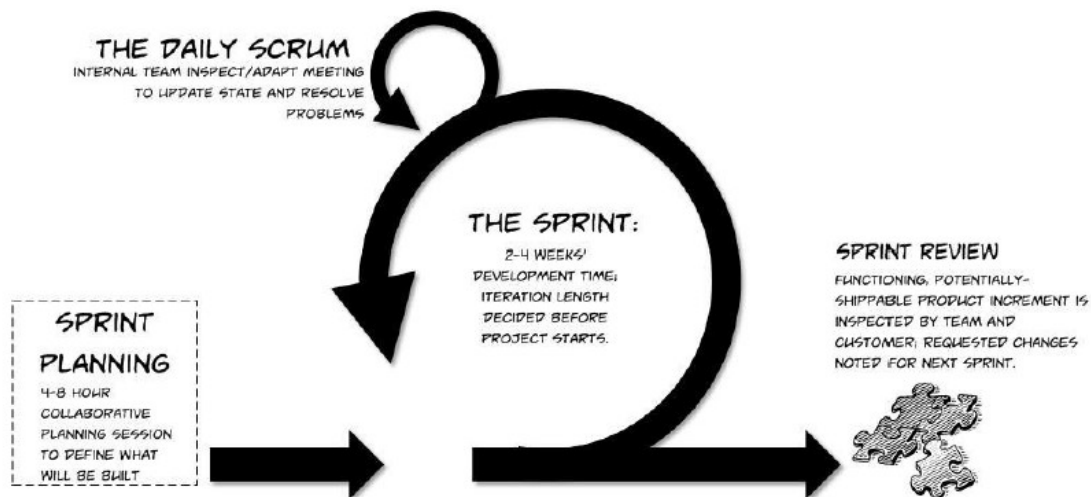
μέχρι την παρούσα περίοδο, προδιαγραφές που πρέπει να πραγματοποιούν. Αυτές κατατάσσονται ανάλογα με την προτεραιότητά τους και υπολογίζεται η προσπάθεια που απαιτείται για την πραγμάτωσή τους. Σε ολόκληρη τη διάρκεια του κύκλου ζωής, ο κατάλογος αυτός ενημερώνεται με νέα πιο λεπτομερή στοιχεία, καθώς επίσης και με πιο ακριβείς εκτιμήσεις για την αναγκαία προσπάθεια, ενώ επιτρέπεται ακόμα και ο ορισμός νέων προτεραιοτήτων. Αναλύονται, τέλος οι λεπτομέρειες μία προς μία, ώστε να διαπιστωθεί ο βαθμός δυσκολίας πραγμάτωσής τους, όπως και η τυχόν ενσωμάτωση επιπλέον τεχνογνωσίας στη διαδικασία πραγμάτωσης.

- Φάση υψηλού επιπέδου σχεδιασμού
Στη φάση αυτή πραγματοποιείται η αρχιτεκτονική και ο σχεδιασμός του συστήματος με βάση τις αρχικές προδιαγραφές, οι οποίες καθορίστηκαν στην προηγούμενη φάση.

2. Ανάπτυξη

Η φάση της ανάπτυξης χαρακτηρίζει την ευελιξία της μεθοδολογίας. Εδώ λαμβάνονται υπόψιν πολλές παράμετροι, όπως ο χρόνος παράδοσης, η ποιότητα, οι απαιτήσεις κ.α., οι οποίες μπορεί να μεταβληθούν κατά τη διάρκεια ανάπτυξης, αλλά και να προβλεφθούν και να αντιμετωπιστούν αστάθμητοι παράγοντες και κίνδυνοι. Σε αυτή τη φάση το σύστημα αναπτύσσεται σε επαναληπτικούς κύκλους εργασίας, οι οποίοι ονομάζονται Sprints. Κάθε κύκλος συμπεριλαμβάνει όλες τις φάσεις ανάπτυξης. Οι επαναλήψεις δε διαρκούν περισσότερο από έναν μήνα και εκτελούνται σειριακά χωρίς οποιεσδήποτε χρονικές καθυστερήσεις μεταξύ τους. Τα Sprints έχουν συγκεκριμένο χρονοδιάγραμμα το οποίο ποτέ δεν παρακάμπτεται, έστω και αν η καθορισμένη εργασία δεν έχει ολοκληρωθεί. Στην αρχή κάθε Sprint η ομάδα επιλέγει τα στοιχεία, τα οποία αποτελούν τις απαιτήσεις των πελατών, από μια λίστα στην οποία οι απαιτήσεις είναι ταξινομημένες βάσει προτεραιότητας, όπως αναφέρθηκε προηγουμένως. Η ομάδα δεσμεύεται να υλοποιήσει τα στοιχεία που επιλέχθηκαν μέχρι το τέλος του Sprint. Κατά τη διάρκεια του Sprint, τα

επιλεγόμενα στοιχεία δεν αλλάζουν. Κάθε μέρα η ομάδα οργανώνει σύντομες συγκεντρώσεις για να επιθεωρήσει την πρόοδο της, και να προσδιοριστούν τα επόμενα βήματα που χρειάζονται για να ολοκληρώσουν την υπόλοιπη εργασία. Στο τέλος κάθε επανάληψης πρέπει να υπάρχει ένα «λειτουργικό παραδοτέο προϊόν». Στην περίπτωση του λογισμικού, αυτό σημαίνει κώδικας που ενσωματώνεται πλήρως και έχει δοκιμαστεί. Για τον λόγο αυτό τα μέλη της ομάδας θα πρέπει να έχουν μια κοινή κατανόηση για το πότε μια εργασία θεωρείται ολοκληρωμένη (Done). Για την ομάδα αυτό περιγράφεται με τον όρο «Definition of Done» και χρησιμοποιείται για να αξιολογείται πότε μια εργασία έχει ολοκληρωθεί και μπορεί να ενσωματωθεί στο προσαυξημένο προϊόν. Με τον τρόπο αυτόν εξασφαλίζεται η ποιότητα για το τελικό προϊόν (Schwaber, K. and Sutherland J., 2011). Αναλυτικότερα, ο όρος θα αναλυθεί σε επόμενη παράγραφο.



Εικόνα 12. Φάσεις του Sprint της Scrum

3. Φάση ολοκλήρωσης

Στη φάση αυτή εκτελούνται όλες εκείνες οι διαδικασίες που απαιτούνται για την ολοκλήρωση και την τελική παράδοση του έργου στον πελάτη.

6.2. Ρόλοι και Αρμοδιότητες στη Scrum

Στη μεθοδολογία υπάρχουν τρεις (3) βασικοί ρόλοι, οι οποίοι περιγράφονται στη συνέχεια. Αυτοί είναι:

- Ο ιδιοκτήτης του προϊόντος (The Product Owner)
- Η ομάδα (The Team)
- Ο διαχειριστής ή ειδικός του Scrum (The Scrum Master)

Το σύνολο των τριών αυτών ρόλων αποτελεί την ομάδα του Scrum.

6.2.1. Ο ιδιοκτήτης του προϊόντος (Product Owner)

Ο ιδιοκτήτης του προϊόντος (Product Owner) είναι υπεύθυνος για την οικονομική αποδοτικότητα (The profitability of the project, ROI) της ανάπτυξης του προϊόντος (Hundermark, 2009), αλλά και για την αποδοτικότητα του συστήματος. Είναι αυτός που κατευθύνει την ομάδα στον σωστό στόχο. Είναι, επίσης, το μοναδικό άτομο που είναι υπεύθυνο για τη διαχείριση του «Product Backlog», της λίστας, δηλαδή, με τα χαρακτηριστικά που πρόκειται να αναπτυχθούν (θα αναλυθεί εκτενώς παρακάτω), αποφασίζοντας ποια από αυτά τα χαρακτηριστικά θα πρέπει να είναι στην κορυφή της λίστας για το επόμενο Sprint, ενώ συνεχώς αναπροσαρμόζει και καθορίζει την προτεραιότητα της λίστας αυτής. Υπάρχουν περιπτώσεις που ο Product Owner είναι ο ίδιος ο πελάτης. Σημειώνεται ότι, ενδεχομένως, ο ρόλος του μπορεί να παραλληλιστεί με τον ρόλο ενός Product Manager των παραδοσιακών μοντέλων, αν και διαφέρει αρκετά.

Επιγραμματικά, ο Product Owner είναι υπεύθυνος για τα κάτωθι:

- Ορίζει τα χαρακτηριστικά του προϊόντος.
- Αποφασίζει την ημερομηνία κυκλοφορίας και το περιεχόμενο.
- Είναι υπεύθυνος για την κερδοφορία του προϊόντος (ROI).
- Δίνει προτεραιότητα στα χαρακτηριστικά ανάλογα με την αξία στην αγορά.
- Ρυθμίζει τα χαρακτηριστικά και την προτεραιότητα σε κάθε επανάληψη, ανάλογα με τις ανάγκες.
- Αποδέχεται ή απορρίπτει τα αποτελέσματα της εργασίας.

6.2.2. Η ομάδα (The Team) ή ομάδα ανάπτυξης

Η ομάδα (The Team) ή ομάδα ανάπτυξης, είναι εκείνοι οι επαγγελματίες που πραγματώνουν το προϊόν που υποδεικνύει σε κάθε επανάληψη ο Product

Owner. Η δομή της ομάδας είναι τέτοια ώστε, με την υποστήριξη του οργανισμού, να οργανώνει και να διαχειρίζεται η ίδια τη δουλειά της, γεγονός που έχει ως αποτέλεσμα να αυξάνεται η αποδοτικότητα και η αποτελεσματικότητά της. Κάθε ομάδα στη Scrum έχει τα ακόλουθα χαρακτηριστικά:

- Η ομάδα απαρτίζεται από άτομα με διαφορετικές λειτουργικές γνώσεις που ωστόσο εργάζονται για έναν κοινό στόχο (Cross-functional). Οι γνώσεις αυτές είναι οι απαραίτητες ώστε να είναι σε θέση να παραδίδουν κάθε φορά το προϊόν, το οποίο είναι αποτέλεσμα ενός Sprint.
- Είναι αυτό-οργανωμένες ομάδες (Self-organizing) με πολύ υψηλό βαθμό αυτονομίας και υπευθυνότητας. Η ομάδα αποφασίζει το πώς θα αναπτύξει ένα προϊόν χωρίς οποιεσδήποτε υποδείξεις από τρίτους.
- Η ομάδα μπορεί να περιλαμβάνει άτομα με δεξιότητες στην ανάλυση, ανάπτυξη, δοκιμή, τον σχεδιασμό διεπαφής, τις βάσεις δεδομένων, την αρχιτεκτονική, και ούτω καθεξής, ενώ συνήθως απαρτίζεται από επτά άτομα το πολύ.
- Η ομάδα αναπτύσσει το προϊόν και δίνει ιδέες στον Product Owner για το πώς να βελτιώσει την ανάπτυξη.
- Οι ομάδες είναι περισσότερο παραγωγικές και αποτελεσματικές εάν όλα τα μέλη είναι αφοσιωμένα στην παραγωγή του προϊόντος κατά την διάρκεια ενός Sprint και αποφεύγονται ανακατατάξεις στη σύνθεση της ομάδας.
- Οι ομάδες με πολλά άτομα συνήθως οργανώνονται σε πολλαπλές μικρότερες ομάδες, η κάθε μια από τις οποίες θα αναλάβει την ανάπτυξη διαφορετικών χαρακτηριστικών του συστήματος, ενώ πάντα θα υπάρχει επικοινωνία και συντονισμός μεταξύ των ομάδων.
- Οι ομάδες αναλαμβάνουν, ώστε ένα προϊόν να περάσει από όλες τις φάσεις ανάπτυξης (σχεδιασμός, ανάλυση, προγραμματισμό και έλεγχο), όπως και στις περισσότερες μεθοδολογίες ανάπτυξης.

Αξίζει να σημειωθεί ότι εντός της ομάδας υπάρχει μόνο ένας ρόλος ουσιαστικά, αυτός του προγραμματιστή (developer) που ιδανικά ενσωματώνει τη γνώση και την εμπειρία τόσο του designer όσο και του tester. Επιπροσθέτως, στην ομάδα

δεν είναι επιθυμητή η ύπαρξη μελών «ηρώων», μελών δηλαδή, που λόγω μεγάλης εμπειρίας ή λόγω ιδιοσυγκρασίας, λαμβάνουν την πρωτοβουλία να επωμισθούν εξ ολοκλήρου κρίσιμα και σύνθετα χαρακτηριστικά και κατ' επέκταση όλο το έργο να στηρίζεται επάνω τους.

6.2.3. Ο διαχειριστής ή ειδικός του Scrum (The Scrum Master)

Ο διαχειριστής ή ειδικός του Scrum (The Scrum Master) είναι ο υπεύθυνος για τη διασφάλιση ότι η μεθοδολογία είναι κατανοητή στο σύνολό της και ακολουθείται σωστά, εξασφαλίζοντας ότι τα μέλη της Scrum υιοθετούν τη θεωρία, τους κανόνες και τις πρακτικές της μεθοδολογίας. Αποτελεί τον αρχηγό-βοηθό της ομάδας. Από τη στιγμή που τα εμπόδια και οι δυσκολίες είναι αναπόφευκτα, είναι σημαντικό να υπάρχει κάποιος που να μπορεί να δώσει ουσιαστικές λύσεις σε αυτά, ώστε η ομάδα ανάπτυξης να διατηρήσει όσο το δυνατόν ψηλότερα το επίπεδο παραγωγικότητάς της. Είναι συνηθισμένο ο Scrum Master να είναι και μέλος της ομάδας, κυρίως σε μικρές ομάδες, αν και δε συνίσταται από τη μεθοδολογία. Σε καμία περίπτωση Scrum Master δεν μπορεί να είναι ο ιδιοκτήτης του προϊόντος.

Επιγραμματικά, οι κύριες αρμοδιότητες του Scrum Master είναι οι κάτωθι:

- Εκπροσωπεί τη διαχείριση του έργου.
- Είναι υπεύθυνος για τη θέσπιση αξιών και πρακτικών Scrum.
- Αφαιρεί εμπόδια.
- Βεβαιώνει ότι η ομάδα είναι πλήρως λειτουργική και παραγωγική.
- Ενεργοποιεί τη στενή συνεργασία όλων των ρόλων και λειτουργιών.
- Θωρακίζει την ομάδα από εξωτερικές παρεμβολές.
- Βοηθάει την ομάδα να γίνει πιο παραγωγική δημιουργώντας υψηλής ποιότητας προϊόντα.

Με βάση τον επίσημο οδηγό της Scrum καθορίζονται οι εξής υπηρεσίες του Scrum Master:

Η υπηρεσία του Scrum Master προς τον Product Owner

Ο Scrum Master υπηρετεί τον Product Owner με διάφορους τρόπους, συμπεριλαμβανομένων των παρακάτω:

- Να βρίσκει τεχνικές για την αποδοτική διαχείριση του Product Backlog.
- Να βοηθάει την Ομάδα Scrum να κατανοήσει την ανάγκη για ξεκάθαρα και ακριβή στοιχεία στο Product Backlog.
- Να κατανοήσει τον σχεδιασμό προϊόντων σε ένα εμπειρικό περιβάλλον.
- Να διασφαλίσει ότι ο Product Owner ξέρει πώς να διατάσσει το Product Backlog ώστε να μεγιστοποιεί την αξία.
- Να κατανοεί και να εφαρμόζει την ευελιξία.
- Να διευκολύνει τις δραστηριότητες του Scrum, όπως του ζητηθεί ή όπως απαιτηθεί.

Η υπηρεσία του Scrum Master προς την Ομάδα Ανάπτυξης

Ο Scrum Master υπηρετεί την Ομάδα Ανάπτυξης με διάφορους τρόπους, συμπεριλαμβανομένων των παρακάτω:

- Να καθοδηγεί την Ομάδα Ανάπτυξης σε αυτό-οργάνωση και ανάπτυξη δεξιοτήτων από διαφορετικές λειτουργικές περιοχές.
- Να βοηθάει την Ομάδα Ανάπτυξης να δημιουργεί προϊόντα υψηλής αξίας.
- Να απομακρύνει τα εμπόδια στην πρόοδο της Ομάδας Ανάπτυξης.
- Να διευκολύνει τις δραστηριότητες του Scrum, όπως του ζητηθεί ή όπως απαιτηθεί.
- Να καθοδηγεί την Ομάδα Ανάπτυξης σε περιβάλλοντα οργανισμών, στα οποία το Scrum δεν έχει ακόμα κατανοηθεί και υιοθετηθεί πλήρως.

Η υπηρεσία του Scrum Master προς τον Οργανισμό

Ο Scrum Master υπηρετεί τον Οργανισμό με διάφορους τρόπους, συμπεριλαμβανομένων των παρακάτω:

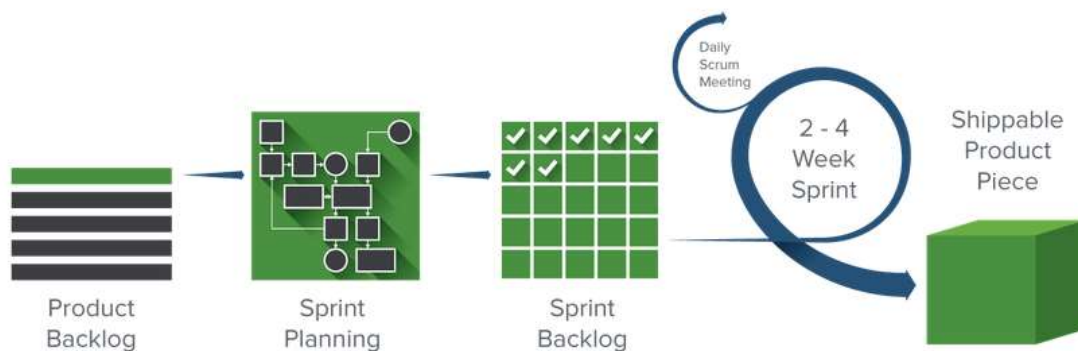
- Να ηγείται και να καθοδηγεί τον οργανισμό ως προς την υιοθέτηση του Scrum.
- Να σχεδιάζει εφαρμογές του Scrum εντός του οργανισμού.
- Να βοηθάει τους εργαζομένους και τα ενδιαφερόμενα μέρη να κατανοήσουν και να εφαρμόσουν το Scrum και την εμπειρική ανάπτυξη προϊόντων.

- Να προκαλεί αλλαγές που αυξάνουν την παραγωγικότητα της Ομάδας του Scrum.
- Να συνεργάζεται με άλλους Scrum Masters ώστε να αυξήσει την αποτελεσματικότητα της εφαρμογής του Scrum εντός του οργανισμού.

Επαναλαμβάνεται ότι στη Scrum δεν υφίσταται ο ρόλος του Project Manager για τον απλούστατο λόγο ότι δεν χρειάζεται να υπάρχει, αφού οι παραδοσιακές αρμοδιότητες ενός Project Manager έχουν διαμοιραστεί ανάμεσα στους τρεις ρόλους που περιγράφηκαν παραπάνω.

6.3. Διαδικασία της Μεθοδολογίας

Στη παρούσα παράγραφο περιγράφεται αναλυτικά η διαδικασία της μεθοδολογίας Scrum με τις δραστηριότητες που περιλαμβάνει, καθώς και τα αντικείμενα της.



Εικόνα 13. Τα βήματα της Scrum

6.3.1. Product Backlog

«Product Backlog» είναι, όπως αναφέρθηκε και στη προηγούμενη παράγραφο, η λίστα που περιέχει με σειρά προτεραιότητας οτιδήποτε μπορεί να χρειαστεί για την πραγμάτωση του προϊόντος και ταυτόχρονα αποτελεί τη μοναδική πηγή των απαιτήσεων του συστήματος. Οι απαιτήσεις μπορεί να προέλθουν από τον πελάτη, τις πωλήσεις και το τμήμα μάρκετινγκ, την υποστήριξη πελατών ή τους ίδιους τους υπεύθυνους για την ανάπτυξη του λογισμικού. Εδώ, καταγράφονται τα χαρακτηριστικά και οι λειτουργίες, οι βελτιώσεις και οι διορθώσεις που

πρέπει να γίνουν στο προϊόν στις μελλοντικές επαναλήψεις. Τα χαρακτηριστικά αυτά που περιλαμβάνονται στη λίστα, μπορούν να διατυπωθούν με οποιονδήποτε σαφή τρόπο, όπως για παράδειγμα μέσω σεναρίων Use cases ή User stories, όπως και στον ακραίο προγραμματισμό. Το «Product Backlog» δεν είναι ποτέ ολοκληρωμένο, αλλάζει δυναμικά και υπάρχει για όσο χρονικό διάστημα υπάρχει και το προϊόν.

Αρχικά, λοιπόν, ο Product Owner προσδιορίζει τον προγραμματισμό της ανάπτυξης του προϊόντος, δηλαδή δημιουργεί το Product Backlog. Ένα παράδειγμα μίας τέτοιας λίστας (ενός Product Backlog) φαίνεται στην παρακάτω εικόνα:

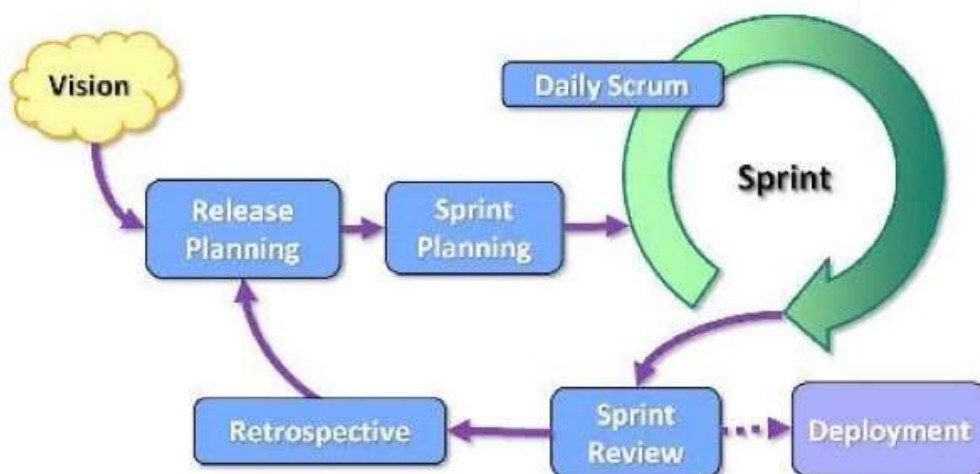
Item	Details (wiki URL)	Priority	Estimate of Value	Initial Estimate of Effort	New Estimates of Effort Remaining as of Sprint...					
					1	2	3	4	5	6
As a buyer, I want to place a book in a shopping cart (see UI sketches on wiki page)	...	1	7	5						
As a buyer, I want to remove a book in a shopping cart	...	2	6	2						
Improve transaction processing performance (see target performance metrics on wiki)	...	3	6	13						
Investigate solutions for speeding up credit card validation (see target performance metrics on wiki)	...	4	6	20						
Upgrade all servers to Apache 2.2.3	...	5	5	13						
Diagnose and fix the order processing script errors (bugzilla ID 14823)	...	6	2	3						
As a shopper, I want to create and save a wish list	...	7	7	40						
As a shopper, I want to add or delete items on my wish list	...	8	4	20						

Εικόνα 14. Ενδεικτικό παράδειγμα ενός Product Backlog (Deemer et al., 2010)

Το υποσύνολο των εργασιών ή χαρακτηριστικών του Product Backlog που επιλέγεται για μια επανάληψη, ονομάζεται Release Backlog. Όπως αναφέρθηκε και προηγουμένως, η σειρά με την οποία επιλέγονται τα χαρακτηριστικά προς πραγμάτωση εξαρτάται από την προτεραιότητα που δίνεται σε αυτά και αποτελεί κύρια ευθύνη του Product Owner. Επιπλέον, ο Product Owner είναι υπεύθυνος για την ανάθεση επιχειρηματικής αξίας σε κάθε μεμονωμένο χαρακτηριστικό που περιλαμβάνεται στη λίστα. Στο σημείο αυτό ενδεχομένως να εμπλακεί και ο Scrum Master. Έχοντας υπόψη του τις δύο αυτές εκτιμήσεις (προσπάθεια και αξία), και, επιπρόσθετα, δίνοντας έμφαση σε επικείμενους κινδύνους, ο Product Owner καθορίζει την ιεράρχηση του Product Backlog. Το σημαντικό στη διαδικασία ανάθεσης προτεραιότητας είναι να επιλέγονται πρώτα χαρακτηριστικά με μεγάλη αξία και τα οποία απαιτούν λιγότερη προσπάθεια για να υλοποιηθούν.

Πουθενά η Scrum δεν καθορίζει τυποποιημένες τεχνικές για το πώς θα γίνει η ιεράρχηση του Product Backlog. Συνήθως εκτιμώνται οι πιο σημαντικοί παράγοντες (απαιτούμενη προσπάθεια, πολυπλοκότητα, αβεβαιότητα) και δίδεται μία τιμή (βάρος) σε κάθε έναν από τους παράγοντες αυτούς. Αυτό γίνεται για όλα τα χαρακτηριστικά της λίστας και οι τιμές αυτές ονομάζονται «βαθμοί» (“points”). Φυσικά, υπάρχουν πολλές προτεινόμενες τεχνικές που αφορούν την ιεράρχηση. Μερικές από αυτές πρότεινε ο Cohn (Cohn, Prioritizing Product Backlog, 2008) και ήταν:

- Πρώτον, η ανάλυση «Κανο», η οποία περιλαμβάνει τη συμμετοχή των χρηστών και που θα αποκαλύψει ποια χαρακτηριστικά είναι σημαντικότερα και τα οποία αυξάνουν την ικανοποίηση των χρηστών, και ποια είναι τα χαρακτηριστικά που οι χρήστες δεν γνωρίζουν, ακόμη, ότι χρειάζονται.
- Δεύτερον, η προτεραιότητα μπορεί να αποδοθεί με κριτήρια επιλογής κάποιων εμπειρογνομόνων.
- Τρίτον, μπορεί να ανατεθούν βάρη στα χαρακτηριστικά με ορισμένα κριτήρια που θα επιλεγούν (αυτή είναι η πιο κοινή τεχνική και περιγράφηκε παραπάνω).



Εικόνα 15. Διαδικασίες της μεθοδολογίας Scrum

6.3.2. Sprint

Προκειμένου να ελαχιστοποιηθούν οι επιπλέον συνεδριάσεις η μεθοδολογία Scrum χρησιμοποιεί διαδικασίες χρονικά προκαθορισμένες (time-boxed

events), ώστε κάθε μία από αυτές να έχει τη μέγιστη απαραίτητη διάρκεια. Έτσι διασφαλίζεται ότι στη διαδικασία αφιερώνεται ένα ικανό χρονικό διάστημα για αποφυγή οποιασδήποτε σπατάλης χρόνου.

Η καρδιά του Scrum είναι το Sprint, ένας χρόνο-περιορισμός ενός μήνα ή λιγότερο, κατά τη διάρκεια του οποίου δημιουργείται μία "Έτοιμη", χρησιμοποιήσιμη και δυνητικά παραδοτέα Επαύξηση προϊόντος. Τα Sprints είναι καλύτερο να έχουν σταθερή διάρκεια, καθ' όλη την περίοδο ανάπτυξης. Κάθε νέο Sprint ξεκινάει αμέσως μετά την ολοκλήρωση του προηγούμενου.

Τα Sprints περιέχουν και αποτελούνται από τον Σχεδιασμό του Sprint, τα Καθημερινά Scrums, τις εργασίες ανάπτυξης, την Επισκόπηση του Sprint και την Αναδρομή στο Sprint (Ken Schwaber & Jeff Sutherland, 2015).

Κατά τη διάρκεια του Sprint:

- Δεν γίνονται αλλαγές που θα έθεταν σε κίνδυνο τον Στόχο του Sprint.
- Οι στόχοι που αφορούν θέματα ποιότητας δε μειώνονται.
- Το αντικείμενο των εργασιών μπορεί να αποσαφηνιστεί και να αποτελέσει αντικείμενο επαναδιαπραγμάτευσης μεταξύ του Product Owner και της Ομάδας Ανάπτυξης, καθώς περισσότερα πράγματα γίνονται γνωστά.

Κάθε Sprint μπορεί να θεωρηθεί ως έργο με ορίζοντα όχι μακρύτερο του ενός μήνα. Όπως τα έργα, έτσι και τα Sprints εκτελούνται με σκοπό την επίτευξη ενός στόχου. Κάθε Sprint περιέχει έναν ορισμό του τι θα δημιουργηθεί, ένα ευέλικτο πλάνο το οποίο θα καθοδηγήσει την ανάπτυξη, τη δουλειά υλοποίησης και το παραγόμενο προϊόν.

Όπως προαναφέρθηκε ο ορίζοντας του Sprint δεν πρέπει να ξεπερνάει τον ένα μήνα και τούτο διότι ο ορισμός του τι αναπτύσσεται μπορεί να μεταβληθεί, ενώ η πολυπλοκότητα και το ρίσκο ενδέχεται να αυξηθούν, αν ο χρονικός ορίζοντας γίνει πολύ μακρινός. Τα Sprints προάγουν την προβλεψιμότητα, με το να διασφαλίζουν την επιθεώρηση και την προσαρμογή της προόδου προς τον στόχο τουλάχιστον μία φορά κάθε ημερολογιακό μήνα, ενώ περιορίζουν το ρίσκο στο κόστος του ενός, αυτού, ημερολογιακού μήνα.

Αναλόγως της εμπειρίας μίας ομάδας, μπορεί να εκτιμηθεί πόση δουλειά απαιτείται για κάθε Sprint. Καθορίζεται ένας μέσος όρος από βαθμούς (points) για κάθε Sprint. Για παράδειγμα, αν ο μέσος όρος αυτός οριστεί στα 30, τότε αυτό σημαίνει πως το άθροισμα των βαθμών (points) των χαρακτηριστικών που θα επιλεγούν για ένα Release Backlog, δε θα πρέπει να υπερβαίνει τα 30. Έτσι μπορεί να εκτιμηθεί ο χρόνος που χρειάζεται για την ολοκλήρωση ενός Sprint, ή να εκτιμηθεί πόσα χαρακτηριστικά μπορούν να υλοποιηθούν σε ένα ορισμένο χρονικό διάστημα. Ο μέσος όρος αυτός ονομάζεται Velocity (ταχύτητα της ομάδας) και εκφράζεται με τον ίδιο τρόπο όπως και τα εκτιμώμενα χαρακτηριστικά του Product Backlog. Με βάση το Velocity, ουσιαστικά, καθορίζονται τα χαρακτηριστικά που θα επιλεγούν για το Product Backlog. Ο Product Owner, στη συνέχεια, χρησιμοποιεί το Velocity για να προβλέψει τον ρυθμό με τον οποίο η ομάδα είναι σε θέση να παραδίδει τμήματα του συστήματος. Αναλυτικότερα ο όρος της ταχύτητας της ομάδας (Velocity) περιγράφεται σε επόμενη παράγραφο.

6.3.3. Σχεδιασμός του Sprint (Sprint Planning)

Με την έναρξη κάθε Sprint, πραγματοποιείται το Sprint Planning Meeting (συνεδρίαση προγραμματισμού του Sprint). Η συνεδρίαση αυτή χωρίζεται σε δύο μέρη. Τα δύο μέρη του Sprint Planning Meeting απαντούν στις ακόλουθες ερωτήσεις, αντίστοιχα (Schwaber, K. and Sutherland J., 2011):

- Τι μπορεί να παραδοθεί στην Επαύξηση που θα προκύψει από το επικείμενο Sprint;
- Με ποιο τρόπο θα πραγματοποιηθεί το έργο που απαιτείται για να παραδοθεί η Επαύξηση;

Σύμφωνα με τους Sutherland & Schwaber (2011) και τον επίσημο οδηγό της Scrum τα δύο αυτά ερωτήματα αναλύονται ως εξής:

Θέμα 1: Τι μπορεί να γίνει σε αυτό το Sprint;

Η Ομάδα Ανάπτυξης εργάζεται για να προβλέψει τη λειτουργικότητα που θα αναπτυχθεί κατά τη διάρκεια του Sprint. Ο Product Owner πραγματεύεται τον στόχο που πρέπει να επιτευχθεί, καθώς και τα στοιχεία εκείνα του Product Backlog τα οποία, εάν ολοκληρωθούν μέσα στο Sprint, θα εκπληρώσουν τον

Στόχο του Sprint. Ολόκληρη η Ομάδα Scrum συνεργάζεται με σκοπό την κατανόηση των εργασιών του Sprint.

Οι είσοδοι σε αυτήν την συνάντηση είναι το Product Backlog, η τελευταία Επαύξηση του προϊόντος, η προβλεπόμενη παραγωγή της Ομάδας Ανάπτυξης για τη διάρκεια του Sprint και η προηγούμενη παραγωγή της Ομάδας Ανάπτυξης. Ο αριθμός των στοιχείων του Product Backlog που θα επιλεγθούν για το Sprint καθορίζεται αποκλειστικά από την Ομάδα Ανάπτυξης. Μόνο η Ομάδα Ανάπτυξης μπορεί να αξιολογήσει τι μπορεί να επιτύχει στο επερχόμενο Sprint.

Αφού η Ομάδα Ανάπτυξης προβλέψει τα στοιχεία του Product Backlog που θα παραδώσει στο Sprint, η Ομάδα Scrum φτιάχνει τον Στόχο του Sprint. Ο Στόχος του Sprint, ο οποίος θα επιτευχθεί μέσα στο Sprint μέσω της υλοποίησης του Product Backlog, παρέχει καθοδήγηση στην Ομάδα Ανάπτυξης σχετικά με τον σκοπό για τον οποίο αναπτύσσει την Επαύξηση.

Θέμα 2: Πώς θα εκτελεστούν οι επιλεγμένες εργασίες;

Αφού θέσει τον Στόχο του Sprint και επιλέξει τα στοιχεία του Product Backlog για το Sprint, η Ομάδα Ανάπτυξης αποφασίζει το πώς θα μετουσιώσει τη λειτουργικότητα αυτή σε μία "Ετοιμη" Επαύξηση προϊόντος κατά τη διάρκεια του Sprint. Τα στοιχεία του Product Backlog που έχουν επιλεγεί για αυτό το Sprint, μαζί με τον σχεδιασμό για την παράδοσή τους, ονομάζονται Sprint Backlog.

Η Ομάδα Ανάπτυξης συνήθως αρχίζει σχεδιάζοντας το σύστημα και τις εργασίες που χρειάζεται να εκτελέσει για να μετατρέψει το Product Backlog σε μια λειτουργική Επαύξηση προϊόντος. Οι εργασίες μπορεί να είναι διαφορετικών μεγεθών ή εκτιμώμενης προσπάθειας. Παρόλα αυτά, η δουλειά που προγραμματίζεται κατά τη διάρκεια ενός Σχεδιασμού Sprint είναι αρκετή ώστε να καλύψει όλα αυτά που η Ομάδα Ανάπτυξης πιστεύει ότι μπορεί να κάνει στο επερχόμενο Sprint. Η δουλειά που έχει προγραμματιστεί για τις πρώτες μέρες του Sprint από την Ομάδα Ανάπτυξης αναλύεται μέχρι το τέλος της συνάντησης, συχνά σε επίπεδο ημέρας ή λιγότερο. Η Ομάδα Ανάπτυξης αυτό-οργανώνεται για να διεκπεραιώσει τις εργασίες στο Sprint Backlog, τόσο

κατά τη διάρκεια του Σχεδιασμού του Sprint, όσο και κατά τη διάρκεια ολόκληρου του Sprint.

Ο Product Owner μπορεί να βοηθήσει στην αποσαφήνιση των επιλεγμένων στοιχείων του Product Backlog και να κάνει συμβιβασμούς. Εάν η Ομάδα Ανάπτυξης αποφασίσει ότι έχει πάρα πολλή ή πολύ λίγη δουλειά, μπορεί να επαναδιαπραγματευθεί τα επιλεγμένα στοιχεία του Product Backlog με τον Product Owner. Η Ομάδα Ανάπτυξης μπορεί επίσης να προσκαλέσει και άλλους ανθρώπους που θα παρέχουν τεχνικές ή εξειδικευμένες συμβουλές.

Με το τέλος του Σχεδιασμού του Sprint, η Ομάδα Ανάπτυξης θα πρέπει να είναι σε θέση να εξηγήσει στον Product Owner και τον Scrum Master με ποιον τρόπο σκοπεύει να δουλέψει ως αυτό-οργανωμένη ομάδα για να επιτύχει τον Στόχο του Sprint και να δημιουργήσει την αναμενόμενη Επαύξηση.

Product Backlog Item	Sprint Task	Volunteer	Initial Estimate of Effort	New Estimates of Effort Remaining at end of Day...				
As a buyer, I want to place a book in a shopping cart	modify database	Sanjay	5					
	create webpage (UI)	Jing	3					
	create webpage (Javascript logic)	Tracy & Sam	2					
	write automated acceptance tests	Sarah	5					
	update buyer help webpage	Sanjay & Jing	3					
	...							
Improve transaction processing performance	merge DCP code and complete layer-level tests		5					
	complete machine order for pRank		3					
	change DCP and reader to use pRank http API		5					
...					
		Total (person hours)	50					

Εικόνα 16. Παράδειγμα Sprint Backlog (Deemer et al., 2010)

Στην παραπάνω εικόνα παρουσιάζεται ένα παράδειγμα ενός Sprint Backlog. Στην πρώτη στήλη φαίνονται τα χαρακτηριστικά τα οποία επιλέγηκαν από το Product Backlog και τα οποία διασπάστηκαν σε επιμέρους Tasks, όπως αυτά φαίνονται στη δεύτερη στήλη. Τα Tasks ανατίθενται σε μέλη της ομάδας. Επιπλέον παρουσιάζεται η εκτίμηση που απαιτείται για την προσπάθεια ολοκλήρωσης του κάθε Task. Στο συγκεκριμένο παράδειγμα, η συνολική εκτίμηση για την προσπάθεια που απαιτείται είναι 50 ώρες (Deemer et al., 2010).

6.3.4. Στόχος του Sprint

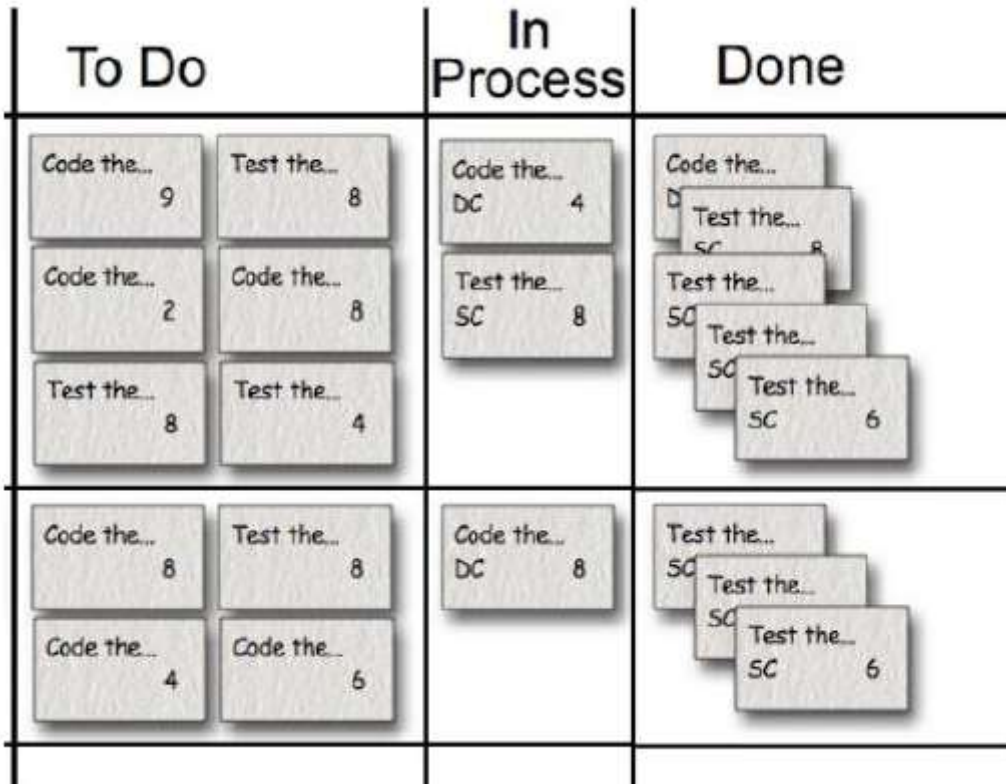
Ο Στόχος του Sprint είναι ένας σκοπός που μπορεί να επιτευχθεί μέσω της υλοποίησης του Product Backlog. Παρέχει καθοδήγηση στην Ομάδα Ανάπτυξης σχετικά με το γιατί υλοποιεί την Επαύξηση. Δημιουργείται κατά τη διάρκεια της συνάντησης για τον Σχεδιασμό του Sprint. Ο Στόχος του Sprint επιτρέπει στην Ομάδα Ανάπτυξης σχετική ευελιξία αναφορικά με τη λειτουργικότητα που θα υλοποιηθεί εντός του Sprint. Τα επιλεγμένα στοιχεία του Product Backlog σχηματίζουν μία συνεκτική λειτουργία, που μπορεί να είναι ο Στόχος του Sprint. Ο Στόχος του Sprint μπορεί να είναι οποιαδήποτε άλλος συνεκτικός στόχος βοηθάει την Ομάδα Ανάπτυξης να εργάζεται συλλογικά αντί με ατομικές πρωτοβουλίες.

Καθώς η Ομάδα Ανάπτυξης εργάζεται, λαμβάνει συνεχώς υπ' όψη τον Στόχο του Sprint. Για να εκπληρώσει τον Στόχο του Sprint, υλοποιεί τη λειτουργικότητα και την τεχνολογία. Εάν οι εργασίες εξελιχθούν σε κάτι διαφορετικό από αυτό που ανέμενε η Ομάδα Ανάπτυξης, αυτή συνεργάζεται με τον Product Owner για να διαπραγματευθεί το συνολικό εύρος του έργου που θα υλοποιηθεί εντός του Sprint.

6.3.5. Διαχείριση του Sprint

Ακόμη και αν η ομάδα χρησιμοποιεί ειδικό λογισμικό για τη διαχείριση των Sprint, είναι επωφελής να χρησιμοποιηθεί ένας πιο φυσικός τρόπος για τη διαχείριση των Tasks. Αυτό επιτυγχάνεται με τη χρήση ενός πίνακα εργασιών (Task Board) στον οποίο καταγράφονται οι εργασίες ενός Sprint σε μορφή σημειώσεων. Ουσιαστικά είναι μία φυσική αναπαράσταση του καταλόγου των εργασιών που η ομάδα έχει δεσμευθεί να υλοποιήσει κατά τη διάρκεια του τρέχοντος Sprint. Κάθε Task είναι γραμμένο σε ένα αυτοκόλλητο, συμπεριλαμβανομένης της εκτιμώμενης προσπάθειας που απαιτείται, π.χ., σε ώρες. Η πρώτη στήλη στον πίνακα περιλαμβάνει τις εργασίες που δεν έχουν ελεγχθεί ακόμη (Not checked out ή To do), δηλαδή τις εργασίες για τις οποίες δεν εργάζεται ακόμη κανένας. Η δεύτερη στήλη περιλαμβάνει τις εργασίες για τις οποίες δουλεύει ήδη η ομάδα και συνεπώς βρίσκονται σε εξέλιξη (Checked out ή In process). Τέλος, η τρίτη στήλη είναι για τις εργασίες που έχουν

ολοκληρωθεί (Done ή Completed). Έτσι, τα αυτοκόλλητα μετακινούνται από τα αριστερά προς τα δεξιά (Kniber, 2007).



Εικόνα 17. Sprint Task Board (Hundermark, 2009)

Στην εικόνα βλέπουμε τις εργασίες που έχουν προγραμματιστεί για αυτό το Sprint, καθώς και την τρέχουσα κατάσταση της προόδου.

Ένας από τους βασικούς κανόνες της Scrum είναι ότι, όταν η ομάδα έχει ήδη ξεκινήσει με την υλοποίηση ενός Sprint, τότε τυχόν προσθήκες ή αλλαγές θα πρέπει να αναβληθούν μέχρι το επόμενο Sprint. Αν δηλαδή ο Product Owner αποφασίσει ότι θα ήθελε η ομάδα να πραγματοποιήσει και ένα άλλο χαρακτηριστικό, δεν μπορεί να κάνει αυτήν την προσθήκη ή την αλλαγή μέχρι την έναρξη του επόμενου Sprint.

6.3.6. Ακύρωση του Sprint

Ενδέχεται ένα Sprint να τερματιστεί πρόωρα εφόσον προκύψουν νέα δεδομένα τα οποία αλλάζουν σημαντικά τις προτεραιότητες και η ομάδα σπαταλάει πολύτιμο χρόνο εάν συνεχίσει να εργάζεται στο τρέχων Sprint και ο Στόχος του Sprint γίνει παρωχημένος. Μόνο ο Product Owner έχει την εξουσία να ακυρώσει

ένα Sprint, παρότι αυτό μπορεί να το κάνει μετά από πρωτοβουλία ή προτροπή των ενδιαφερομένων μερών, της Ομάδας Ανάπτυξης, ή του Scrum Master. Λόγω, ωστόσο, της μικρής διάρκειας των Sprints, η ακύρωση σπάνια έχει νόημα.

Όταν ένα Sprint ακυρώνεται, γίνεται επισκόπηση όλων των "Έτοιμων" στοιχείων του Product Backlog. Εάν μέρος της δουλειάς είναι δυνητικά παραδοτέο, ο Product Owner συνήθως το αποδέχεται. Όλα τα ανολοκλήρωτα στοιχεία του Product Backlog επανεκτιμώνται και τοποθετούνται πίσω στο Product Backlog. Η δουλειά που έχει γίνει πάνω σε αυτά χάνει γρήγορα την αξία της, και πρέπει συχνά να επανεκτιμάται. Οι ακυρώσεις των Sprints καταναλώνουν πόρους, καθώς όλοι πρέπει να ανασυνταχθούν για να πάνε σε ένα νέο Σχεδιασμό Sprint, ώστε να ξεκινήσουν ένα νέο Sprint. Οι ακυρώσεις είναι συχνά τραυματικές για την Ομάδα Scrum και δεν είναι καθόλου συνηθισμένες.

6.3.7. Καθημερινή συνεδρίαση Scrum (Daily Scrum)

Όταν πλέον το Sprint έχει αρχίσει τότε η ομάδα συμμετέχει σε μια ακόμη από τις βασικές πρακτικές της Scrum, την καθημερινή συνεδρίαση Scrum (The Daily Scrum Meeting). Είναι μια περιορισμένη, χρονικά, δραστηριότητα 15 λεπτών η οποία επιτρέπει στην Ομάδα Ανάπτυξης να συγχρονίσει τις δραστηριότητές της και να δημιουργήσει ένα πλάνο για τις επόμενες 24 ώρες. Τούτο γίνεται με την επιθεώρηση των εργασιών που έγιναν από το τελευταίο Καθημερινό Scrum (ή καθημερινή συνεδρίαση Scrum αλλιώς), καθώς και με την πρόβλεψη των εργασιών που θα μπορούσαν να γίνουν πριν από το επόμενο Καθημερινό Scrum. Το Καθημερινό Scrum λαμβάνει χώρα κάθε μέρα, στον ίδιο τόπο και ώρα, ώστε να μειωθεί η πολυπλοκότητα. Κατά τη διάρκεια της συνάντησης, τα μέλη της Ομάδας Ανάπτυξης εξηγούν:

- Τι έκανα χθες, που βοήθησε την Ομάδα Ανάπτυξης να επιτύχει τον Στόχο του Sprint;
- Τι θα κάνω σήμερα για να βοηθήσω την Ομάδα Ανάπτυξης να επιτύχει τον Στόχο του Sprint;

- Βλέπω κάποιο εμπόδιο το οποίο αποτρέπει εμένα ή την Ομάδα Ανάπτυξης από την επίτευξη του Στόχου του Sprint;

Ουσιαστικά με το Καθημερινό Scrum η Ομάδα Ανάπτυξης επιθεωρεί την πρόοδο προς τον Στόχο του Sprint καθώς και την εξέλιξη της προόδου προς την ολοκλήρωση των εργασιών του Sprint Backlog. Μετά το καθημερινό Scrum πιθανόν να ακολουθήσουν συναντήσεις μεταξύ των μελών της ομάδας για λεπτομερείς συζητήσεις, ή για προσαρμογή, ή για επανασχεδιασμό του υπολοίπου των εργασιών του Sprint.

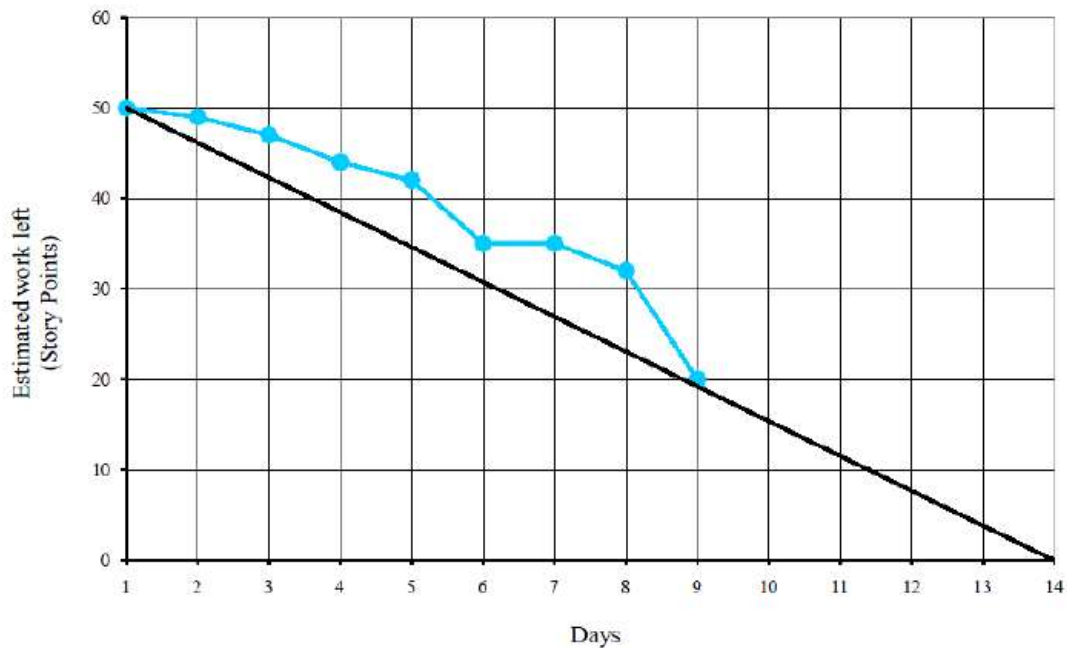
Η συνάντηση είναι σημαντική όσον αφορά την αυτό-οργάνωση της ομάδα (Deemer et al., 2010), όπως επίσης σημαντική είναι και η ανανέωση του Task Board με τα καινούρια στοιχεία που έχουν προκύψει. Με τον τρόπο αυτό ενημερώνεται η τρέχουσα κατάσταση της προόδου και η ομάδα προγραμματίζει τις αμέσως επόμενες ενέργειές της (Kniber, 2007).

Ο Scrum Master διασφαλίζει ότι η Ομάδα Ανάπτυξης θα κάνει τη συνάντηση, αλλά υπεύθυνη για τη διενέργεια του Καθημερινού Scrum, είναι η ίδια η Ομάδα Ανάπτυξης. Ο Scrum Master μαθαίνει την Ομάδα Ανάπτυξης να διατηρεί το Καθημερινό Scrum εντός του χρονικού περιορισμού των 15 λεπτών. Επίσης επιβάλλει ότι μόνο τα μέλη της Ομάδας Ανάπτυξης συμμετέχουν. Η καθημερινή συνεδρίαση Scrum είναι μια καίρια συνάντηση επιθεώρησης και προσαρμογής.

6.3.7.1. Παρακολούθηση της προόδου προς τον Στόχο του Sprint

Το συνολικό έργο που απομένει στο Sprint Backlog μπορεί να αθροιστεί ανά πάσα στιγμή, κατά τη διάρκεια του Sprint. Η Ομάδα Ανάπτυξης παρακολουθεί αυτό το συνολικό υπολειπόμενο έργο, τουλάχιστον για το Καθημερινό Scrum, ώστε να προβάλει την πιθανότητα επίτευξης του Στόχου του Sprint. Με την παρακολούθηση του εναπομείναντος έργου καθ' όλη την διάρκεια του Sprint, η Ομάδα Ανάπτυξης μπορεί να διαχειριστεί την πρόοδό της (Schwaber, K. and Sutherland J., 2011). Τα μέλη της ομάδας, δηλαδή, ενημερώνουν το εκτιμώμενο υπόλοιπο του χρόνου που χρειάζεται για να ολοκληρώσουν την εργασία τους. Μετά από αυτήν την ενημέρωση, προκύπτει μια συνολική

ανανέωση του χρόνου που απομένει, ώστε να ολοκληρωθεί ολόκληρο το Sprint. Αποτέλεσμα αυτής της ενημέρωσης είναι ένα γράφημα το οποίο ονομάζεται Sprint Burndown Chart. Το γράφημα αυτό αναπαριστά την υπόλοιπη εργασία που απομένει για ένα Sprint Backlog. Ενημερώνεται καθημερινά και παρουσιάζει με έναν απλό τρόπο την πρόοδο κάθε Sprint. Το διάγραμμα αυτό παρουσιάζει, κάθε μέρα, μια νέα εκτίμηση για το πόση δουλειά απομένει μέχρι η ομάδα να ολοκληρώσει όλα τα Tasks σε ένα Sprint. Ονομάζεται «Burndown» αφού αποκαλύπτει τον φόρτο εργασίας που απομένει σαν συνάρτηση του χρόνου σε μια καμπύλη που έχει κλίση προς τα κάτω. Η γραφική αυτή παράσταση ενημερώνεται κατά τη διάρκεια του Daily Scrum Meeting (Kniber, 2007) και απεικονίζεται παρακάτω.



Εικόνα 18. Παράδειγμα Sprint Burndown Chart [Deemer et al., 2010]

Στην παραπάνω εικόνα απεικονίζεται ένα Burndown Chart για Sprint διάρκειας 14 ημερών. Στον άξονα των Χ εμφανίζεται ο χρόνος εργασίας σε ημέρες και στον Υ παρουσιάζεται το ποσοστό της υπολειπόμενης εργασίας. Για κάθε μέρα η τρέχουσα κατάσταση σημειώνεται στο γράφημα με μία τελεία και κάθε μία ενώνεται με αυτήν της προηγούμενης μέρας με μία γραμμή (η μπλε γραμμή του σχήματος). Επίσης, υπάρχει μια γραμμή η οποία ονομάζεται «ιδανική γραμμή» (Idealized Line) και η οποία αναπαριστά την ιδανική πρόοδο που η ομάδα θα έπρεπε να πετύχει (στο σχήμα παρουσιάζεται με μαύρη γραμμή). Αυτή υπάρχει

πάντα πάνω στο γράφημα ώστε να μπορεί να υπάρχει άμεσα μια σύγκριση της παρούσας κατάστασης της προόδου με την ιδανική. Η γραμμή αντιπροσωπεύει μια γραμμική πρόοδο από την πρώτη μέρα του Sprint μέχρι και την τελευταία μέρα. Έτσι, η γραμμή ξεκινά από το σημείο όπου καθόλου δουλειά δεν έχει γίνει και τελειώνει στο σημείο στο οποίο δεν έχει απομείνει δουλειά κατά την τελευταία ημέρα.

Τέλος, ενώ το Burndown Chart μπορεί να δημιουργηθεί και να προβληθεί χρησιμοποιώντας ένα ειδικό λογισμικό (π.χ Excel), εντούτοις πολλές ομάδες διαπίστωσαν ότι είναι πιο αποτελεσματικό να χρησιμοποιούν έναν πιο φυσικό τρόπο αναπαράστασης του γραφήματος, όπως πάνω σε έναν πίνακα στον χώρο εργασίας τους. Η λύση αυτή είναι γρήγορη, απλή, και συχνά πιο ορατή από ένα διάγραμμα στον υπολογιστή. Συνήθως, το Burndown Chart τοποθετείται κοντά στο Task Board ώστε να είναι προσβάσιμο και ορατό από όλους ανά πάσα στιγμή (Schwaber K. , 2008).

6.3.7.2. Επαύξηση

Σύμφωνα με τους δημιουργούς την μεθοδολογίας Scrum και τον οδηγό που έχουν εκδώσει, η Επαύξηση είναι το άθροισμα όλων των στοιχείων του Product Backlog που ολοκληρώθηκαν κατά τη διάρκεια ενός Sprint και της αξίας των επαυξήσεων όλων των προηγούμενων Sprints. Στο τέλος ενός Sprint, η νέα Επαύξηση πρέπει να είναι "Ετοιμη", με την έννοια ότι πρέπει να βρίσκεται σε λειτουργική κατάσταση και να τηρεί τον ορισμό του "Ετοιμου" για την Ομάδα Scrum. Πρέπει να βρίσκεται σε λειτουργική κατάσταση ανεξάρτητα του εάν ο Product Owner αποφασίσει να την εκδώσει ή όχι. Η κάθε Επαύξηση είναι προσθετική σε όλες τις προηγούμενες Επαυξήσεις και λεπτομερώς δοκιμασμένη, ώστε να εξασφαλιστεί ότι όλες οι Επαυξήσεις μαζί λειτουργούν.

6.3.8. Επισκόπηση του Sprint (Sprint Review)

Στο τέλος του Sprint, διοργανώνεται η συνάντηση Επισκόπησης του Sprint, με σκοπό την επιθεώρηση της Επαύξησης και την προσαρμογή του Product Backlog, εφόσον απαιτείται. Με βάση τα αποτελέσματα και τις όποιες αλλαγές προέκυψαν στο Product Backlog κατά τη διάρκεια του Sprint, η Ομάδα Scrum και τα ενδιαφερόμενα μέρη συνεργάζονται πάνω στο επόμενο πράγματα που

θα μπορούσαν να γίνουν ώστε να βελτιστοποιηθεί η αξία του προϊόντος. Πρόκειται για ανεπίσημη συνάντηση, όχι για αναφορά προόδου, ενώ η παρουσίαση της Επαύξεσης γίνεται με σκοπό την ενθάρρυνση υποβολής σχολίων και την προαγωγή της συνεργασίας.

Η επισκόπηση είναι περιορισμένη στις τέσσερις ώρες για μηνιαία Sprints. Για Sprints μικρότερης διάρκειας, είναι συνήθως ακόμα συντομότερη. Ο Scrum Master φροντίζει η επισκόπηση να γίνεται πάντα και οι συμμετέχοντες να κατανοούν πλήρως το σκοπό της. Επίσης, μαθαίνει σε όλους πώς να την ολοκληρώνουν μέσα στα προκαθορισμένα χρονικά πλαίσια.

Τι περιλαμβάνει, συνοπτικά, η Επισκόπηση του Sprint:

- Στη συνάντηση συμμετέχει η Ομάδα Scrum, καθώς και τα κύρια ενδιαφερόμενα μέρη, οι οποίοι προσκαλούνται από τον Product Owner.
- Ο Product Owner εξηγεί ποια από τα στοιχεία του Product Backlog είναι, και ποια δεν είναι “Έτοιμα”.
- Η Ομάδα Ανάπτυξης συζητά τι πήγε καλά κατά τη διάρκεια του Sprint, τι προβλήματα αντιμετώπισε και πώς αυτά επιλύθηκαν.
- Η Ομάδα Ανάπτυξης επιδεικνύει τα “Έτοιμα” αποτελέσματα της δουλειάς της και απαντά σε ερωτήσεις σχετικές με την Επαύξεση.
- Ο Product Owner αναλύει το Product Backlog στην τωρινή του κατάσταση. Επίσης, εφόσον απαιτείται, αναφέρει τις προβλεπόμενες ημερομηνίες ολοκλήρωσης με βάση τη μέχρι τώρα πρόοδο.
- Όλοι οι συμμετέχοντες συνεργάζονται με θέμα τις επόμενες ενέργειες. Έτσι, η Επισκόπηση του Sprint παρέχει πολύτιμα στοιχεία στον επερχόμενο Σχεδιασμό του Sprint.
- Επισκόπηση του πώς η αγορά ή οι δυνητικές χρήσεις του προϊόντος μπορεί να έχουν μεταβάλει το ποιο είναι το επόμενο πράγμα που έχει μεγαλύτερη αξία να γίνει.
- Επισκόπηση του χρονοδιαγράμματος, του προϋπολογισμού, των δυνατοτήτων και της αγοράς για την επόμενη προβλεπόμενη έκδοση του προϊόντος.

Τελικώς, το αποτέλεσμα της Επισκόπησης του Sprint είναι το αναθεωρημένο Product Backlog το οποίο καθορίζει τα επικρατέστερα στοιχεία για υλοποίηση στο επόμενο Sprint. Το Product Backlog μπορεί να αναπροσαρμοστεί συνολικά, ώστε να μας επιτρέπει να εκμεταλλευθούμε νέες ευκαιρίες.

6.3.9. Αναδρομή στο Sprint (Sprint Retrospective)

Η Αναδρομή στο Sprint είναι μια διαδικασία που ακολουθεί της Επισκόπησης του Sprint (Sprint Review). Το Sprint Review αφορά την εξέταση του παραγομένου αποτελέσματος, δηλαδή του προϊόντος. Η Αναδρομή στο Sprint αφορά την εξέταση και αναθεώρηση της διαδικασίας που ακολουθήθηκε κατά το τελευταίο Sprint. Η Αναδρομή στο Sprint είναι μια ευκαιρία για την Ομάδα Scrum να επιθεωρήσει τον εαυτό της και να δημιουργήσει ένα πλάνο βελτιώσεων το οποίο θα εκτελέσει κατά τη διάρκεια του επόμενου Sprint. Είναι μια τρίωρη χρονοπεριορισμένη συνάντηση για μηνιαία Sprints. Για Sprint μικρότερης διάρκειας είναι συνήθως μικρότερη. Ο Scrum Master διασφαλίζει ότι η δραστηριότητα γίνεται και ότι οι συμμετέχοντες κατανοούν τον σκοπό της. Επίσης, μαθαίνει σε όλους πώς να την ολοκληρώνουν μέσα στα προκαθορισμένα χρονικά πλαίσια. Ο Scrum Master συμμετέχει στη συνάντηση ως ισότιμο μέλος της ομάδας με την ιδιότητα του υπεύθυνου της εφαρμογής της διαδικασίας του Scrum.

Ο σκοπός της Αναδρομής στο Sprint είναι (Schwaber, K. and Sutherland J., 2011):

- Να επιθεωρήσει πώς πήγε το τελευταίο Sprint όσον αφορά τους ανθρώπους, τις σχέσεις, τη διαδικασία και τα εργαλεία.
- Να εντοπίσει και να ταξινομήσει τα πιο σημαντικά στοιχεία που πήγαν καλά, καθώς και πιθανές βελτιώσεις, και
- Να δημιουργήσει ένα πλάνο εφαρμογής των βελτιώσεων αναφορικά με τον τρόπο που λειτουργεί η Ομάδα Scrum. Ο Scrum Master ενθαρρύνει την Ομάδα Scrum να βελτιώνει, στο πλαίσιο της διαδικασίας Scrum, τις πρακτικές και τη διαδικασία ανάπτυξής της, προκειμένου να γίνουν πιο αποτελεσματικές και ευχάριστες στο επόμενο Sprint. Κατά τη διάρκεια κάθε Αναδρομής στο Sprint, η Ομάδα Scrum σχεδιάζει τρόπους για να

αυξήσει την ποιότητα του προϊόντος προσαρμόζοντας κατάλληλα τον ορισμό του "Έτοιμο".

Μέχρι το τέλος της Αναδρομής στο Sprint, η Ομάδα Scrum θα πρέπει να έχει εντοπίσει βελτιώσεις τις οποίες θα εφαρμόσει στο επόμενο Sprint. Η εφαρμογή αυτών των βελτιώσεων στο επόμενο Sprint αποτελεί την προσαρμογή στην επιθεώρηση αυτής καθ' αυτής της Ομάδας Scrum. Αν και βελτιώσεις μπορούν να εφαρμόζονται ανά πάσα στιγμή, η Αναδρομή στο Sprint δίνει μία θεσμοθετημένη ευκαιρία στην ομάδα να εστιάσει στην επιθεώρηση και την προσαρμογή.

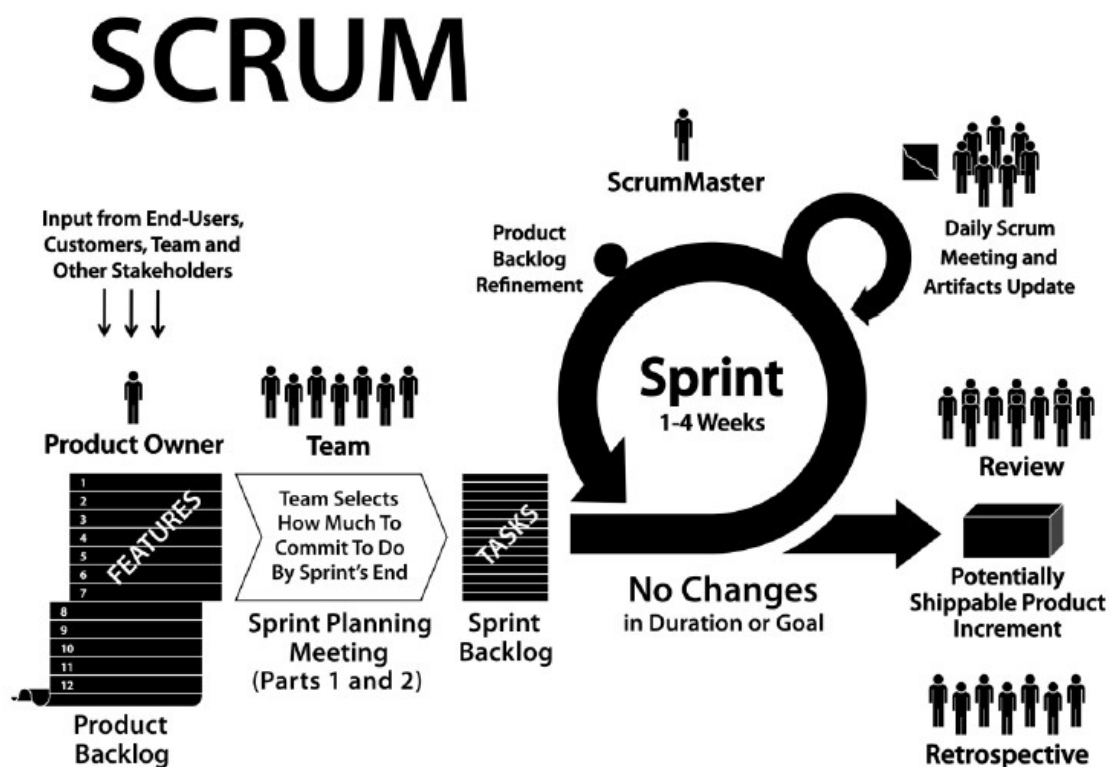
6.3.10. Ορισμός του «Έτοιμου» (Definition of Done)

Όταν ένα στοιχείο του Product Backlog ή της Επαύξεσης χαρακτηρίζεται ως "Έτοιμο" ('Done'), όλοι πρέπει να καταλαβαίνουν τι σημαίνει "Έτοιμο". Παρόλο που αυτό διαφέρει σημαντικά από ομάδα σε ομάδα, τα μέλη της ίδιας ομάδας πρέπει να κατανοούν με τον ίδιο τρόπο το τι σημαίνει η δουλειά να είναι ολοκληρωμένη, ώστε να διασφαλίζεται η διαφάνεια. Αυτός είναι ο ορισμός του "Έτοιμου" (Definition of Done) για την Ομάδα Scrum και χρησιμοποιείται για να αξιολογηθεί εάν οι εργασίες που εκτελούνται επάνω στην Επαύξηση του προϊόντος έχουν ολοκληρωθεί.

Ο ίδιος ορισμός καθοδηγεί την Ομάδα Ανάπτυξης να γνωρίζει πόσα από τα στοιχεία του Product Backlog μπορεί να επιλέξει κατά τη διάρκεια του Σχεδιασμού του Sprint. Ο σκοπός του κάθε Sprint είναι να παραδοθούν επαυξήσεις δυνητικά παραδοτέας λειτουργικότητας, που να τηρούν τον τρέχοντα ορισμό του "Έτοιμου" της Ομάδας Scrum. Οι Ομάδες Ανάπτυξης παραδίδουν μία Επαύξηση λειτουργικότητας του προϊόντος σε κάθε Sprint. Αυτή η Επαύξηση είναι αξιοποιήσιμη και έτσι ο Product Owner μπορεί να επιλέξει να την εκδώσει αμέσως. Εάν ο ορισμός του "Έτοιμου" για μία Επαύξηση είναι μέρος των συμβάσεων, των προτύπων και των κατευθυντήριων γραμμών του οργανισμού ανάπτυξης, όλες οι Ομάδες Scrum πρέπει να τον ακολουθούν κατ' ελάχιστον. Εάν το "Έτοιμο" για μία Επαύξηση δεν αποτελεί σύμβαση για τον οργανισμό ανάπτυξης, η Ομάδα Ανάπτυξης θα πρέπει να θεσπίσει έναν ορισμό "Έτοιμου", κατάλληλο για το προϊόν. Εάν

υπάρχουν πολλαπλές Ομάδες Scrum που δουλεύουν στην έκδοση του συστήματος ή προϊόντος, οι Ομάδες Ανάπτυξης σε όλες τις Ομάδες Scrum πρέπει από κοινού να δημιουργήσουν τον ορισμό του "Έτοιμου".

Όπως σαφώς αναφέρουν οι δημιουργοί της μεθοδολογίας, όσο οι Ομάδες Scrum ωριμάζουν, οι ορισμοί τους για το τι είναι "Έτοιμο" θα επεκταθούν, ώστε να συμπεριλάβουν αυστηρότερα κριτήρια, για υψηλότερη ποιότητα. Το κάθε ένα προϊόν ή σύστημα θα πρέπει να έχει έναν ορισμό "Έτοιμου", το οποίο να αποτελεί το πρότυπο για οποιοδήποτε έργο εκτελείται επάνω σε αυτό.



Εικόνα 19. Διαδικασία Scrum (Deemer et al., 2010)

6.3.11. Ταχύτητα (Velocity)

Η ταχύτητα αποτελεί τη βασική μετρική παρακολούθησης της εξέλιξης του έργου στη μεθοδολογία Scrum. Η μέτρηση της ταχύτητας γίνεται σε επίπεδο Sprint και γίνεται είτε με άθροιση των πόντων από τις ιστορίες χρηστών (User Story) σε κάθε Sprint είτε με την άθροιση των πραγματικών ανθρωποωρών που εκτελούνται από την ομάδα στα πλαίσια ενός Sprint.

Ο Layton (Layton, 2012) περιέγραψε τη διαδικασία για πρόβλεψη της ταχύτητας:

1. Αρχικά αθροίζονται οι εκτιμώμενες ανθρωποώρες για τις απαιτήσεις που παραμένουν προς εκτέλεση στο Product Backlog.
2. Διαιρείται το παραπάνω άθροισμα με τη μέση ταχύτητα, ώστε να προσδιοριστούν τα Sprints που απομένουν για την ολοκλήρωση του έργου. Υπάρχει και η εναλλακτική επιλογή πρόβλεψης καλύτερου και χειρότερου σεναρίου με χρησιμοποίηση μέγιστης και ελάχιστης ταχύτητας αντίστοιχα.
3. Πολλαπλασιασμός του αριθμού των Sprint που προέκυψε από την παραπάνω πράξη, με τη διάρκεια σε εβδομάδες του Sprint. Το αποτέλεσμα αποτελεί τις εβδομάδες που απομένουν για την ολοκλήρωση τη δεδομένη χρονική στιγμή όλων των απαιτήσεων του Backlog.

Σκοπός της μετρικής της ταχύτητας είναι η ομάδα έργου να μπορεί να ρυθμίζει την ταχύτητά της, ώστε να επιτύχει να ολοκληρώσει την πραγμάτωση των απαιτήσεων εντός του ημερολογιακού χρόνου που ορίστηκε. Φυσικά η διατήρηση της ταχύτητας καθώς και η αύξησή της επηρεάζεται από εξωτερικούς παράγοντες, όπως οι άδειες του προσωπικού, η απόσπαση του προσωπικού με άλλες εργασίες κτλ.

Θα πρέπει να σημειωθεί ότι σύμφωνα με τον Cohn (Cohn, Succeeding with Agile - Software Development using Scrum, 2012) για να αποτελεί η ταχύτητα αξιόπιστο μέγεθος πρόβλεψης ενός χρονοδιαγράμματος πραγμάτωσης ενός έργου θα πρέπει να συντρέχουν τα εξής:

- Τα Sprint να έχουν σταθερή διάρκεια σε όλη τη διάρκεια του έργου, ώστε να μπορεί να γίνει σύγκριση μεταξύ τους
- Σταθερός χρόνος διαθεσιμότητας (ανθρωποχρόνος) κάθε μέλους της ομάδας σε κάθε Sprint.
- Σταθερή ομάδα ανάπτυξης του έργου.

Είναι αλήθεια ότι δύσκολα μπορεί να τηρηθούν τα παραπάνω σε ένα έργο, ωστόσο η ταχύτητα είναι ότι πιο αξιόπιστο υπάρχει στη μεθοδολογία Scrum για πρόβλεψη του χρονοδιαγράμματος ολοκλήρωσης ενός έργου.

6.4. Διαφάνεια Αντικειμένων της Scrum

Η μεθοδολογία Scrum βασίζεται στη διαφάνεια, στο ότι, δηλαδή, τμήματα της διαδικασίας πρέπει να είναι ορατά και κατανοητά σε αυτούς που είναι υπεύθυνοι για το αποτέλεσμα. Η διαφάνεια, ουσιαστικά, απαιτεί τα τμήματα αυτά να είναι ορισμένα πάνω σε ένα κοινό πρότυπο, ώστε αυτοί που παρακολουθούν να έχουν μία κοινή άποψη για το τι ακριβώς παρακολουθούν. Με πιο απλά λόγια, θα πρέπει να «μιλούν την ίδια γλώσσα».

Οι αποφάσεις που στοχεύουν στη βελτιστοποίηση της αξίας και τον έλεγχο του ρίσκου λαμβάνονται με βάση την αντιληπτή κατάσταση των διαφόρων αντικειμένων. Στο βαθμό εκείνο που η διαφάνεια είναι πλήρης, οι αποφάσεις έχουν ορθή βάση. Στο βαθμό εκείνο που τα στοιχεία δεν είναι πλήρως διαφανή, οι αποφάσεις μπορεί να είναι λανθασμένες, η αξία μπορεί να μειωθεί και το ρίσκο να αυξηθεί.

Ο Scrum Master πρέπει να δουλέψει με τον Product Owner, την Ομάδα Ανάπτυξης και τα άλλα ενδιαφερόμενα μέρη για να καταλάβει εάν τα αντικείμενα είναι απολύτως διαφανή. Υπάρχουν πρακτικές για να αντιμετωπιστεί η μη-πλήρης διαφάνεια και ο Scrum Master πρέπει να βοηθήσει όλους να εφαρμόσουν τις πιο κατάλληλες πρακτικές όταν δεν υπάρχει πλήρης διαφάνεια. Ένας Scrum Master μπορεί να διακρίνει την έλλειψη διαφάνειας με την επιθεώρηση των αντικειμένων, με την παρατήρηση μοτίβων, ακούγοντας προσεκτικά το τι λέγεται και διακρίνοντας τις διαφορές μεταξύ προσδοκώμενων και πραγματικών αποτελεσμάτων.

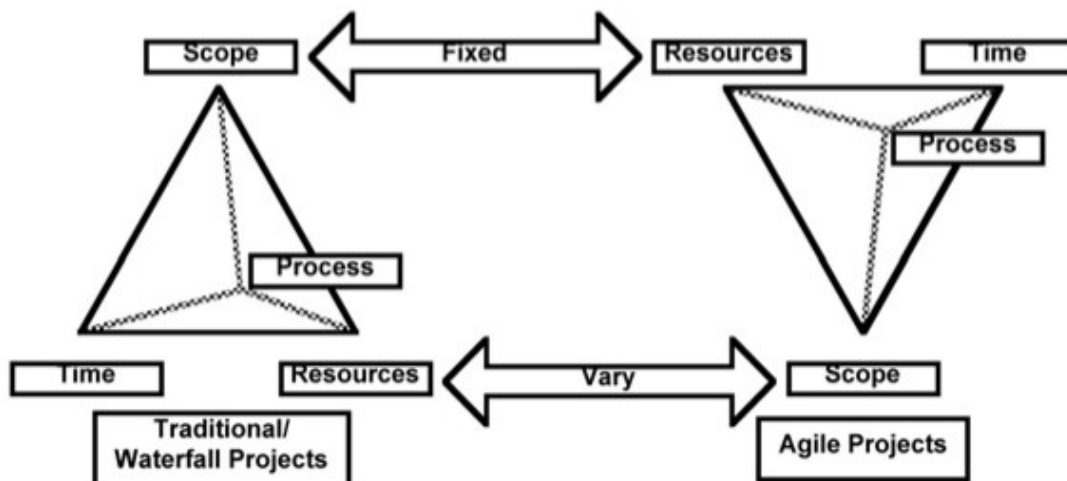
Η δουλειά του Scrum Master είναι να εργαστεί με την Ομάδα Scrum και τον οργανισμό για να αυξήσει τη διαφάνεια των αντικειμένων του Scrum. Η δουλειά αυτή συνήθως περιλαμβάνει μάθηση, πειθώ και αλλαγή.

«Η διαφάνεια δε συμβαίνει εν μία νυκτί. Είναι ένα μονοπάτι» (Ken Schwaber & Jeff Sutherland, 2015).

7. Σύγκριση Μεθοδολογίας Scrum με «Παραδοσιακές» Μεθοδολογίες

Σύμφωνα με τον Sauer (Sausser B. et al., 2009), κάθε έργο είναι μοναδικό και απαιτεί εξειδικευμένο τρόπο διαχείρισης. Εδώ και αρκετά χρόνια, οι παραδοσιακές μεθοδολογίες χρησιμοποιούνταν ευρέως και η επιτυχία τους στη βιομηχανία λογισμικού έχει επισημανθεί από διάφορους μελετητές (Parke-Shields, Beise, K.E., Quan, J., 2009). Η αυξανόμενη πίεση για παράδοση όλο και πιο γρήγορα ποιοτικών προϊόντων λογισμικού, ήταν η αιτία εμφάνισης των ευέλικτων μεθοδολογιών. Όπως αναφέρθηκε και περιγράφηκε στις προηγούμενες παραγράφους η ευέλικτη ανάπτυξη, σε αντίθεση με τις παραδοσιακές προσεγγίσεις, δίνει μεγαλύτερη έμφαση στην επικοινωνία και όχι σε μια διαδικασία ή σχέδιο και ως εκ τούτου οι δύο διαφέρουν ως προς το πεδίο εφαρμογής τους.

Ο Owen (Owen, R., Koskela, L.J., Henrich, G., and Codinhoto, R., 2006), όρισε την εννοιολογική διαφορά μεταξύ παραδοσιακών και ευέλικτων πρακτικών. Ενώ, λοιπόν, στην παραδοσιακή διαχείριση έργου το πεδίο εφαρμογής (Scope) παραμένει σταθερό, στις ευέλικτες μεθοδολογίες οι πόροι του έργου και ο χρόνος θεωρούνται σταθερές, ενώ το πεδίο εφαρμογής μεταβάλλεται.



Εικόνα 20. Διαφορά Παραδοσιακών και Ευέλικτων Μεθοδολογιών (Owen, R., Koskela, L.J., Henrich, G., and Codinhoto, R., 2006)

Ενώ για δεκαετίες, τα έργα ανάπτυξης λογισμικού ακολουθούσαν το κλασικό μοντέλο του καταρράκτη, σήμερα, από όλες τις ευέλικτες μεθοδολογίες, η

Scrum είναι αυτή που ξεχωρίζει, γιατί εισάγαγε την ιδέα του εμπειρικού ελέγχου των διαδικασιών.

Παρακάτω θα αναφερθούν, σε μορφή πίνακα, οι σημαντικότερες διαφορές των δύο μεθοδολογιών για όλες τις λειτουργίες και διαδικασίες τους όπως περιγράφηκαν από τον Cocco και τον Loeser (Cocco, L., Mannaro, K., Concas, G. and Marchesi, M., 2001) (Loeser, 2006).

Κριτήριο	Μοντέλο Καταρράκτη	Scrum
Διαδικασίες		
<i>Προγραμματισμός Έργου (Σκοπός, πρόγραμμα, επικοινωνίες και ανθρώπινοι πόροι)</i>	<p>1. Δεν υπάρχει αντίστοιχη αναφορά στο μοντέλο του καταρράκτη.</p> <p>2. Δεν υπάρχει αντίστοιχη αναφορά στο μοντέλο του καταρράκτη.</p>	<p>1. Vision: Μία σύντομη αναφορά για την κατεύθυνση του έργου (πραγματοποιείται από τον Product Owner)</p> <p>2. Roadmap: Μία σύντομη αναφορά για τις υψηλού προτεραιότητας λειτουργίες που πρέπει να υλοποιηθούν (γράφεται από Product Owner)</p>
	<p>3. Μόλις ένα έργο έχει οριστικοποιηθεί, ο Project Manager (PM) ηγείται της προδιαγραφής απαιτήσεων και του υπολογισμού της διάρκειας των εργασιών.</p> <p>4. Ο PM δημιουργεί το Project Plan και το αναθεωρεί με την ομάδα διαχείρισης του έργου.</p> <p>5. Ο PM συναντιέται με την ομάδα έργου περιοδικά (σύμφωνα με το πλάνο επικοινωνίας) για την</p>	<p>3. Release Plan: Προγραμματισμός επαναλήψεων κατά τις οποίες παραδίδονται τμήματα του προϊόντος.</p> <p>4. Sprint Plan: Προσδιορίζονται όλες τις εργασίες που πρέπει να γίνουν και προβλέπεται ο χρόνος για κάθε εργασία (εμπλέκεται ο Product Owner και η ομάδα ανάπτυξης και ενημερώνεται καθημερινά</p>

	<p>ανανέωση του Project Plan με πραγματικές ώρες εργασίας και αναθεωρεί το έργο. Καταγράφονται κίνδυνοι και προβλήματα.</p>	<p>από την ομάδα στο Daily Scrum meeting).</p> <p>5. Daily Scrum Meeting: συνάντηση που διαρκεί 15 λεπτά και την συντονίζει ο Scrum Master. Είναι το κύριο μέσο για την εσωτερική επικοινωνία της ομάδας. Ο Scrum Master ρωτάει το κάθε μέλος της ομάδας τι έγινε χθες, τι πρέπει να γίνει σήμερα και τυχόν προβλήματα. Ενημερώνεται το Sprint.</p>
	<p>6. Line Management: Ο PM διαπραγματεύεται με τη διοίκηση της εταιρίας σε κάθε στιγμή για τους πόρους και μπορεί να τους αλλάξει.</p> <p>7. Το έργο οδηγείται από την ημερομηνία παράδοσης και περιλαμβάνει όσο το δυνατόν περισσότερη τεκμηρίωση.</p>	<p>6. Line Management: Η ομάδα του Scrum δεν αλλάζει κατά τη διάρκεια ενός Sprint.</p> <p>7. Ο προγραμματισμός του έργου εξάγεται από το Release Plan. Το κάθε Sprint έχει συγκεκριμένη διάρκεια και οι εργασίες με την υψηλότερη προτεραιότητα πρέπει να γίνονται στα αρχικά Sprint.</p>
<p><i>Παρακολούθηση και έλεγχος των αλλαγών</i></p>	<p>8. Οι αλλαγές στις απαιτήσεις χειρίζονται μέσω μιας διαδικασίας αλλαγών που επιβλέπει ο PM.</p> <p>9. Ο PM είναι υπεύθυνος για την ανάλυση του ρίσκου και συνήθως διατηρεί ένα αντίστοιχο έγγραφο (Risk Document).</p>	<p>8. Μόνο η ομάδα Scrum μπορεί να αλλάξει τις λειτουργίες και τις εργασίες σε ένα Sprint και μόνο αν συμφωνούν όλα τα μέλη της. Αλλιώς η επιθυμητή αλλαγή προστίθεται στο Product Backlog σαν καινούρια εργασία.</p>

		9. Η ομάδα διαχειρίζεται την ανάλυση του ρίσκου του έργου πριν ξεκινήσει να το αναπτύσσει.
<i>Χρονοπρογραμματισμός και ανθρώπινοι πόροι</i>	<p>10. Ο PM δημιουργεί ένα Project Plan με εργασίες, αναθέσεις εργασιών και εκτιμήσεις για το χρόνο διεκπεραίωσής τους. Το Project Plan αναθεωρείται από την ομάδα έργου. Για τις αποκλίσεις από το Project Plan είναι υπεύθυνος ο PM.</p> <p>11. Ο PM ενημερώνει περιοδικά το Project Plan με τις πραγματικές ώρες και νέες εκτιμήσεις.</p> <p>12. Ο PM λειτουργεί ως αρχηγός για την ομάδα έργου και βοηθάει την ομάδα έργου να ξεπερνάει τα εμπόδια.</p>	<p>10. Τα μέλη της ομάδας δημιουργούν τις εργασίες και προβλέπουν το χρόνο διεκπεραίωσής τους.</p> <p>11. Τα μέλη της ομάδας ενημερώνουν τις εκτιμήσεις των εργασιών και προσθέτουν, τροποποιούν, διαγράφουν άλλες.</p> <p>12. Ο Scrum Master είναι υπεύθυνος για την ομάδα, αλλά τα μέλη της ομάδας λειτουργούν μόνα τους.</p>
<i>Ανέφικτη ημερομηνία ολοκλήρωσής του έργου</i>	<p>13. Αν η ημερομηνία ολοκλήρωσής του έργου είναι ανέφικτη, τότε ο PM διαπραγματεύεται τα εξής:</p> <ul style="list-style-type: none"> • Το σκοπό του έργου • Το χρονικό ορίζοντα ολοκλήρωσης • Το κόστος 	<p>13. Αν κάποιο παραδοτέο φαίνεται ότι θα καθυστερήσει, τότε οι εργασίες προστίθενται στο Product Backlog και μετατίθενται στο επόμενο Sprint. Δε μπορεί το Sprint να καθυστερήσει και ο Scrum Master δε μπορεί να</p>

	<ul style="list-style-type: none"> • Τους πόρους • Την ποιότητα 	<p>διαπραγματευτεί άλλους πόρους ή χρόνο.</p>
<p><i>Κόστος έργου</i></p>	<p>14. Ο PM συνήθως διαχειρίζεται το budget του έργου, που περιλαμβάνει κόστη για ανθρώπινους πόρους και άλλα έξοδα. Πιθανές υπερβάσεις του κόστους πρέπει να τεκμηριωθούν και να επιβεβαιωθούν από τη διοίκηση.</p>	<p>14. Στην αρχή του κάθε Sprint, ανατίθεται ένα συγκεκριμένο κόστος στο Sprint. Στη συνέχεια τα μέλη της ομάδας μπορούν να χρησιμοποιήσουν το budget αυτό όπως κρίνουν.</p>
<p><i>Έλεγχος ποιότητας έργου</i></p>	<p>15. Από την πλευρά του πελάτη, η ποιότητα ορίζεται ως η έγκαιρη παράδοση αυτών που είχαν συμφωνηθεί με το διαθέσιμο budget.</p> <p>16. Το προσωπικό που ελέγχει τον κώδικα ελέγχει ολόκληρο τον κώδικα στο τέλος του έργου.</p>	<p>15. Στο Scrum, ποιότητα σημαίνει δημιουργία επαυξημένων προϊόντων μετά από κάθε Sprint που να συμφωνούν με τις λειτουργίες που αναμένονταν.</p> <p>16. Το προσωπικό που ελέγχει τον κώδικα ελέγχει κάθε Sprint ξεχωριστά.</p>

Πίνακας 1. Σύγκριση μεθοδολογίας "Καταρράκτη" με Scrum αναφορικά με τις διαδικασίες

Κριτήριο	Μοντέλο Καταρράκτη	Scrum
Έγγραφα		
<i>Εξέταση του κόστους</i>	17. Cost Benefit Analysis: το έγγραφο κατατίθεται από τον PM και εγκρίνεται από τον πελάτη.	17. Δεν υπάρχει αντίστοιχο έγγραφο.
<i>Απαιτήσεις</i>	18. Τα έγγραφα που παράγονται από τις απαιτήσεις είναι τα εξής: <ul style="list-style-type: none"> • Feasibility Study • Statement of Work • Project Scope • Requirements Document • Functional Design • Detailed Design • Test Plans και Test Cases 19. Όλα τα έγγραφα εγκρίνονται από τον PM.	18. Τα έγγραφα που παράγονται από τις απαιτήσεις είναι τα εξής: <ul style="list-style-type: none"> • Product Backlog • Release Plan • Sprint Backlog/plan 19. Όλα τα έγγραφα αναθεωρούνται από Product Owner και την ομάδα ανάπτυξης.
<i>Προγραμματισμός επικοινωνίας</i>	20. Ο PM δημιουργεί ένα πρόγραμμα επικοινωνίας (Communication Plan) και καθορίζει τις συναντήσεις του έργου, τα παραδοτέα και ορίζει ποιοι θα συμμετέχουν σε κάθε συνάντηση.	20. Δεν υπάρχει αντίστοιχο Communication Plan στο Scrum αφού η ομάδα αυτό-οργανώνεται.

<i>Διαχείριση προϋπολογισμού</i>	21. Ο PM είναι υπεύθυνος για την επίβλεψη και αναθεώρηση του προϋπολογισμού του έργου.	21. Δεν υπάρχει αντίστοιχη διαδικασία. Εναπόκειται στην ομάδα η διαχείριση του προϋπολογισμού.
<i>Διαχείριση των κινδύνων</i>	22. Ο PM είναι υπεύθυνος για την επίβλεψη των κινδύνων που προκύπτουν και οφείλει να συμβουλεύει την ομάδα ώστε να ξεπερνάει εμπόδια και δυσκολίες, ώστε το προϊόν να παραδοθεί με επιτυχία.	22. Οι κίνδυνοι και τα προβλήματα που προκύπτουν διαχειρίζονται από τον Product Owner και την ομάδα ανάπτυξης κατά το Release Plan. Η ομάδα είναι υπεύθυνη για την επίλυση των προβλημάτων.
<i>Προγραμματισμός του έργου</i>	23. Ο PM δημιουργεί το συγκεκριμένο έγγραφο (Project Plan) στην αρχή του έργου με εκτιμήσεις για τις εργασίες του έργου. Στη συνέχεια, το χρονοδιάγραμμα αυτό ενημερώνεται τακτικά με τις πραγματικές ώρες και τις αναθεωρημένες εκτιμήσεις/καθήκοντα.	23. Ο προγραμματισμός κάθε επανάληψης γίνεται κατά το Sprint Planning Meeting από τον Product Owner και την ομάδα ανάπτυξης και ενημερώνεται καθημερινά.
<i>Αναφορές προόδου</i>	24. Ο PM δημιουργεί σε τακτά χρονικά διαστήματα αναφορές για την πρόοδο του έργου.	24. Δεν υπάρχει αντίστοιχο έγγραφο. Το Scrum επικεντρώνεται στην προσωπική επικοινωνία και έτσι η ομάδα είναι συνεχώς ενήμερη για την πρόοδο της ανάπτυξης.

Πίνακας 2. Σύγκριση μεθοδολογίας "Καταρράκτη" με Scrum αναφορικά με τα έγγραφα

Κριτήριο	Μοντέλο Καταρράκτη	Scrum
Συναντήσεις ομάδας		
<i>Συναντήσεις/Συντονισμός ομάδας</i>	25. Σύμφωνα με το Communication Plan, ο PM συνήθως προγραμματίζει συναντήσεις για την επίβλεψη του έργου. Οι συναντήσεις αυτές μπορεί να διαρκέσουν μία ώρα ή και περισσότερο.	25. Στο Scrum υπάρχει μια καθημερινή 15-λεπτη συνάντηση για τη συζήτηση της πορείας του έργου. Στο τέλος του κάθε Sprint, ένα demo πραγματοποιείται κατά το οποίο παρουσιάζεται στον πελάτη του προϊόντος ό,τι καινούριο δημιουργήθηκε κατά τη διάρκεια του Sprint.
<i>Αναθεωρήσεις μετά την υλοποίηση του έργου</i>	26. Ο PM πραγματοποιεί μια αναθεώρηση μετά την ολοκλήρωση του έργου, προκειμένου να καταγράψει ότι πήγε καλά και αυτά που θα μπορούσαν να βελτιωθούν.	26. Κατά το Sprint Retrospective, το οποίο πραγματοποιείται μετά το τέλος κάθε Sprint, τα μέλη της ομάδας αναθεωρούν το παρελθόν, διατυπώνουν συστάσεις για μελλοντικές βελτιώσεις ή αλλαγές.

Πίνακας 3. Σύγκριση μεθοδολογίας "Καταρράκτη" με Scrum αναφορικά με τις συναντήσεις των ομάδων

Από τα παραπάνω, συμπερασματικά, μπορεί να εξαχθούν τα πλεονεκτήματα καθώς και τα μειονεκτήματα της μεθοδολογίας Scrum έναντι των παραδοσιακών μεθοδολογιών και πιο συγκεκριμένα με την καταρακτοειδή.

7.1. Πλεονεκτήματα της Scrum συγκριτικά με την «Καταρακτοειδή» Μεθοδολογία.

- Ευκολία αλλαγών αντί προσκόλλησης στο πλάνο: Είναι εξαιρετικά σπάνιο, έως αδύνατο, στην παραγωγή λογισμικού να μην προκύψουν αλλαγές κατά τη διάρκεια του κύκλου ζωής. Επιπροσθέτως, συνήθως, ο πελάτης ενδιαφέρεται απλώς το λογισμικό να ικανοποιεί τις ανάγκες του, αλλά δεν ενδιαφέρεται για το πώς θα γίνει αυτό. Η μεθοδολογία Scrum

είναι σχεδιασμένη, ώστε να παρέχει ευελιξία σε όλη τη διάρκεια του κύκλου ζωής ανάπτυξης ενός πληροφοριακού συστήματος και να παρέχει μια συνεχή, ταχεία αναδιοργάνωση, επιτρέποντας ξαφνικές μαζικές αλλαγές στο έργο, χωρίς μεγάλες ζημιές από άποψη χρόνου και κόστους. Τούτο επιτυγχάνεται με τον διαχωρισμό του έργου σε μικρότερα, όπως αναλύθηκε και σε προηγούμενη παράγραφο. Αντίθετα, στις παραδοσιακές προσεγγίσεις και συγκεκριμένα στη καταρρακτοειδή μεθοδολογία, ακολουθείται το πλάνο που έχει ορισθεί εξ' αρχής και οι αλλαγές που μπορεί να προκύψουν, ίσως αποδειχθούν καταστροφικές.

- Ανθρωποκεντρικός προσανατολισμός αντί προσανατολισμού στις διαδικασίες: Στη Scrum, όλοι είναι ενήμεροι για το τι πραγματώνεται στην ομάδα, αφού ο καθένας ξέρει τι κάνει ο κάθε άλλος και με τη συνεργασία μεταξύ των ομάδων και των μελών επιτυγχάνεται η ευκολότερη ολοκλήρωση των εργασιών. Στη μεθοδολογία του «Καταρράκτη», αντίθετα, ο προσανατολισμός είναι προς τις διαδικασίες με αποτέλεσμα την ανεξάρτητη εργασία που κάνει δύσκολη και πολύπλοκη τη διασύνδεση όλων των εμπλεκόμενων μερών.
- Κώδικας αντί τεκμηρίωσης: Η συγγραφή κώδικα αντί των παραδοσιακών εγγράφων τεκμηρίωσης αποτελεί, ίσως, το σημαντικότερο πλεονέκτημα της μεθοδολογίας Scrum. Η Scrum βασίζεται σε επαναλήψεις, αποτέλεσμα των οποίων είναι ένα μικρό τμήμα του λογισμικού κάθε φορά. Η τεκμηρίωση κάθε επανάληψης είναι ο ίδιος ο κώδικας που παράγεται. Αντιθέτως, στην καταρρακτοειδή μεθοδολογία πρέπει πρώτα να υπάρχει πλήρης τεκμηρίωση (έγγραφο απαιτήσεων κτλ.) και στη συνέχεια συγγράφεται ο κώδικας που στην ουσία πραγματώνει το σύστημα.
- Διαπροσωπικές σχέσεις αντί τυπικών διαδικασιών με τον πελάτη: Λόγω του μικρού χρόνου των επαναλήψεων (Sprints) ο πελάτης έχει γνώση για την εξέλιξη του έργου, καθώς και συχνή ανατροφοδότηση σχετικά με τις λειτουργίες του προϊόντος, ενώ μπορεί να βλέπει νέες εκδόσεις του λογισμικού σε τακτά χρονικά διαστήματα. Έτσι, αναπτύσσεται μια σχέση εμπιστοσύνης η οποία μπορεί να βελτιώσει τις διαδικασίες ανάπτυξης. Δεν υπάρχει κάτι αντίστοιχο στο καταρρακτοειδές μοντέλο.

- Ποιοτικότερα πληροφοριακά συστήματα: Ακολουθώντας τη μεθοδολογία Scrum η ανάπτυξη επικεντρώνεται στα υψηλότερης προτεραιότητας και, συνεπώς, επιχειρηματικής αξίας χαρακτηριστικά σε ένα σύστημα. Με τον τρόπο αυτό αυξάνεται η παραγωγικότητα και μακροπρόθεσμα η ποιότητα του παραγόμενου προϊόντος. Ως επακόλουθο, επιφέρεται μείωση του συνολικού κόστους πραγμάτωσης (ανάπτυξης και συντήρησης). Κάθε επανάληψη (Sprint) οδηγεί στην παραγωγή ποιοτικότερου προϊόντος, μιας και πριν από κάθε Sprint, εξασφαλίζεται ένα πλήρως λειτουργικό τμήμα του συνολικού συστήματος. Επιπλέον, εξετάζοντας κάθε Sprint, κατά τη διάρκεια του Sprint Retrospective Meeting, πριν προχωρήσουμε στο επόμενο, σημαίνει ότι οι διαδικασίες που διεξάγονται σε όλη τη διάρκεια της ανάπτυξης βελτιώνονται συνεχώς μέσα από εμπειρικές διαδικασίες, με αποτέλεσμα και τη βελτίωση του παραγόμενου αποτελέσματος.
- Ικανοποίηση των πελατών: Όπως αναφέρθηκε ήδη, η ενεργή συμμετοχή των πελατών στη διαδικασία ανάπτυξης, είναι ένα από τα σημαντικά χαρακτηριστικά της μεθοδολογίας Scrum. Με αυτό επιτυγχάνεται η μεγιστοποίηση της ικανοποίησης των πελατών, μιας και είναι εκείνος (ο πελάτης) ο οποίος ορίζει τη λίστα προτεραιοτήτων στο ανεκτέλεστο υπόλοιπο του προϊόντος. Επίσης, είναι συνεχώς ενήμερος για τη διαδικασία ολοκλήρωσης του συστήματος από τα τμήματα λογισμικού που παρουσιάζονται μετά από κάθε επανάληψη. Είναι η διαφάνεια που κάνει τη διαφορά και εξασφαλίζεται τόσο η παραγωγικότητα όσο και η ικανοποίηση των πελατών.

7.2. Μειονεκτήματα της Scrum συγκριτικά με την «Καταρρακτοειδή» Μεθοδολογία.

- Η συνεργασία με τον πελάτη είναι ζωτικής σημασίας: Στο καταρρακτοειδές μοντέλο η συμμετοχή των πελατών περιορίζεται στην αρχική καταγραφή των απαιτήσεων και την ανατροφοδότηση προς το τέλος, με περιορισμένες συναντήσεις μεταξύ αυτών και της ομάδας ανάπτυξης. Αντίθετα, η συνεργασία με τον πελάτη είναι ένας σημαντικός παράγοντας επιτυχίας για τη Scrum, αφού όπως είδαμε, εμπλέκεται

ενεργά σε όλη τη διαδικασία ανάπτυξης του συστήματος (καθορισμό των user stories, συζητώντας τα χαρακτηριστικά του προϊόντος, την ιεράρχηση του Product Backlog, και την παροχή ανατροφοδότησης στην ομάδα ανάπτυξης σε τακτική βάση). Ο πελάτης είναι, επίσης, το μέσο για να διασφαλιστεί η πρόοδος του συστήματος και η επικύρωση των παραγόμενων τμημάτων σε κάθε επανάληψη. Αποτέλεσμα των παραπάνω είναι ότι, σε περίπτωση που υπάρχει έλλειψη συμμετοχής του πελάτη, υπάρχουν, αυτομάτως, προβλήματα στη συλλογή και την αποσαφήνιση των απαιτήσεων, καθώς και μείωση της παραγωγικότητας και της ποιότητας του προϊόντος.

- Η έλλειψη τελικών αποφάσεων από τα ανώτερα στελέχη του οργανισμού: Αν και στη Scrum η ομάδα αυτό-διοικείται και μπορεί να πάρει αποφάσεις (κατευθύνεται μόνη της και πάντα με τη βοήθεια του Scrum Master), τα ανώτερα στελέχη του οργανισμού πρέπει να είναι πρόθυμα ώστε να προβούν σε τροποποιήσεις και να λαμβάνουν σαφείς, τελικές αποφάσεις προκειμένου η ομάδα ανάπτυξης να ξεπερνά εμπόδια και δυσκολίες.
- Μη γνώση της μεθοδολογίας από το προσωπικό: Η Scrum είναι μία μεθοδολογία η οποία χρειάζεται έμπειρα μέλη με εξειδίκευση και αφοσίωση στο έργο ανάπτυξης. Η ομάδα αυτό-διοικείται και τα μέλη της πρέπει να έχουν εμπειρία στην παροχή βασικών λειτουργιών μέσα σε σύντομο χρονικό διάστημα.
- Όχι ιδανική για πολύ μεγάλα ή πολύπλοκα έργα: Αυτός είναι και ένας από τους γενικότερους περιορισμούς των ευέλικτων μεθοδολογιών. Έτσι και στη Scrum, το ότι βασίζεται στη διαφάνεια και την επικοινωνία, την καθιστά δύσκολη για μεγάλα έργα, καθώς αυτά συνεπάγονται και πολλούς συμμετέχοντες. Όσο αυξάνεται, όμως, ο αριθμός των μελών μιας ομάδας τόσο μειώνεται η δυνατότητα για επαρκή επικοινωνία και συντονισμό ανάμεσα στα μέλη της.

8. Μελέτη Περίπτωσης

8.1. Εισαγωγή

Στόχος του παρόντος κεφαλαίου είναι η κριτική αξιολόγηση και η παράθεση των ευρημάτων της μεθοδολογίας Scrum σε πραγματικές συνθήκες. Για τον σκοπό αυτόν, επιχειρείται η ακριβής εφαρμογή της μεθοδολογίας, όπως αυτές καταγράφηκαν στο θεωρητικό μοντέλο της παρούσας εργασίας καθώς και οι δυσχέρειες και αποκλίσεις που παρατηρήθηκαν κατά την εφαρμογή της.

8.2. Ερευνητική Μεθοδολογία

Στη μελέτη περίπτωσης της παρούσας διπλωματικής εργασίας επιλέχθηκε η ποιοτική ερευνητική μεθοδολογία. Σύμφωνα με έναν εκ των ορισμών της «η ποιοτική έρευνα είναι μια πλαισιοθετημένη δραστηριότητα (situated activity), η οποία τοποθετεί τον παρατηρητή στον κόσμο. Αυτή συνίσταται σε ένα σύνολο ερμηνευτικών και υλικών πρακτικών, οι οποίες κάνουν τον κόσμο ορατό. Αυτές οι πρακτικές μετασχηματίζουν τον κόσμο. Μετατρέπουν τον κόσμο σε μια σειρά από αναπαραστάσεις του εαυτού, συμπεριλαμβανομένων των σημειώσεων πεδίου, των συνεντεύξεων, των συνομιλιών, των φωτογραφιών, των μαγνητοφωνήσεων και των σημειώσεων σε ημερολόγια. Σε αυτό το επίπεδο, η ποιοτική έρευνα περιλαμβάνει μια ερμηνευτική, νατουραλιστική προσέγγιση στον κόσμο. Αυτό σημαίνει ότι οι ποιοτικοί ερευνητές μελετούν τα πράγματα στο φυσικό τους πλαίσιο, επιχειρώντας να δώσουν νόημα ή να ερμηνεύσουν τα φαινόμενα με όρους των νοημάτων που οι άνθρωποι δίνουν σε αυτά» (Lincoln & Denzin, 2005).

Για να θεωρηθεί μία ποιοτική έρευνα σωστή θα πρέπει να ακολουθεί τα παρακάτω συνήθη στάδια, όπως προτάθηκε από τους Cohen, Manion & Morrison (2011).

- Προσδιορισμός του θέματος/ομάδας/φαινομένου της έρευνας.
- Βιβλιογραφική ανασκόπηση .
- Σχεδιασμός του ερευνητικού ερωτήματος, της έρευνας και της συλλογής δεδομένων .

- Εντοπισμός των τομέων που αφορούν την έρευνα και τον ρόλο του ερευνητή στην έρευνα/κατάσταση.
- Εντοπισμός των πηγών πληροφόρησης (π.χ. ατόμων/ομάδων που θα συμμετέχουν στην έρευνα).
- Ανάπτυξη σχέσεων συνεργασίας με τους συμμετέχοντες.
- Διεξαγωγή της έρευνας και της συλλογής δεδομένων ταυτόχρονα.
- Διεξαγωγή της ανάλυσης των δεδομένων, είτε ταυτόχρονα, σε συνεχή βάση, δεδομένου ότι η κατάσταση προκύπτει και εξελίσσεται, ή διεξαγωγή της ανάλυσης των δεδομένων μετά την έρευνα.
- Έκθεση αποτελεσμάτων και της εμπειρικά θεμελιωμένης θεωρίας (grounded theory) ή των απαντήσεων στις ερωτήσεις που προκύπτουν από την έρευνα.
- Δημιουργία μιας υπόθεσης για περαιτέρω έρευνα ή δοκιμή.

Στην παρούσα μελέτη περίπτωσης ακολουθήθηκε η παραπάνω προσέγγιση με τα στάδιά της, καθώς πραγματοποιήθηκαν επισκέψεις σε τόπους κατάταξης, στις αρμόδιες διευθύνσεις του στρατού, παρατηρήθηκαν αδυναμίες και δυσκολίες, συζητήθηκαν διεξοδικά θέματα που την αφορούσαν, αναπτύχθηκαν σχέσεις συνεργασίας με τους συμμετέχοντες κτλ.

8.3. Στοιχεία του Έργου

Το έργο που πραγματεύεται η παρούσα μελέτη περίπτωσης, αφορά το Πληροφοριακό Σύστημα Κατάταξης Νεοσυλλέκτων στον Στρατό Ξηράς. Το έργο αυτό αφορά στην διαδικασία κατάταξης των νεοσυλλέκτων στρατιωτών στα κέντρα κατάταξης τους. Δεν κρίνεται σκόπιμο να αναφερθεί στην παρούσα η διαδικασία κατάταξης, ωστόσο, σημειώνεται ότι η διαδικασία έως τώρα γινόταν με εφαρμογή σε MS Access με τους όποιους σοβαρούς περιορισμούς αυτό συνεπάγεται. Επιπροσθέτως, ένας επιπλέον σημαντικός περιορισμός είναι ότι η εφαρμογή «έτρεχε» σε κατά τόπους διακομιστές χωρίς σύνδεση μεταξύ τους, ενώ η μεταφορά δεδομένων μεταξύ αυτών, καθώς και του Κέντρου Πληροφοριακής Υποστήριξης του Ελληνικού Στρατού (ΚΕΠΥΕΣ), γινόταν εκτός εφαρμογής με άλλα μέσα.

Για την ορθότερη κατανόηση της προβληματικής κατάστασης πραγματοποιήθηκε η Μεθοδολογία Μαλακών Συστημάτων (παρατίθεται στο Παράρτημα Α).

Η πελάτης του πληροφοριακού συστήματος είναι η αρμόδια για την κατάταξη των οπλιτών θητείας διεύθυνση και το προϊόν είναι ένα νέο πληροφοριακό σύστημα κατάταξης νεοσυλλέκτων.

Αρχικά δίδεται ένα περίγραμμα της ανάπτυξης του πληροφοριακού συστήματος με τη μεθοδολογία του «Καταρράκτη» (παρατίθεται στο Παράρτημα Β), με την οποία εκτελούνται τα περισσότερα έργα στον υπόψη οργανισμό. Ωστόσο, αποφασίστηκε, να εισαχθεί η μεθοδολογία Scrum και το πληροφοριακό σύστημα να αναπτυχθεί τελικώς με βάση τη μεθοδολογία αυτή.

Στις παραγράφους που ακολουθούν καταγράφεται αναλυτικά η διαδικασία της μεθοδολογίας Scrum με τα επί μέρους βήματά της, για την πραγμάτωση του νέου πληροφοριακού συστήματος. Τα πρόσωπα της ομάδας του έργου είναι φανταστικά, ωστόσο οι απαιτήσεις ανάπτυξης του έργου είναι πραγματικές.

8.4. Ανάθεση Ρόλων

Είναι σημαντικό πριν την έναρξη του έργου, ο οργανισμός να έχει κατανοήσει τις βασικές αρχές και τις πρακτικές των ευέλικτων μεθοδολογιών. Τούτο, θα επιτευχθεί μετά από μία σειρά σχετικών παρουσιάσεων.

Εντός του οργανισμού για την εφαρμογή της μεθοδολογίας και την πραγμάτωση του πληροφοριακού συστήματος επιλέγονται οι κάτωθι αναλόγως των δεξιοτήτων τους και της ειδικότητας τους.

- Scrum Master: Για τον ρόλο επιλέγεται ο Σπύρος, καθώς είναι ο πλέον έμπειρος της διαδικασίας.
- Product Owner: Για τον ρόλο επιλέγεται ο Βαγγέλης, ο οποίος εξειδικεύεται στην καταγραφή και ανάλυση των απαιτήσεων των πελατών.
- Στην Ομάδα Ανάπτυξης εντάσσονται οι προγραμματιστές Γιάννης, Κώστας, Γιώργος και Ελένη.

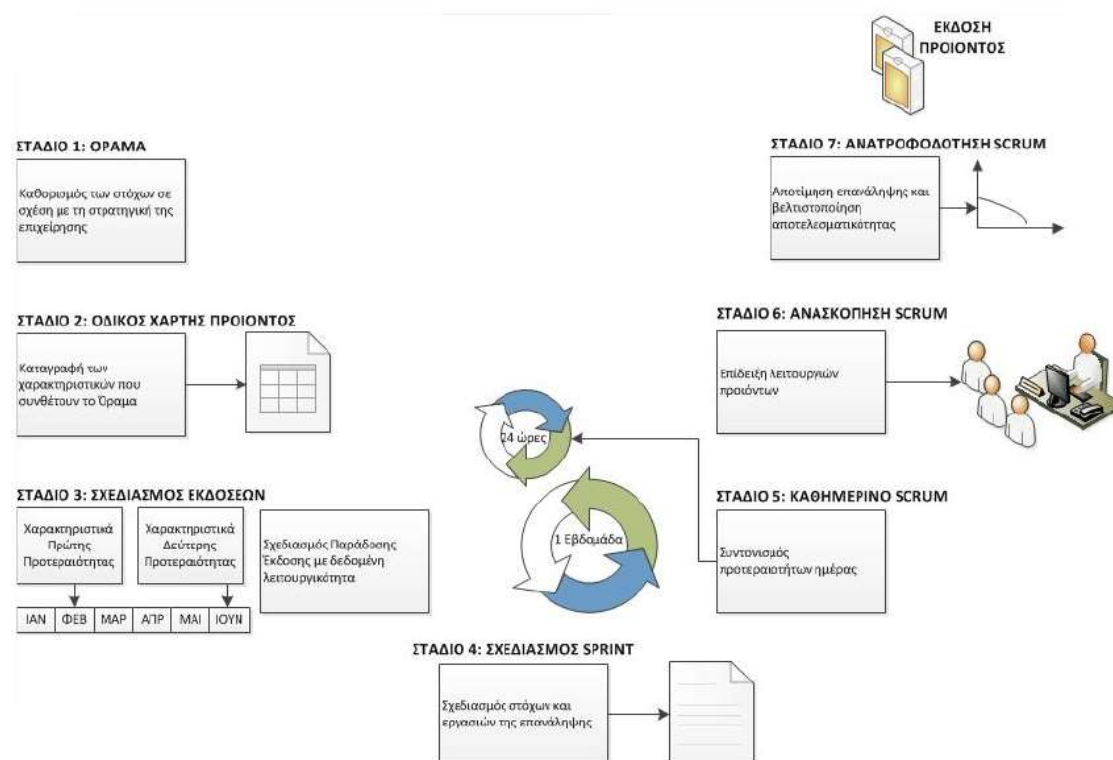
8.5. Παρακολούθηση Μεθοδολογίας

Για την εφαρμογή και παρακολούθηση της μεθοδολογίας Scrum υπάρχουν πολλοί τρόποι και εφαρμογές, όπως trello.com, app.pipify.com, app.scrumdo.com, icescrum.com, targetprocess.com, tfs.visualstudio.com κ.α. Η χρήση παραδοσιακών μέσων, όπως οι πίνακες σημειώσεων, αποτελούν εξίσου σημαντικά εργαλεία για τον σχεδιασμό και προγραμματισμό του έργου.

Στην παρούσα μελέτη περίπτωσης θα χρησιμοποιηθεί η παραδοσιακή μέθοδος με πίνακες, sticker notes κτλ.

8.6. Εφαρμογή Μεθοδολογίας Scrum

Η εφαρμογή της μεθοδολογίας σε όλη τη διαδικασία ανάπτυξης του πληροφοριακού συστήματος ακολουθεί τα βήματα όπως ακριβώς περιγράφηκαν στο κεφάλαιο 6 της μεθοδολογίας Scrum και φαίνονται παρακάτω στην εικόνα.



Εικόνα 21. Βήματα - Στάδια της Scrum (Layton, 2012)

8.6.1. Βήμα 1 – Όραμα

Η δήλωση του οράματος συντάσσεται από τον Ιδιοκτήτη του Προϊόντος και εγκρίνεται, εκτός αν υπάρχουν τροποποιήσεις από τον πελάτη.

Όραμα (Vision) για το έργο «Πληροφοριακό Σύστημα Κατάταξης Νεοσυλλέκτων Στρατού Ξηράς»	
<i>Για</i>	ΥΠΕΘΑ/ΓΕΣ – Αρμόδια Διεύθυνση
<i>Ο οποίος</i>	<p>Επιθυμεί</p> <ul style="list-style-type: none"> • την ανάπτυξη νέου διαδικτυακού πληροφοριακού συστήματος κατάταξης • την βελτίωση της διαδικασίας κατάταξης • την εναρμόνιση της κατάταξης σύμφωνα με τους νέους νόμους που την αφορούν.
<i>Η</i>	Ολοκληρωμένη διαδικτυακή πύλη που θα αναπτυχθεί.
<i>Η οποία είναι</i>	Ένα σύνθετο πληροφοριακό σύστημα παροχής ηλεκτρονικών υπηρεσιών κατάταξης στρατιωτών.
<i>Το οποίο</i>	<p>Επιτρέπει</p> <ul style="list-style-type: none"> • την επίσπευση της διαδικασίας κατάταξης • την ορθότερη λειτουργία της διαδικασίας κατάταξης • τη λειτουργία της διαδικασίας κατάταξης σε απομακρυσμένες μονάδες.

<i>Σε αντίθεση με</i>	Την υπάρχουσα «αρχαία» εφαρμογή σε MS Access.
<i>Το προϊόν μας</i>	Βελτιώνει τη διαδικασία κατάταξης και είναι διαθέσιμο διαδικτυακά σε ολόκληρη την ελληνική επικράτεια πλήρως εναρμονισμένο με τους νόμους που διέπουν την κατάταξη.

8.6.2. Βήμα 2 – Product Backlog

Το Backlog του πληροφοριακού συστήματος αρχικά περιλαμβάνει απαιτήσεις πολύ υψηλού επιπέδου και συνεπώς χαμηλής ανάλυσης. Για παράδειγμα:

- Όμορφο εικαστικό βασισμένο στα σύγχρονα πρότυπα για το πληροφοριακό σύστημα.
- Περιοχή αναζήτησης και ανάκτησης νεοσυλλέκτου.
- Κατασκευή σελίδων για όλες τις κατηγορίες στοιχείων των νεοσυλλέκτων (Προσωπικά, Οικογενειακά, Στοιχεία Ταυτότητας, Υγεία, Κατάταξη, Γνώσεις, Μοριοδότηση)
- Δυνατότητα καταχώρησης νεοσυλλέκτου με το δυνατόν λιγότερα κλικ.
- Απαλοιφή από τη διαδικασία και την εφαρμογή διαδικασιών που δεν είναι πλέον έγκυρες.
- Δημιουργία νέας βάσης δεδομένων για σύνδεση με την παλιά.
- Υποστήριξη εκτυπωτικών διαδικασιών.
- Υποστήριξη διαχειριστικού για χρήστες στις μονάδες εκστρατείας.

Έπειτα από την παραπάνω καταγραφή η ομάδα εργασίας καλείται να εκτιμήσει την προσπάθεια (effort), την επιχειρηματική αξία (business value) και την προτεραιότητα κάθε απαίτησης από τις παραπάνω. Κατά τη διάρκεια ανάπτυξης και μέσα από τη διαδικασία της επαναλαμβανόμενης αποσύνθεσης το Product Backlog γίνεται πιο εξειδικευμένο.

8.6.3. Βήμα 3 – Προγραμματισμός εκδόσεων

Με βάση το Product Backlog που αναφέρθηκε προηγουμένως και την εκτίμηση του απαιτούμενου άνθρωπο-χρόνου το επόμενο βήμα στη διαδικασία είναι ο προγραμματισμός των εκδόσεων του πληροφοριακού συστήματος και των επαναλήψεων στα πλαίσια κάθε έκδοσης και θα περιλαμβάνει:

1. Έκδοση 1.0 του πληροφοριακού συστήματος στην οποία προγραμματίζεται η κατασκευή της βάσης δεδομένων, η δημιουργία διαδικασίας εισόδου καθώς και η κατασκευή των βασικών μενού. Προβλέπονται τρεις επαναλήψεις (Sprints), διάρκειας μίας εβδομάδας.
2. Έκδοση 2.0 του πληροφοριακού συστήματος στην οποία προγραμματίζεται η κατασκευή όλων των σελίδων - καρτελών με τις επί μέρους λειτουργίες τους σε δύο επαναλήψεις διάρκειας δύο εβδομάδων.
3. Έκδοση 3.0 του πληροφοριακού συστήματος στην οποία προγραμματίζεται η κατασκευή όλων των εκτυπωτικών λειτουργιών της εφαρμογής σε τρεις επαναλήψεις διάρκειας μίας εβδομάδας.
4. Έκδοση 4.0 του πληροφοριακού συστήματος στην οποία προγραμματίζεται η κατασκευή της υπό-εφαρμογής έκδοσης ταυτοτήτων καθώς και λοιπών υπό-εφαρμογών υγειονομικού ενδιαφέροντος.

8.6.4. Βήματα 4,5,6,7 Εκτέλεση των Sprints

8.6.4.1. Το πρώτο Sprint

Το πρώτο Sprint της έκδοσης 1.0 του πληροφοριακού συστήματος έχει ως στόχο την εικαστική ολοκλήρωση του, την κατασκευή των πρώτων ενοτήτων λογισμικού, καθώς και την παραμετροποίηση του. Η παρούσα έκδοση εκτιμήθηκε αρχικά ότι θα ολοκληρωθεί σε τρία συνολικά Sprints με στόχο να κατασκευαστεί πλήρως και η βάση δεδομένων.

Σχεδιασμός του Sprint (*Sprint Planning*)

Ο συντονιστής, που είναι ο Ιδιοκτήτης του Προϊόντος (Product Owner), σχεδιάζει το Sprint μαζί με την ομάδα έργου και προχωρά στην πρώτη αποσύνθεση του Product Backlog που παρουσιάστηκε παραπάνω. Όσες από τις απαιτήσεις αναμένεται να αναπτυχθούν στα πλαίσια του Sprint, αναλύονται

σε επί μέρους εργασίες, οι οποίες ανατίθενται στα μέλη της ομάδας ανάπτυξης. Ένα παράδειγμα αυτής της ανάθεσης εργασιών για το Sprint φαίνεται στον πίνακα παρακάτω:

Προσπάθεια/ Effort	Εργασία/Task	Κατάσταση/State	Ανάθεση σε/Assigned to	Απομένουσα Εργασία
16	Ανάπτυξη template CSS	New	Γιάννης	16
	Δημιουργία κατάλληλων εικόνων και εικονιδίων	In progress	Γιάννης	4
	Κωδικοποίηση σε CSS	In progress	Γιάννης	12
4	Παραμετροποίηση template	New	Κώστας	
	Παραμετροποίηση για σελίδες εισόδου, μενού και πίνακες	To do	Κώστας	
4	Δημιουργία αρχικών μενού	Committed	Γιώργος	4
	Δημιουργία των αρχικών μενού της εφαρμογής καθώς και της λειτουργίας του toolbar	In Progress	Γιώργος	4
16	Κατασκευή βάσης δεδομένων σε MySQL	Committed	Ελένη	16
	Ανάλυση παλαιότερης ΒΔ σύγκριση με Oracle DB	In Progress	Ελένη	6
	Δημιουργία νέας ΒΔ σε MySQL	In Progress	Ελένη	10

Εικόνα 22. Sprint Backlog και ανάθεση εργασιών (Task Allocation)

Καθημερινές συνεδριάσεις Scrum (Daily Scrum)

Κάθε ημέρα της εβδομάδας και για χρονικό διάστημα ίσο με 15 λεπτά η ομάδα εργασίας διενεργεί καθημερινή συνάντηση που ως στόχο έχει την παρακολούθηση της εξέλιξης των εργασιών, αλλά και τον εντοπισμό εμποδίων.

Η παρακολούθηση γίνεται εποπτικά μέσω πίνακα σημειώσεων στον οποίο υπάρχουν οι στήλες: Product Backlog Item - PBI(αντικείμενο εργασίας), Το Do, In Progress και Done. Τα sticker notes, στα οποία είναι γραμμένες οι εργασίες του Sprint, κολλώνται, αναλόγως της κατάστασής στην οποία βρίσκονται, στην κατάλληλη στήλη, ώστε όλοι να γνωρίζουν τι έχει γίνει και τι γίνεται αλλά και τι πρέπει να γίνει για να ολοκληρωθεί το Sprint. Κάθε μέρα τα sticker notes αναδιοργανώνονται με τελικό σκοπό να καταλήξουν, στο τέλος του Sprint, στη στήλη Done.

PBI	Todo	In Progress	Done
			
			
			
			

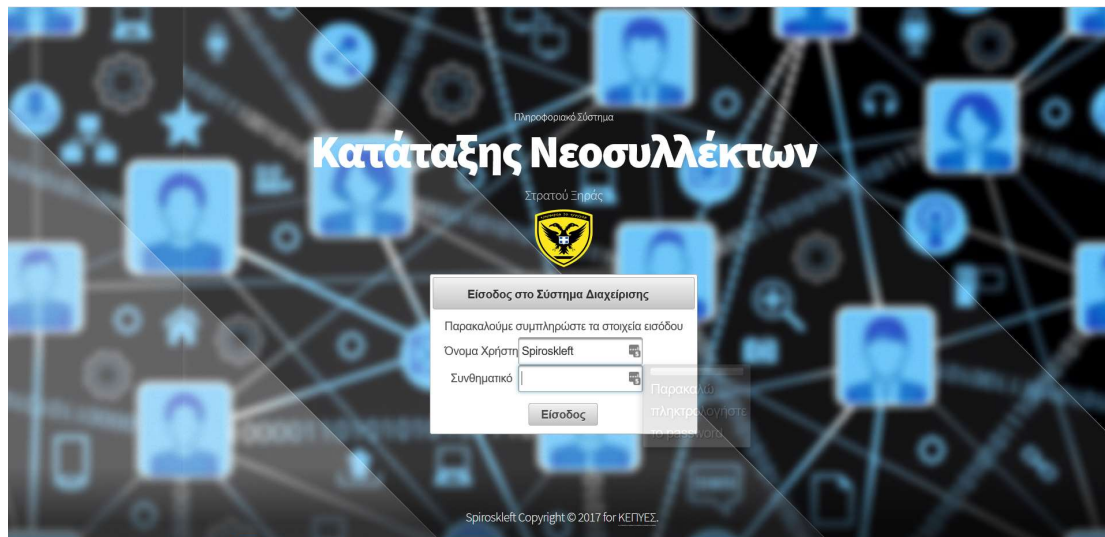
Εικόνα 23. Πίνακας εποπτείας Sprint

Η ομάδα έργου κατά τη διάρκεια των καθημερινών συνεδριάσεων Scrum αναφέρει ως εμπόδιο ότι ο πελάτης (η αρμόδια διεύθυνση του στρατού) δεν έχει αποδώσει ολοκληρωμένο τον χάρτη πλοήγησης της εφαρμογής (sitemap), με αποτέλεσμα να καθίσταται ανέφικτη η πλήρης παραμετροποίηση. Η ομάδα έργου αποφασίζει να ολοκληρώσει την ανάπτυξη και να ζητήσει feedback από τον πελάτη στο τέλος του Sprint (sprint review). Παρατηρήθηκε, επίσης, ότι στοιχεία του CSS δε λειτουργούν ορθά στον Microsoft Edge και αποφασίστηκε η προσθήκη στο backlog της απαίτησης ελέγχου σε όλους τους γνωστούς browsers.

Επισκόπηση του Sprint

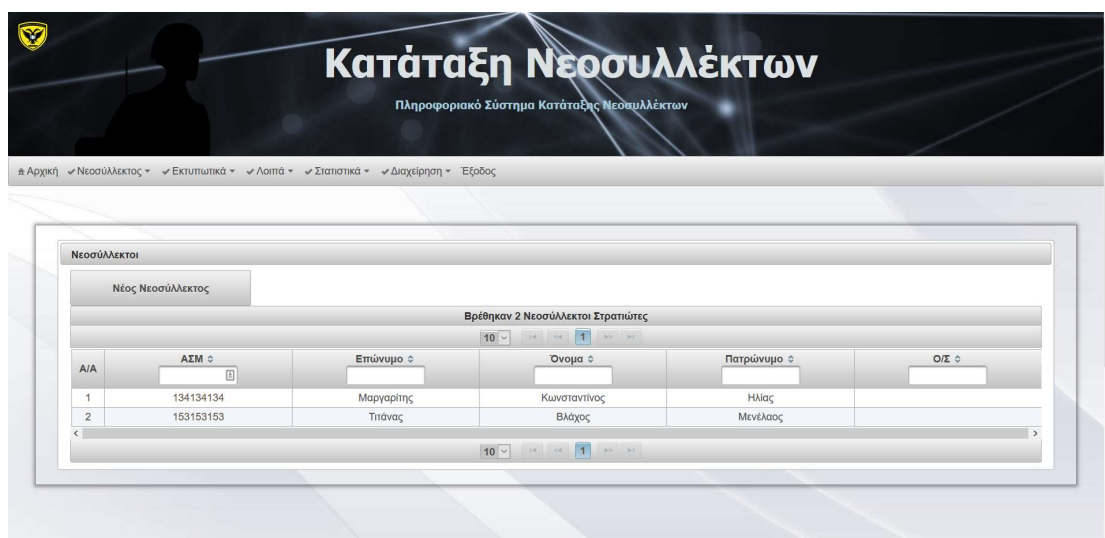
Την τελευταία μέρα του Sprint και για διάρκεια μίας ώρας λαμβάνει χώρα η επισκόπηση του Sprint. Εκεί παρουσιάζονται αναλυτικά όλες οι εργασίες που εκτελέστηκαν. Ορισμένα παραδείγματα εργασιών που παρουσιάστηκαν είναι:

1. Επισημαίνονται οι απαιτήσεις καθώς και οι βασικοί στόχοι που αναμένεται να αναπτυχθούν στα επόμενα Sprints.
2. Παρουσίαση της αρχικής σελίδας εισόδου (login). Παρουσιάζονται οι βασικές αρχές εικαστικής προσέγγισης καθώς και θέματα γραμματοσειρών, εικόνων και χρωματολογίου.



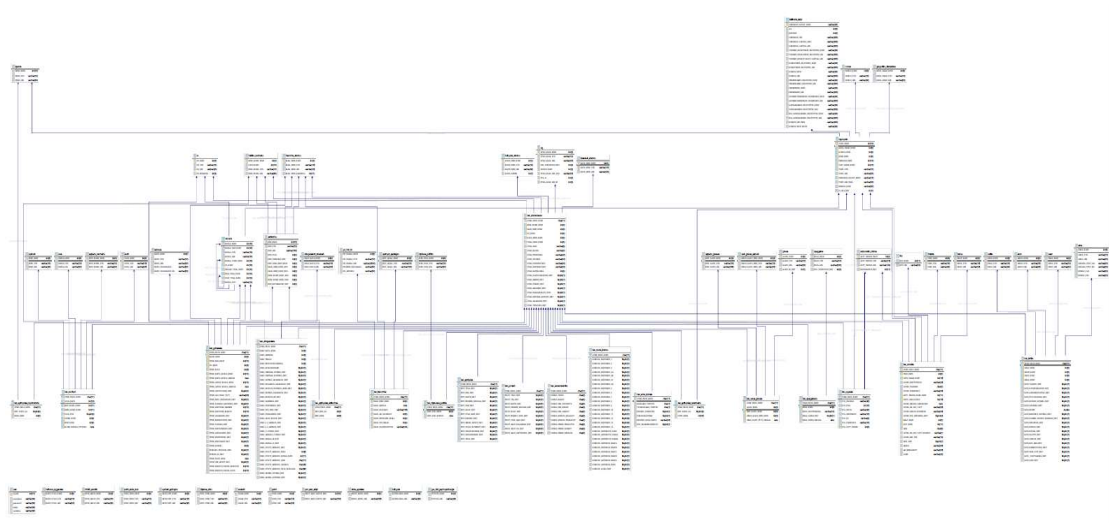
Εικόνα 24. Σελίδα εισόδου (login) του Πληροφοριακού Συστήματος

3. Παρουσίαση αρχικής σελίδας αναζήτησης νεοσυλλέκτου. Επίσης και εδώ παρουσιάζονται οι βασικές αρχές εικαστικής προσέγγισης καθώς και θέματα γραμματοσειρών, εικόνων και χρωματολογίου.



Εικόνα 25. Σελίδα αναζήτησης νεοσυλλέκτου

4. Παρουσίαση του σχήματος της βάσης δεδομένων.



Εικόνα 26. Σχήμα Βάσης Δεδομένων Πληροφοριακού Συστήματος

5. Πλοήγηση σε όλες τις διαθέσιμες σελίδες της εφαρμογής.

Τελειώνοντας την επισκόπηση του Sprint και αν δεν υπάρχουν νέες παρατηρήσεις, απαιτήσεις ή εμπόδια προς αξιολόγηση, ακολουθεί το επόμενο βήμα που είναι η αναδρομή στο Sprint.

Αναδρομή στο Sprint

Η Αναδρομή στο Sprint είναι μια ευκαιρία για την Ομάδα Scrum να επιθεωρήσει τον εαυτό της και να δημιουργήσει ένα πλάνο βελτιώσεων, το οποίο θα εκτελέσει κατά τη διάρκεια του επόμενου Sprint. Λαμβάνει χώρα την τελευταία μέρα του Sprint και διαρκεί περίπου 1,5 ώρα.

Στα πλαίσια της αναδρομής καταγράφονται τα ακόλουθα:

- Τι πήγε καλά στο Sprint;
 - Οι καθημερινές συνεδριάσεις Sprint γινόταν εντός χρόνου.
 - Καλές ιδέες για εικαστικές βελτιώσεις.
 - Καλή συνεργασία μεταξύ των ατόμων της ομάδας.
 - Ενθουσιασμός επί της νέας μεθοδολογίας.
- Τι πρέπει να διορθωθεί;
 - Η αναζήτηση νεοσυλλέκτου πρέπει να γίνει πιο αποδοτική.
 - Η ομάδα λόγω φόρτου εργασίας απασχολείται εμβόλιμα με θέματα υποστήριξης άλλων πελατών (διευθύνσεων του στρατού).

Αποτέλεσμα τούτου είναι μα μην επιτυγχάνεται συγκέντρωση στις απαιτήσεις του έργου. (distraction)

- Πώς θα επιτευχθούν οι διορθώσεις;
 - Οι αλλαγές στη διαδικασία αναζήτησης θα πραγματοποιθούν σε επόμενο Sprint
 - Ο Scrum Master θα πρέπει να απομακρύνει τα όποια εμπόδια και λοιπά στοιχεία αποσπούν την συγκέντρωση των μελών της ομάδας στο υφιστάμενο έργο

Συμπεράσματα εκτέλεσης πρώτου Sprint

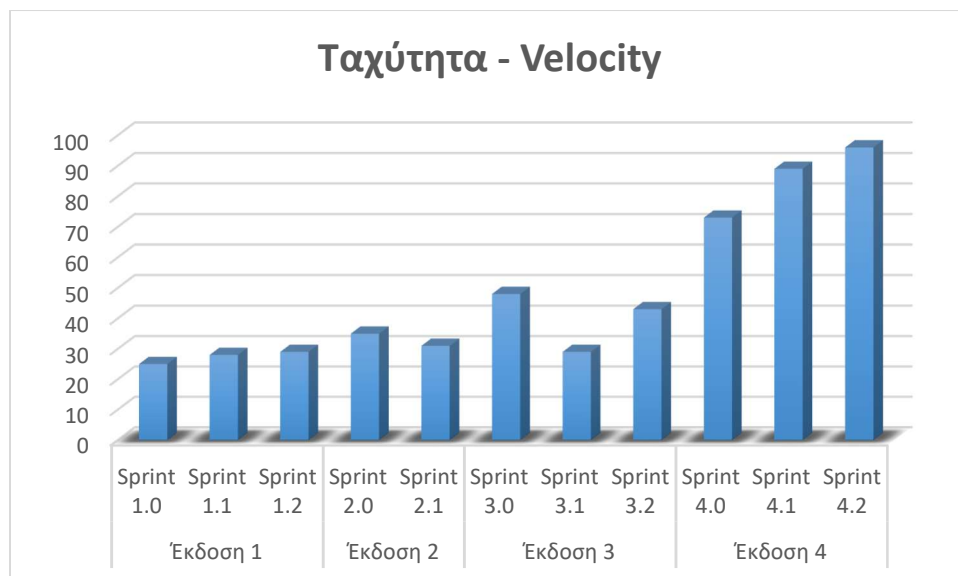
Για τον οργανισμό στον οποίο αναφέρεται η μελέτη περίπτωσης είναι η πρώτη φορά που εφαρμόζεται μία ευέλικτη μεθοδολογία, συνεπώς το πρώτο Sprint δεν ενδείκνυται για εξαγωγή ασφαλών συμπερασμάτων και λήψεις αποφάσεων. Ουσιαστικά, η ομάδα προσπαθεί να εμπεδώσει την διαδικασία των επαναλήψεων. Εδώ εξάγεται η βασική ταχύτητα (velocity) που είναι και η ταχύτητα ολοκλήρωσης του Sprint. Η μετρική αυτή χρησιμοποιείται ώστε να εκτιμηθεί ο χρόνος ολοκλήρωσης του έργου. Για παράδειγμα, αν η ταχύτητα του Sprint είναι 25 και ο αριθμός των καταγεγραμμένων απαιτήσεων σε άνθρωπο-χρόνο είναι 800, το έργο αναμένεται να ολοκληρωθεί σε 32 εβδομάδες.

8.6.4.2. Τα επόμενα Sprints

Τα επόμενα Sprints κάθε έκδοσης, ουσιαστικά, θα οδηγήσουν στην πλήρη ανάπτυξη της εφαρμογής. Με την έναρξη του δεύτερου Sprint επανεκτιμώνται κάποια από τα User Stories καθώς αποδείχθηκε ότι ήταν περισσότερο απαιτητικά από ότι αρχικά είχε εκτιμηθεί. Για παράδειγμα, η διαδικασία της αναζήτησης των νεοσυλλέκτων στην προβλεπόμενη σελίδα. Άλλα User Stories, από την άλλη αποδείχθηκαν πιο απλά με αποτέλεσμα να μειωθεί η βαθμολογία τους. Η παραπάνω διαδικασία της αναβαθμολόγησης είναι απολύτως αναμενόμενη στα πρώτα Sprint ενός έργου, ενώ όσο περνά ο χρόνος θα πρέπει να εξαλείφεται. Αξίζει, επίσης, να αναφερθεί ότι στο δεύτερο Sprint της έκδοσης 3.0, προέκυψαν ορισμένες αλλαγές σε σχέση με τις αρχικές απαιτήσεις του πελάτη (αρμόδια διεύθυνση στρατού) που αφορούσαν τον νέο νόμο περί

μοριοδότησης των νεοσυλλέκτων. Ως εκ τούτου, κάποιες σελίδες της εφαρμογής απαιτήθηκε να επανασχεδιαστούν, καθώς και το σχήμα της βάσης δεδομένων να μεταβληθεί. Φυσικά, αυτά τα νέα User Stories τοποθετήθηκαν στο Product Backlog, χωρίστηκαν σε μικρότερα tasks και τοποθετήθηκαν στα επόμενα Sprints αναλόγως της σημαντικότητάς τους.

Παρατηρήθηκε, επίσης, μία βελτίωση της ταχύτητας της ομάδας, αναφορικά με τον χρόνο από Sprint σε Sprint μέχρι και το δεύτερο Sprint της έκδοσης 3.0, όταν και προέκυψε το πρόβλημα που προαναφέραμε και είχε ως αποτέλεσμα τη πτώση της ταχύτητας ανάπτυξης. Ομοίως, και στο Sprint 2.1, λόγω ενός προγραμματιστικού προβλήματος η ταχύτητα εμφάνισε πτώση. Τα παραπάνω αποτυπώνονται στη επόμενη εικόνα:



Εικόνα 27. Ταχύτητα των Sprints στις εκδόσεις του Πληροφοριακού Συστήματος Κατάταξης Νεοσυλλέκτων

Με το πέρας όλων των Sprints, ουσιαστικά, ολοκληρώθηκε το προϊόν και ακολούθησε το ονομαζόμενο Feature Retrospective, το οποίο συνιστά μια συνεδρίαση, στην οποία παρευρίσκονται όλα τα μέλη της ομάδας Scrum, μαζί με τον Scrum Master και τον Product Owner και ως σκοπό έχει τη συζήτηση των προβλημάτων που αντιμετωπίστηκαν κατά τη διάρκεια όλων των Sprints αναπτύξεως του έργου. Επίσης, κατά τη διάρκεια αυτού, αναζητήθηκαν τρόποι βελτίωσης των διαδικασιών που ακολουθήθηκαν και επισημάνθηκαν τα θετικά και αρνητικά στοιχεία που παρατηρήθηκαν.

8.6.5. Συμπεράσματα Εφαρμογής της Μεθοδολογίας Scrum

Συνοψίζοντας τα αποτελέσματα και τις παρατηρήσεις όλων των Sprint Retrospectives, καθώς και αυτά του Feature Retrospective, καταλήγουμε στα κάτωθι θετικά και αρνητικά χαρακτηριστικά που αναγνωρίστηκαν κατά την εφαρμογή της μεθοδολογίας Scrum για το συγκεκριμένο έργο.

Ως θετικά χαρακτηριστικά θα πρέπει να αναφέρουμε επιγραμματικά τα εξής:

- Ομαδικό πνεύμα.
- Γρήγορη ανάπτυξη ικανοτήτων.
- Συντονισμός εργασιών.
- Βελτίωση εργαλείων.
- Υποστήριξη – συνεργασία μεταξύ των μελών της ομάδας Scrum.
- Ενθουσιασμός επί της νέας μεθοδολογίας για τον οργανισμό (πρώτη εφαρμογή).

Ως χαρακτηριστικά τα οποία επιδέχονται βελτίωσης θα πρέπει να αναφέρουμε ενδεικτικά τα εξής:

- Κάποια User Stories δεν ήταν αρκετά σαφή.
- Κάποιες εκ των απαιτήσεων δεν ήταν ξεκάθαρες από την αρχή.
- Το Product Backlog άλλαξε στην πορεία και άργησε ουσιαστικά να ολοκληρωθεί.

Από τη μελέτη περίπτωσης, που αφορά το έργο της ανάπτυξης του Πληροφοριακού Συστήματος Κατάταξης Νεοσυλλέκτων στον Στρατό Ξηράς, κατέστη σαφές ότι η εφαρμογή της μεθοδολογίας Scrum σε ένα παραγωγικό περιβάλλον εγκυμονεί κινδύνους, όχι όμως δυσάρεστες εκπλήξεις. Οι κίνδυνοι αυτοί υπάρχουν τόσο για τον πελάτη όσο και για την ομάδα ανάπτυξης του έργου. Είναι η ουσιαστική συμμετοχή του πελάτη στις δραστηριότητες εξέλιξης του έργου αλλά και η συμμετοχή όλων των μελών της ομάδας έργου που εξασφαλίζουν την υψηλή ποιότητα του παραγόμενου έργου. Για την ουσιαστική μετάβαση του οργανισμού που μελετήθηκε, αλλά και κάθε επιχείρησης γενικότερα στο κόσμο της ευέλικτης μεθοδολογίας, απαιτείται ένα συμπαγές πλαίσιο αξιολόγησης, καθώς και η συγκριτική αξιολόγηση των αποτελεσμάτων

στα έργα που πραγματώνονται. Είναι, τέλος, βέβαιο ότι η εφαρμογή μίας ευέλικτης μεθοδολογίας σε ένα και μόνο έργο , δεν αρκεί για την εξασφάλιση ασφαλών συμπερασμάτων.

9. Συμπεράσματα

Στόχος των ευέλικτων μεθοδολογιών είναι να παρέχουν σαφείς βελτιώσεις στη διαδικασία ανάπτυξης πληροφοριακών συστημάτων. Η ικανότητα να ανταποκρίνονται στις απαιτήσεις των πελατών, η αυξημένη παραγωγικότητα και η παράδοση ποιοτικότερων προϊόντων λογισμικού είναι ορισμένα μόνο από τα οφέλη που προσφέρουν. Οι ευέλικτες μεθοδολογίες ενσωμάτωσαν μία ευρεία συλλογή από καλές και δοκιμασμένες αξίες και πρακτικές που βοηθούν στην ανάπτυξη ποιοτικού λογισμικού. Επιπλέον, όμως, αποδεικνύεται και η αδυναμία τους να πραγματώσουν έργα σε κατανεμημένα περιβάλλοντα, στα οποία οι παραδοσιακές μεθοδολογίες, όπως του «καταρράκτη», τα καταφέρνουν ακόμα καλύτερα.

Μεταξύ αυτών των ευέλικτων μεθοδολογιών συγκαταλέγεται η Scrum, μία επαναληπτική και αυξητική μεθοδολογία ανάπτυξης έργων λογισμικού. Η επιτυχία της μεθόδου στηρίζεται στη διαδικασία με την οποία προσεγγίζει την ανάπτυξη του πληροφοριακού συστήματος, προσδίδοντας στη διαδικασία αυτή ευελιξία, με έμφαση στη δραστική βελτίωση της παραγωγικότητας των ομάδων ανάπτυξης, οι οποίες αυτό-οργανώνονται, καθώς και προσαρμοστικότητα στις αλλαγές που προκύπτουν κατά τη διάρκεια της.

Η μεθοδολογία Scrum, ουσιαστικά, υιοθετεί μία εμπειρική προσέγγιση εστιάζοντας στην ανάπτυξη πληροφοριακών συστημάτων από ομάδες με μικρό αριθμό μελών, που με τη μεγιστοποίηση των ικανοτήτων τους, θα ανταποκριθούν με ευέλικτο τρόπο στις όποιες προκλήσεις τους παρουσιαστούν. Χαρακτηριστικό της είναι ότι το σύνολο των εργασιών της ταξινομείται κατά απόλυτη σειρά προτεραιότητας με βάση συγκεκριμένα κριτήρια, ενώ κάθε σύνολο εργασιών περατώνεται σε σύντομες επαναλήψεις, τα Sprints. Στην καρδιά της λειτουργίας της μεθοδολογίας, βρίσκεται το καθημερινό Scrum, η καθημερινή, δηλαδή, ολιγόλεπτη συνάντηση κατά την οποία αναλύεται η πρόοδος του έργου και περιγράφονται οι επόμενες εργασίες, καθώς και τα ενδεχόμενα εμπόδια.

Δεδομένη είναι, αποδεδειγμένα, επίσης, η διείσδυση των ευέλικτων μεθοδολογιών (και της Scrum πιο συγκεκριμένα) σε παγκόσμιο επίπεδο, αφού

είναι σχεδιασμένες να παράγουν μέγιστη επιχειρηματική αξία και πληροφοριακά συστήματα υψηλής ποιότητας και λειτουργικότητας.

Είναι, λοιπόν, η Scrum η ιδανική μεθοδολογία για την ανάπτυξη έργων λογισμικού; Δυστυχώς δεν υπάρχει απάντηση σε αυτό το ερώτημα, διότι κάθε έργο λογισμικού είναι μοναδικό και παρουσιάζει τις δικές του ιδιαιτερότητες. Ίσως η υβριδική προσέγγιση , ο συνδυασμός, δηλαδή, περισσότερων της μίας μεθοδολογίας, να είναι η πιο κατάλληλη σήμερα.

Βιβλιογραφία

- Agile Alliance. (2001). *Manifesto for Agile Software Development*. Ανάκτηση από Manifesto for Agile Software Development: <http://www.agilemanifesto.org>
- Alleman, G. (2002). *Agile Project Management Methods for IT Projects, The Story of Managing Projects: A Global, Cross-Disciplinary Collection of Perspectives*. Berkeley: Greenwood Press.
- Anderson, A. B. (1998). *A Chrysler goes to extremes* . Distributed Computing.
- Bauer, F. et al. (1968). *Software Engineering: A Report on a Conference Sponsored by NATO Science Committee*. NATO.
- Beck K. (1999). *Embracing change with Extreme Programming*. IEEE Computer, Vol. 32, Issue 10.
- Beck, K. (2003). *Test Driven Development: By Example*. Addison-Wesley.
- Beck, K. (2004). *Extreme Programming explained: Embrace change*. Reading, Mass: Addison-Wesley.
- Beck, K. et al. (2001). *Manifesto for Agile Software Development*. Ανάκτηση Δεκέμβριος 15, 2017, από www.agilemanifesto.org
- Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R., Mellor, S., Schwaber, K., Sutherland, J. and Thomas, D. (2002). *Manifesto for Agile Software Development*.
- Bhardwaj, D. (2011). *Scrumming it up: A Survey on Current Software Industry Practices, University School of Information Technology*. Delhi: Guru Gobind Singh Indraprastha University.
- Boehm B. (1998). *A Spiral Model of Software Development and Enhancement, Computer*, (Τόμ. pp 61-72.). IEEE.
- Chow, T., Cao D. B. (2008). *A survey study of critical success factors in agile*. Journal of Systems and Software.

- Coad, P. Lefebvre, E. & De Luca, J. (1999). *Java Modeling In Color With UML: Enterprise Components and Process*. Prentice Hall International.
- Cocco, L., Mannaro, K., Concas, G. and Marchesi, M. (2001). *Simulating Kanban and Scrum vs. Waterfall with System Dynamics, Agile Processed in Software Engineering and Extreme Programming*. Lecture Notes in Business Information Processing, vol. 77(1), pp. 117-131.
- Cohen, Manion & Morrison. (2011). *Research Methods in Education*. UK: Routledge: Oxon.
- Cohn, M. (2006). *Agile Estimating and Planning*. Boulder: Prentice Hall.
- Cohn, M. (2008). *Prioritizing Product Backlog*. Chicago: Presentation at Chicago Scrum Gathering.
- Cohn, M. (2012). *Succeeding with Agile - Software Development using Scrum*. Michigan: Addison - Wesley.
- David Avison & Guy Fitzgerald. (2006). *Ανάπτυξη Προηγμένων Πληροφοριακών Συστημάτων: Μεθοδολογίες & Εργαλεία* (1η Έκδοση εκδ.). Αθήνα: Εκδόσεις Νέων Τεχνολογιών.
- Deemer et al. (2010). *The Scrum Primer* (Version 1.2 εκδ.). Scrum Training Institute.
- DSMD Consortium. (2012). *DSDM Tour*. Dynamic System Development Method.
- Excirial. (2009). *A quick overview of the Test-driven development lifecycle*. Ανάκτηση από Wikipedia.
- Fowler, M. (2005). *The New Methodology*.
- Highsmith, J. A. (2000). *Adaptive Software Development: A Collaborative Approach to Managing Complex Systems*. New York, NY: Dorset House Publishing.
- Highsmith, J. and Cockburn, A. (2004). *Agile Software Development: The Business of Innovation*. IEEE Computer.

- Hundermark, P. (2009). *Do Better Scrum An unofficial set of tips and insights into how implement scrum well* (2nd edition εκδ.). Scrum Sense.
- Ken Schwaber & Jeff Sutherland. (2015). *Ο Οδηγός του Scrum*. Ανάκτηση από Ο Οδηγός του Scrum: <http://www.scrumguides.org>
- Kerzner, H. (2001). *Strategic planning for project management using a project management maturity model*. New York: John Wiley & Sons, Inc.
- Kniber, H. (2007). *Scrum and XP from the Trenches - How we do Scrum*. C4 Media Inc., InfoQ Enterprise Software Development Series.
- Layton, M. C. (2012). *Agile Project Management For Dummies*. New Jersey: John Wiley & Sons.
- Lincoln & Denzin. (2005). *The Sage handbook of qualitative research* (3rd Edition εκδ.). CA: Sage: Thousand Oaks.
- Loeser, A. (2006). *Project Management and Scrum – A Side by Side Comparison*.
- Owen, R., Koskela, L.J., Henrich, G., and Codinhoto, R. (2006). *Is Agile Project Management Applicable To Construction?.* Ponteficia Universidad Catolica de Chile, Santiago, Chile: Proceedings 14th Annual Conference of the International Group for Lean Construction.
- Palmer , S.R. and Felsing J.M. (2002). *A Practical Guide to Feature-Driven Development*. Upper Saddle River, NJ: Prentice-Hall.
- Papke-Shields, Beise, K.E., Quan, J. (2009). *Do project managers practice what they preach, and does it matter to project success?* International Journal of Project Management, Corrected Proof.
- Pfleeger, S. L. (2003). *Τεχνολογία Λογισμικού, Θεωρία και Πράξη* . (2 εκδ.). (Τ. (Ι. Σταμέλος, Επιμ.) Εκδόσεις Κλειδάριθμος .
- PMBOK. (2004). *A Guide to the Project Management Body of Knowledge*. 3.Four Campus Boulevard Newton Square: USA: project management institute.

- Sauser B. et al. (2009). *System maturity incides for decision support in the defence acquisition process.*
- Schach. (1990). *Software Engineering, Vanderbilt University.* Aksen Association.
- Schwaber, K. a. (2001). *Agile Software Development with Scrum* (1st Edition εκδ.). Upper Saddle River, NJ: Prentice – Hall.
- Schwaber, K. and Beedle, M. (2001). *Agile Software Development with Scrum* (1 εκδ.). Upper Saddle River, NJ: Prentice – Hall.
- Schwaber, K. and Sutherland J. (2011). *The Scrum Guide, The Definitive Guide to Scrum: The Rules of the Game.* Scrum.org.
- Schwaber, K. (2008). *It's Not Scrum If...* Stockholm : Presentation at Stockholm Scrum Gathering.
- Stapleton, J. (1997). *Dynamic systems development method – The method in practice.* Addison Wesley.
- Sutherland, J. (2004). *Agile Development: Lessons learned from the first Scrum.*
- Takeuchi, H. and Nonaka, I. (1986). *The New Product Development Game.* Harvard Business Review.
- Βασιλακόπουλος Γ. (2018). *Πληροφοριακά Συστήματα.* Εκδόσεις Τσότρα.

Παράρτημα Α

Στο παρόν παράρτημα παρατίθεται η Μεθοδολογία Μαλακών Συστημάτων που πραγματώθηκε προς κατανόηση της προβληματικής κατάστασης, καθώς και η μοντελοποίηση του συστήματος της κατάταξης νεοσυλλέκτων στον ελληνικό στρατό.

Περιεχόμενα

1.Εισαγωγή.....	A-3
2. Θεωρητικά Στοιχεία	A-3
2.1 Ορισμοί A-3	
2.1.1 Σύστημα	A-3
2.1.2 Μεθοδολογία Μαλακών Συστημάτων	A-4
2.2 Φάσεις Μεθοδολογίας Μαλακών Συστημάτων A-4	
3. Εφαρμογή ΜΜΣ στο σύστημα «Πληροφοριακό Σύστημα Κατάταξης Νεοσυλλέκτων»	A-6
3.1 Φάση 1 ^η . Προσπάθεια αντίληψης της προβληματικής κατάστασης A-6	
3.2 Φάση 2 ^η . Κατασκευή της «Πλούσιας Εικόνας» A-7	
3.3 Φάση 3 ^η . Επιλογή θεμελιακού ορισμού A-8	
3.3.1 Θεμελιακός Ορισμός ΠΣΚΝ.....	A-9
3.3.2 Αναγωγή σε CATWOE	A-10
3.4 Φάση 4 ^η Κατασκευή εννοιολογικών μοντέλων A-11	
3.4.1 Εννοιολογικό μοντέλο του συστήματος θεμελιακού ορισμού.....	A-11
3.4.2 «Γνωστικό Σύστημα».....	A-12
3.4.3 Σύστημα «απόκτησης οργάνωσης και οργάνωσης πόρων»	A-13
3.4.4 «Σύστημα Κατάταξης Νεοσυλλέκτων»	A-14
3.4.5 «Σύστημα Παρακολούθησης και Ελέγχου». A-15	

3.5 Φάση 5 ^η . Σύγκριση των ιδεατών εννοιολογικών μοντέλων με την παρούσα κατάσταση	A-16
3.5.1 «Γνωστικό Σύστημα».....	A-17
3.5.2 «Σύστημα Κατάταξης Νεοσυλλέκτων»	A-17
3.5.3 Σύστημα «απόκτησης οργάνωσης και οργάνωσης πόρων»	A-18
3.5.4 «Σύστημα Παρακολούθησης και Ελέγχου»	A-18
3.6 Φάση 6 ^η . Πραγμάτωση εφικτών και επιθυμητών αλλαγών	A-18
3.7 Φάση 7 ^η . Υλοποίηση των αλλαγών	A-19
4. Συμπεράσματα – Προοπτικές	A-19
5. Βιβλιογραφία	A-20

1. Εισαγωγή

Η ανάγκη του ανθρώπου για την κατανόηση και την ερμηνεία των φαινομένων που συνέβαιναν γύρω του και ταυτόχρονα η αναγκαιότητα για εξεύρεση λύσεων στα προβλήματα που τον απασχολούσαν οδήγησε στην ανάπτυξη των επιστημονικών αρχών και νόμων. Η συστημική σκέψη (systems thinking) εμφανίστηκε εκεί όπου τα φαινόμενα και τα προβλήματα που απασχολούσαν τον άνθρωπο άρχισαν να γίνονται ολοένα και πιο περίπλοκα και οι αναλυτικές μέθοδοι της επιστήμης αδυνατούσαν να παρέχουν ικανοποιητικά αποτελέσματα. Η συστημική θεώρηση (systems approach) έρχεται σε αυτό ακριβώς το σημείο να αποτελέσει μια εναλλακτική και περισσότερο σφαιρική προσέγγιση της υπό εξέταση κατάστασης. Πιο συγκεκριμένα, μία από τις πιο γνωστές συστημικές μεθοδολογίες είναι η Μεθοδολογία Μαλακών Συστημάτων (ΜΜΣ). Πρόκειται για μία από τις πλέον σημαντικές συστημικές μεθοδολογίες που έχει εφαρμοστεί σε εκατοντάδες πραγματικές καταστάσεις με ικανοποιητικά αποτελέσματα.

2. Θεωρητικά Στοιχεία

2.1 Ορισμοί

Σκοπός της παρούσας εργασίας είναι η μοντελοποίηση, με τη χρήση της μεθόδου των μαλακών συστημάτων, του συστήματος της κατάταξης νεοσυλλέκτων στον ελληνικό στρατό. Κρίνεται σκόπιμο, ως εκ τούτου, να δοθούν κάποιοι απαραίτητοι ορισμοί που θα βοηθήσουν στην κατανόηση της μεθόδου που θα αναπτυχθεί.

2.1.1 Σύστημα

Οι Schoderbek et al (1990) ορίζουν το σύστημα ως *«ένα σύνολο αντικειμένων με τα χαρακτηριστικά γνωρίσματα τους, τα οποία είναι σε αλληλοσυσχέτιση μεταξύ τους και με το περιβάλλον, έτσι ώστε να αποτελούν μία ενιαία ολότητα»*.

Σύμφωνα με τον Βασιλακόπουλο (2016) *«σύστημα είναι ένα οργανωμένο και ολοκληρωμένο σύνολο από αλληλεξαρτώμενα και αλληλεπιδρώντα συστατικά στοιχεία (components). Ένα σύστημα έχει*

αντικειμενικούς σκοπούς ή στόχους που συχνά είναι δύσκολο να παρατηρηθούν».

«Κάθε σύστημα μπορεί να θεωρηθεί ως μια οντότητα η οποία δέχεται κάποιες εισροές (inputs) και παράγει κάποιες εκροές (outputs). Το σύστημα μέσω μιας διαδικασίας μετασχηματισμού (transformation process) μετατρέπει τις εισροές αυτές σε εκροές. Κάθε σύστημα επίσης αποτελείται από συστατικά στοιχεία (components) που ονομάζονται υποσυστήματα (sub-systems) και λειτουργεί και υπάρχει εντός ενός περιβάλλοντος (environment). Κάθε σύστημα έχει όρια (boundaries) και διαθέτει μηχανισμό παρακολούθησης της αποδοτικότητας για ανάληψη ελεγκτικής δράσης αν απαιτηθεί (monitoring and control mechanism)» [Βασιλακόπουλος, 2016].

2.1.2 Μεθοδολογία Μαλακών Συστημάτων

«Η μεθοδολογία μαλακών συστημάτων - ΜΜΣ (Soft Systems Methodology – SSM) συνιστά μια προσέγγιση για τη μοντελοποίηση οργανωσιακών ή επιχειρησιακών διεργασιών (organizational or business process modeling) και μπορεί να χρησιμοποιηθεί για την εντύπωση επί γενικών διοικητικών προβλημάτων και στην διοίκηση των αλλαγών» [Βασιλακόπουλος, 2016].

Για τον Checkland (1981), η μεθοδολογία «δεν είναι μία μέθοδος άλλα ένα σύνολο αρχών μιας μεθόδου που σε κάθε συγκεκριμένη κατάσταση πρέπει να αναχθεί σε μέθοδο, μοναδικά κατάλληλη για αυτήν τη συγκεκριμένη κατάσταση».

2.2 Φάσεις Μεθοδολογίας Μαλακών Συστημάτων

Παρακάτω καταγράφονται οι φάσεις της ΜΜΣ οι οποίες θα χρησιμοποιηθούν για τη μοντελοποίηση του «Πληροφοριακού Συστήματος Κατάταξης Νεοσυλλέκτων». Επιγραμματικά αναφέρονται ως εξής:

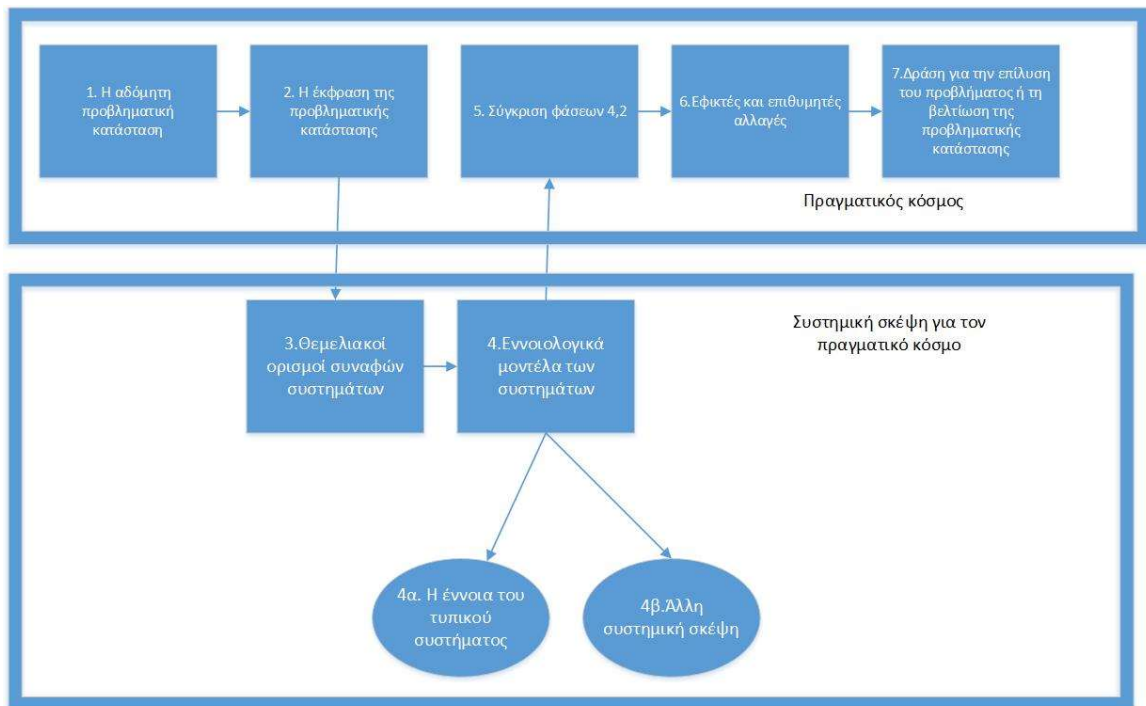
- Φάσεις 1 & 2: Αναγνώριση προβληματικής κατάστασης
- Φάση 3: Επιλογή θεμελιακού ορισμού
- Φάση 4: Κατασκευή εννοιολογικών μοντέλων

- **Φάση 5: Σύγκριση**
- **Φάσεις 6 & 7: Πραγμάτωση εφικτών και επιθυμητών αλλαγών**

Ειδικότερα, η ΜΜΣ περιλαμβάνει δύο ομάδες φάσεων:

Οι φάσεις 1, 2, 5, 6 και 7 αναφέρονται στον πραγματικό κόσμο και σ' αυτές συμμετέχουν τα άτομα που εμπλέκονται στην προβληματική κατάσταση.

Οι φάσεις 3, 4, 4α και 4β αναφέρονται στη συστημική σκέψη και σ' αυτές μπορεί να συμμετέχουν, ή όχι, άτομα που εμπλέκονται στην προβληματική κατάσταση ανάλογα με τις επιμέρους ιδιομορφίες της μελέτης.



Διάγραμμα 1. Διαγραμματική παρουσίαση της ΜΜΣ.

3. Εφαρμογή ΜΜΣ στο σύστημα «Πληροφοριακό Σύστημα Κατάταξης Νεοσυλλέκτων»

3.1 Φάση 1^η . Προσπάθεια αντίληψης της προβληματικής κατάστασης

Σκοπός της πρώτης φάσης είναι να δοθεί μία λεπτομερής δυνατή προσέγγιση που θα επιτρέψει την εισχώρηση στην προβληματική κατάσταση. Στην προκειμένη περίπτωση, η προβληματική αυτή κατάσταση έγκειται στην κατάταξη των νεοσυλλέκτων στον στρατό ξηράς.

Ως εκ τούτου, αρχικά καταγράφονται τα στοιχεία που αφορούν στην οργανωτική δομή της διαδικασίας κατάταξης ως σύστημα. Ωστόσο, πρέπει να σημειωθεί ότι αυτό απαιτεί τη συλλογή όσο το δυνατόν περισσότερων απόψεων από τους εμπλεκόμενους στην προβληματική κατάσταση, χωρίς όμως να έχει σκοπό την πλήρη αποδόμηση του συστήματος, καθώς κάτι τέτοιο είναι πρακτικά αδύνατο και πολλές φορές αρνητικό [Armson,2001. Checkland 1981,2000].

Η οργανωτική δομή μπορεί να εξετασθεί από τη στιγμή που ο νεοσύλλεκτος εισέρχεται στον χώρο της κατάταξης και συγκεκριμένα στον χώρο που υπάρχει η υποδομή για καταχώρηση των στοιχείων του. Μετά από σειρά ερωτήσεων του στρατολόγου στον νεοσύλλεκτο, ο στρατολόγος καταχωρεί τα απαραίτητα προσωπικά στοιχεία του νεοσυλλέκτου και τα καταγράφει στις ειδικές καρτέλες του πληροφοριακού συστήματος της κατάταξης. Για την παραπάνω διαδικασία απαιτείται ο στρατολόγος να έχει στη διάθεσή του το αρχείο της εφαρμογής με τα προσυμπληρωμένα στοιχεία κάθε νεοσυλλέκτου, το οποίο παραλαμβάνει από το Κέντρο Πληροφοριακής Υποστήριξης του Ελληνικού Στρατού (ΚΕΠΥΕΣ). Με το πέρας της διαδικασίας, το νέο αρχείο θα πρέπει να αποσταλεί στα κεντρικά γραφεία του ΚΕΠΥΕΣ για να ολοκληρωθεί η διαδικασία.

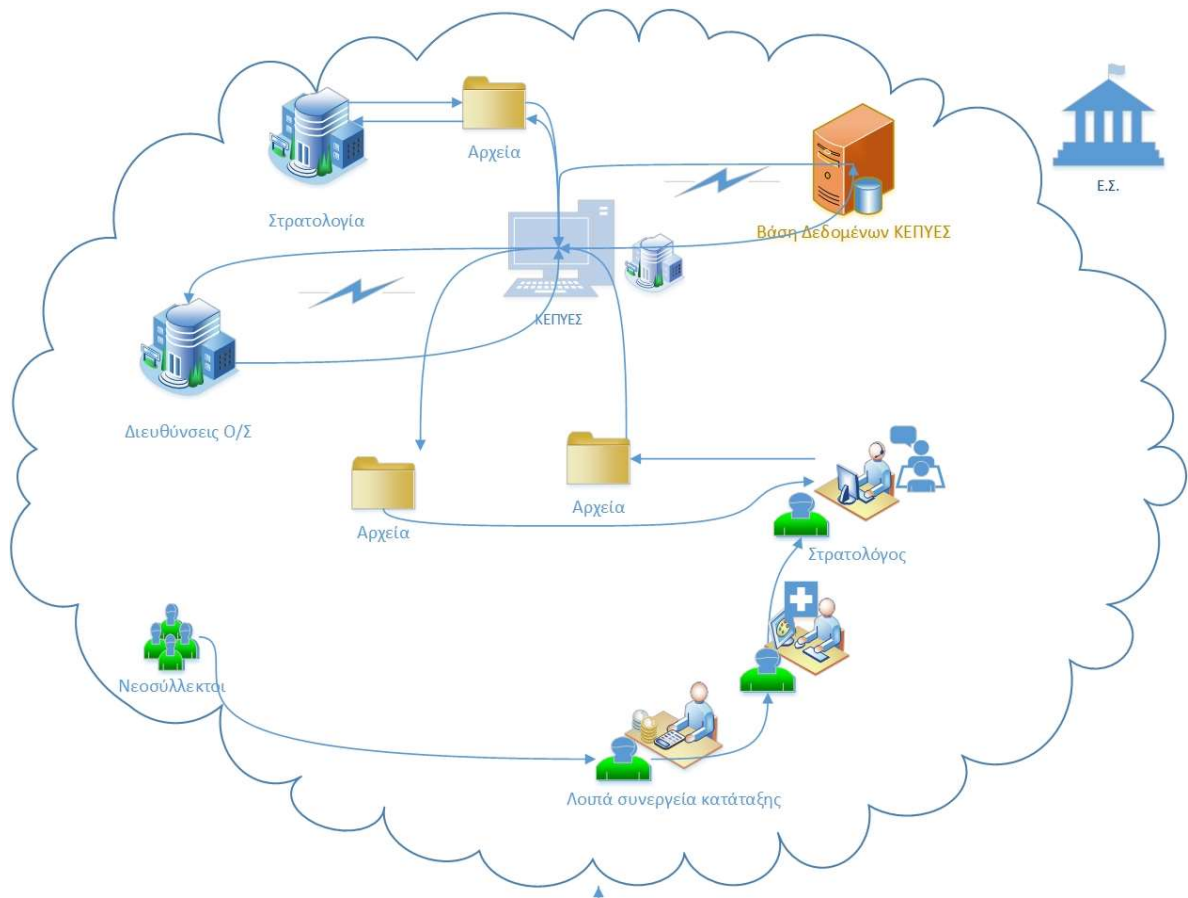
Όσον αφορά στους στρατολόγους, είναι κατάλληλα εκπαιδευμένοι, εξειδικευμένοι και καταρτισμένοι στον τομέα της κατάταξης. Επιπλέον, οι στρατολόγοι οφείλουν να παρακολουθούν την πρόοδο των γνώσεων της επιστήμης τους (state of the art), καθώς και να συμμετέχουν σε διάφορα ερευνητικά έργα, προγράμματα και μετεκπαιδεύσεις της ειδικότητάς τους.

Σκοπός του Πληροφοριακού Συστήματος Κατάταξης Νεοσυλλέκτων (ΠΣΚΝ) είναι οι στρατολόγοι να απαλλαγούν από την οποιαδήποτε μορφή χειρόγραφων εντύπων κατά την κατάταξη, να αυτοματοποιηθούν οι διαδικασίες που αφορούν την κατάταξη των νεοσυλλέκτων και να απαλλαγούν από την αλληλο-αποστολή αρχείων μεταξύ κέντρων κατάταξης, Διεύθυνσης Οπλιτών Θητείας και ΚΕΠΥΕΣ.

Θα πρέπει να αναφερθεί σε αυτό το σημείο ότι η υλοποίηση του ΠΣΚΝ εξαρτάται και από επιχειρησιακές διαδικασίες που αφορούν νομοθετικούς, τεχνολογικούς και γνωσιακούς περιορισμούς.

3.2 Φάση 2^η . Κατασκευή της «Πλούσιας Εικόνας»

Στο στάδιο αυτό, οι απόψεις που συγκεντρώθηκαν προηγουμένως καταγράφονται και παρουσιάζονται σε μια «πλούσια εικόνα», η οποία αποτελεί ένα βοήθημα για την καλύτερη κατανόηση της προβληματικής κατάστασης. Σύμφωνα με τον Checkland (1981), θα πρέπει η παρουσίαση να γίνεται με τρόπο που να αναδεικνύονται τα στοιχεία δομής, διαδικασίας και οι συσχετίσεις μεταξύ τους. Αυτά θα φανερώσουν και το «κλίμα» που επικρατεί στο υφιστάμενο σύστημα (Κιουντούζης, 2002). Με πιο απλά λόγια, με την «πλούσια εικόνα» μπορούμε να αποτυπώσουμε και να αναπαραστήσουμε την προβληματική κατάσταση σε μια μόνο κόλλα χαρτί, ενώ η γραπτή διατύπωσή της θα απαιτούσε πολύ περισσότερες. Στην πραγματικότητα, μέσω της δημιουργίας της «πλούσιας εικόνας» επιτυγχάνεται η δημιουργία ενός πλαισίου επικοινωνίας μεταξύ των εμπλεκομένων (Flood και Jackson, 1996). Η ΜΜΣ δεν προτείνει κάποια μέθοδο κατασκευής της «πλούσιας εικόνας» και αφήνεται στην κρίση του αναλυτή (Williams, 1999). Στο διάγραμμα 2 παρουσιάζεται η «πλούσια εικόνα» του ΠΣΚΝ.



Διάγραμμα 2. «Πλούσια Εικόνα» Πληροφοριακού Συστήματος Κατάταξης Νεοσυλλέκτων

3.3 Φάση 3^η . Επιλογή θεμελιακού ορισμού

Η κατασκευή της «πλούσιας εικόνας» μας δίνει τη δυνατότητα εντοπισμού ενός εύρους συστημάτων που σχετίζονται με το πρόβλημα. Ο ορισμός που θα δοθεί παρακάτω αποτελεί μια εξιδανικευμένη περιγραφή του συστήματος όπως θα έπρεπε να είναι. Σε καμία περίπτωση, φυσικά, το «σχετικό» αυτό σύστημα δε σημαίνει ότι είναι και το επιθυμητό, όπως επίσης και το ότι είναι και αυτό που θα πρέπει να αντικαταστήσει το υπάρχον στον πραγματικό κόσμο.

Ο ορισμός που θα δοθεί στη συνέχεια περιλαμβάνει τα εξής στοιχεία που κωδικοποιούνται στο ακρωνύμιο **CATWOE** (Βασιλακόπουλος, 2016):

- *Πελάτες (customers - C), δηλαδή τα άτομα που επηρεάζονται (θετικά ή αρνητικά) από τις δραστηριότητες του συστήματος.*

- *Λειτουργούς (actors - A), δηλαδή τα άτομα που εκτελούν ή προκαλούν την εκτέλεση των κύριων δραστηριοτήτων του συστήματος.*
- *Μετασχηματισμό (transformation - T), δηλαδή την επεξεργασία κατά την οποία οι εισροές μετατρέπονται σε εκροές. Ο μετασχηματισμός αποτελεί τον πυρήνα του συστήματος.*
- *Κοσμοθεώρηση (world view - weltanschauungen - W), δηλαδή μια άποψη, ένα πλαίσιο ή μια παραστατική αντίληψη του κόσμου.*
- *Ιδιοκτησία (ownership - O), δηλαδή τα άτομα που έχουν την κυριότητα του συστήματος και, κατά συνέπεια, τη δύναμη να προκαλέσουν τον τερματισμό της ύπαρξης και της λειτουργίας του.*
- *Περιβάλλον (environment - E), δηλαδή τους περιορισμούς επί του συστήματος που επιβάλλονται από τον περιβάλλοντα χώρο στον οποίο υπάρχει και λειτουργεί.*

3.3.1 Θεμελιακός Ορισμός ΠΣΚΝ

Με βάση, λοιπόν, τα παραπάνω δίδεται ο θεμελιακός ορισμός του ΠΣΚΝ:

«Το Πληροφοριακό Σύστημα Κατάταξης Νεοσυλλέκτων (ΠΣΚΝ) είναι ένα σύστημα που ανήκει στη Διεύθυνση Οπλιτών Θητείας, παρέχεται δωρεάν στους νεοσύλλεκτους στρατιώτες και λειτουργεί βάση του νομικού πλαισίου που ορίζεται από το Υπουργείο Εθνικής Άμυνας σχετικά με τη στράτευση, καθώς και των προεδρικών διαταγμάτων και κανονισμών που το αφορούν. Περιλαμβάνει μία διαδικασία εργασιών ώστε να καταχωρούνται από τους κατάλληλους εξειδικευμένους στρατολόγους τα στοιχεία των νεοσύλλεκτων στρατιωτών. Υπό την οργανωτική δομή αυτοματοποιημένων διαδικασιών και στα πλαίσια που ορίζουν οι σύγχρονοι στρατιωτικοί, υγειονομικοί, ηθικοί, πολιτιστικοί και κοινωνικοί κανόνες, αλλά και τα ανάλογα πρότυπα κατάταξης πρέπει να παρέχει αυτές τις υπηρεσίες αδιάλειπτα και με ταχύτητα αλλά και με τη τελευταία λέξη της τεχνολογία (state of the art). Έχει σαν ειδικότερο σκοπό τη αυτοματοποίηση και τυποποίηση της κατάταξης νεοσυλλέκτων στρατιωτών . Επιπροσθέτως το ΠΣΚΝ στοχεύει στην καλύτερη στρατιωτική θητεία των νεοσυλλέκτων».

3.3.2 Αναγωγή σε CATWOE

Τα παραπάνω ανάγονται σε στοιχεία CATWOE κατά την ακόλουθη περιγραφή:

Ο βασικός μετασχηματισμός (Τ) στον οποίο επιδίδεται αυτό το σύστημα είναι ότι λαμβάνει ως είσοδο τους νεοσύλλεκτους στρατιώτες και με τους κατάλληλους πόρους (επαγγελματικές ικανότητες, ιατρική γνώση, επιστημονική γνώση, εγκαταστάσεις, εξοπλισμό και υλικά, κτλ.) που χρησιμοποιεί για την παροχή της αδιάλειπτης, τάχιστης και αποτελεσματικής κατάταξης, έχοντας ενημερωμένη και επικυρωμένη στρατολογική γνώση, «μετατρέπει» τους νεοσύλλεκτους αυτούς σε εξυπηρετούμενους καταταγμένους στρατιώτες, έτοιμους να υπηρετήσουν τη θητεία τους στις μονάδες του Ε.Σ. Τα στοιχεία εξόδου, δηλαδή, είναι οι στρατιώτες που αποδίδονται στις μονάδες τους αλλά και οι πληροφοριακές ροές προς την Διεύθυνση Οπλιτών Θητείας, τον Ελληνικό Στρατό και το Υπουργείο Εθνικής Άμυνας. Να σημειωθεί εδώ ότι *«για τα στοιχεία εισόδου και εξόδου χρησιμοποιούμε οντότητες και για τη μετατροπή τους από τη μία μορφή στην άλλη ρήματα (Basden και Wood-Harper, 2006)»*.

Οι λειτουργοί (Α) του συστήματος είναι το στρατολογικό και βοηθητικό προσωπικό που συμβάλλει για την αδιάλειπτη παροχή αποτελεσματικής και αποδοτικής κατάταξης και ο ωφελημένος ή ζημιωμένος από το σύστημα (C) είναι οι νεοσύλλεκτοι στρατιώτες, η Διεύθυνση Οπλιτών Θητείας, ο Ελληνικός Στρατός, το Υπουργείο Εθνικής Άμυνας και η κοινωνία.

Η Διεύθυνση Οπλιτών Θητείας είναι επίσης η ιδιοκτήτης (O) του συστήματος υπό την έννοια που αναφέρθηκε παραπάνω, ότι δηλαδή, μόνο αυτή μπορεί να αποφασίσει να καταργήσει το σύστημα και να παύσει τη λειτουργία του. Οι κύριοι περιορισμοί του περιβάλλοντος (E) στους οποίους υπόκειται η λειτουργία αυτού του συστήματος αναφέρθηκαν παραπάνω και η κοσμοθεώρηση (W) που ενυπάρχει στο θεμελιακό ορισμό είναι ότι το σύστημα λειτουργεί σε έναν κόσμο στον οποίο η παροχή ποιοτικής κατάταξης είναι απαραίτητο συστατικό ενός σύγχρονου στρατού, αλλά και γενικότερα μιας αναπτυγμένης, ανταγωνιστικής και προηγμένης κοινωνίας.

3.4 Φάση 4^η Κατασκευή εννοιολογικών μοντέλων

Στο στάδιο αυτό θα κατασκευάσουμε το μοντέλο αυτό που επιτυγχάνει τον θεμελιακό ορισμό που δώσαμε παραπάνω. Ο μεγαλύτερος κίνδυνος στο σημείο αυτό είναι ο επηρεασμός από την πραγματικότητα. Το εννοιολογικό μοντέλο πρέπει να είναι αποτέλεσμα του ριζικού-θεμελιακού ορισμού και μόνο.

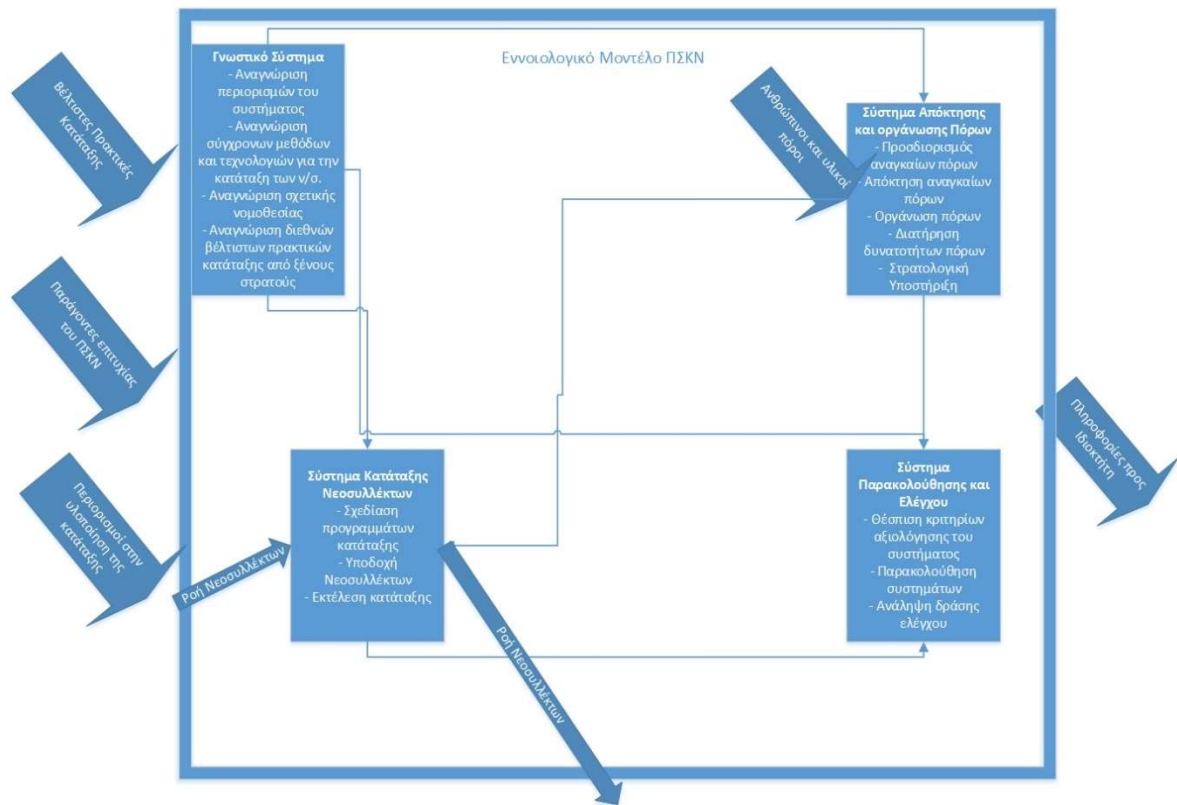
«Το αρχικό εννοιολογικό μοντέλο κατασκευάζεται σε ανώτερο επίπεδο αφαίρεσης (ανώτερο επίπεδο ανάλυσης ή μικρό βαθμό λεπτομέρειας). Κατόπιν, κάθε κύρια δραστηριότητα του μοντέλου αναπτύσσεται περαιτέρω (σε κατώτερο επίπεδο αφαίρεσης) κατά την έννοια των ιεραρχικών συστημάτων. Η ιεραρχική διαδικασία ανάπτυξης μοντέλων σταματά στο επίπεδο αφαίρεσης (επίπεδο ανάλυσης ή βαθμό λεπτομέρειας) που, κατά την κρίση του αναλυτή, περιγράφει στον απαιτούμενο βαθμό τις δραστηριότητες του αντιστοίχου συστήματος» (Βασιλακόπουλος, 2016).

Ο ελάχιστος αριθμός υποσυστημάτων που θα πρέπει να υπάρχουν σε ένα εννοιολογικό μοντέλο είναι:

- ένα “γνωστικό” υποσύστημα που έχει ως σκοπό τη διαχείριση της γνώσης του συστήματος,
- ένα “επιχειρησιακό” υποσύστημα που έχει ως σκοπό την εκτέλεση των επιχειρησιακής (παραγωγικής) δραστηριότητας του συστήματος, και
- ένα (υπο)σύστημα “παρακολούθησης & ελέγχου” το οποίο θέτει συγκεκριμένα κριτήρια αποτελεσματικότητας και αποδοτικότητας του συστήματος, λαμβάνει πληροφορίες για όλες τις δραστηριότητες του συστήματος με βάση αυτά τα κριτήρια προκειμένου να παρακολουθεί την αποδοτικότητα και αποτελεσματικότητα του συστήματος και αναλαμβάνει διορθωτική δράση αν, όπου και όταν απαιτείται (Βασιλακόπουλος, 2016).

3.4.1 Εννοιολογικό μοντέλο του συστήματος θεμελιακού ορισμού

Στο διάγραμμα 3 αναπαρίστανται το εννοιολογικό μοντέλο του συστήματος που περιεγράφηκε στον θεμελιακό ορισμό.



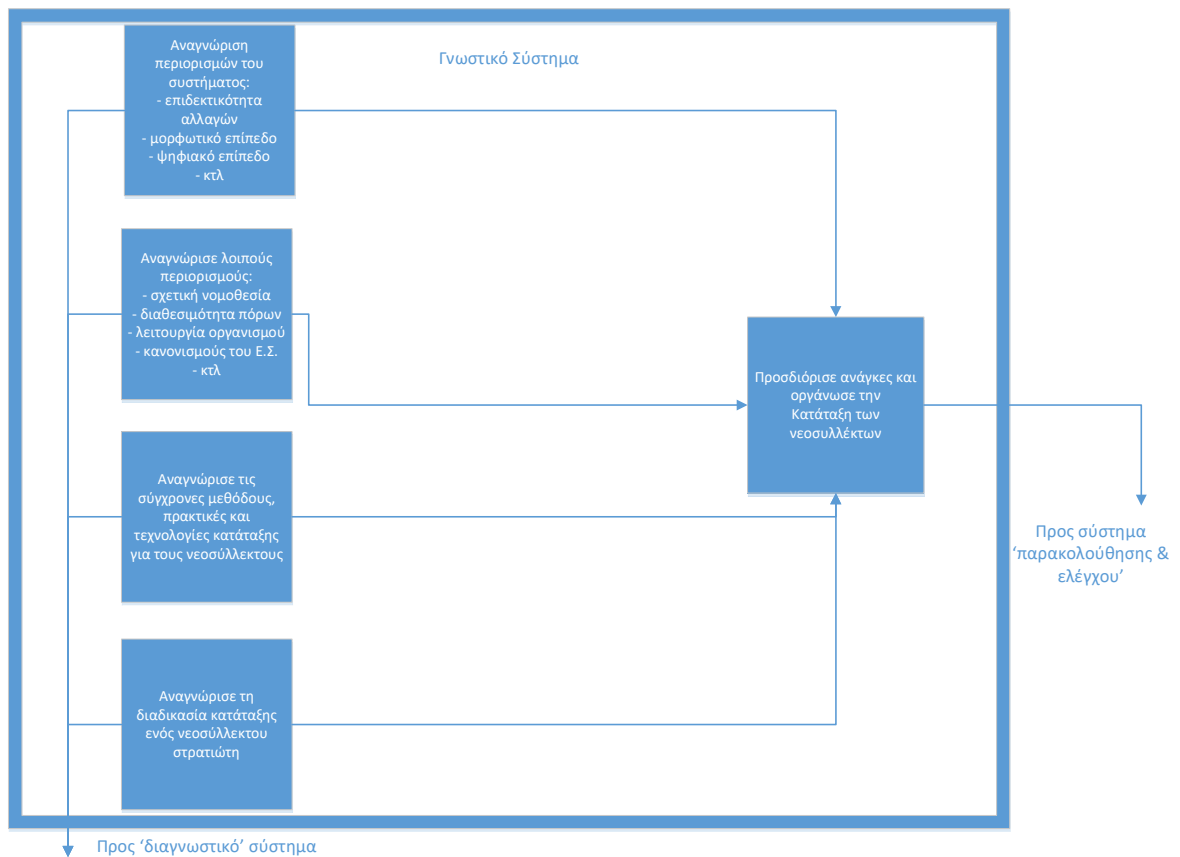
Διάγραμμα 3. Εννοιολογικό μοντέλο του συστήματος που περιγράφηκε στο θεμελιακό ορισμό.

Σε πρώτο επίπεδο αφαίρεσης, λοιπόν, διακρίνονται οι ακόλουθες βασικές δραστηριότητες που είναι το «γνωστικό σύστημα», το «σύστημα απόκτησης και οργάνωσης πόρων», «το σύστημα κατάταξης νεοσυλλέκτων» και το «σύστημα παρακολούθησης και ελέγχου».

3.4.2 «Γνωστικό Σύστημα»

Η ύπαρξη αυτού του υποσυστήματος σημαίνει ότι το σύστημα πρέπει να διαθέτει τις κατάλληλες γνώσεις, να αναγνωρίζει τις πρακτικές κατάταξης ξένων στρατών, αλλά και να αναγνωρίζει τους περιορισμούς από τη σχετική νομοθεσία καθώς επίσης και τυχόν άλλους περιορισμούς.

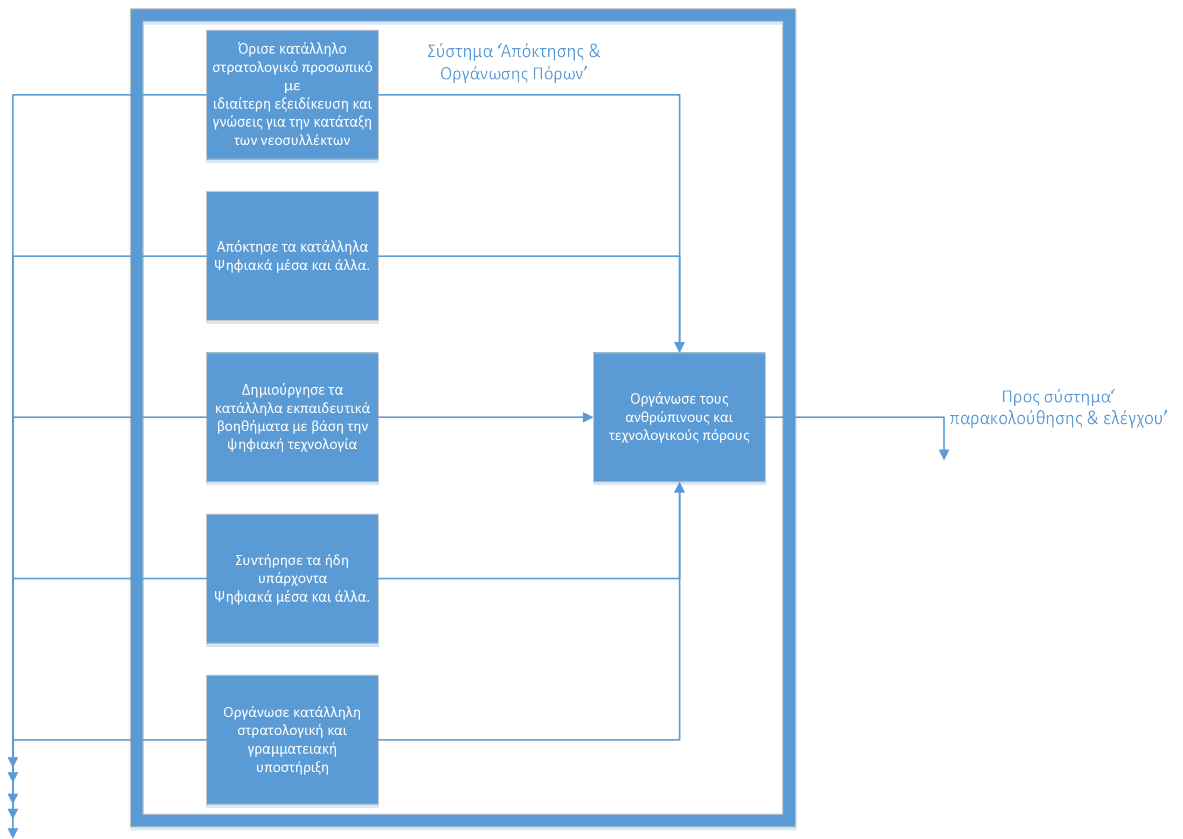
Σε μεγαλύτερο επίπεδο ανάλυσης η δομή του «Γνωστικού Συστήματος» φαίνεται στο διάγραμμα 4.



Διάγραμμα 4. Μοντέλο του γνωστικού (υπό)συστήματος σε μεγαλύτερο επίπεδο λεπτομέρειας.

3.4.3 Σύστημα «απόκτησης οργάνωσης και οργάνωσης πόρων»

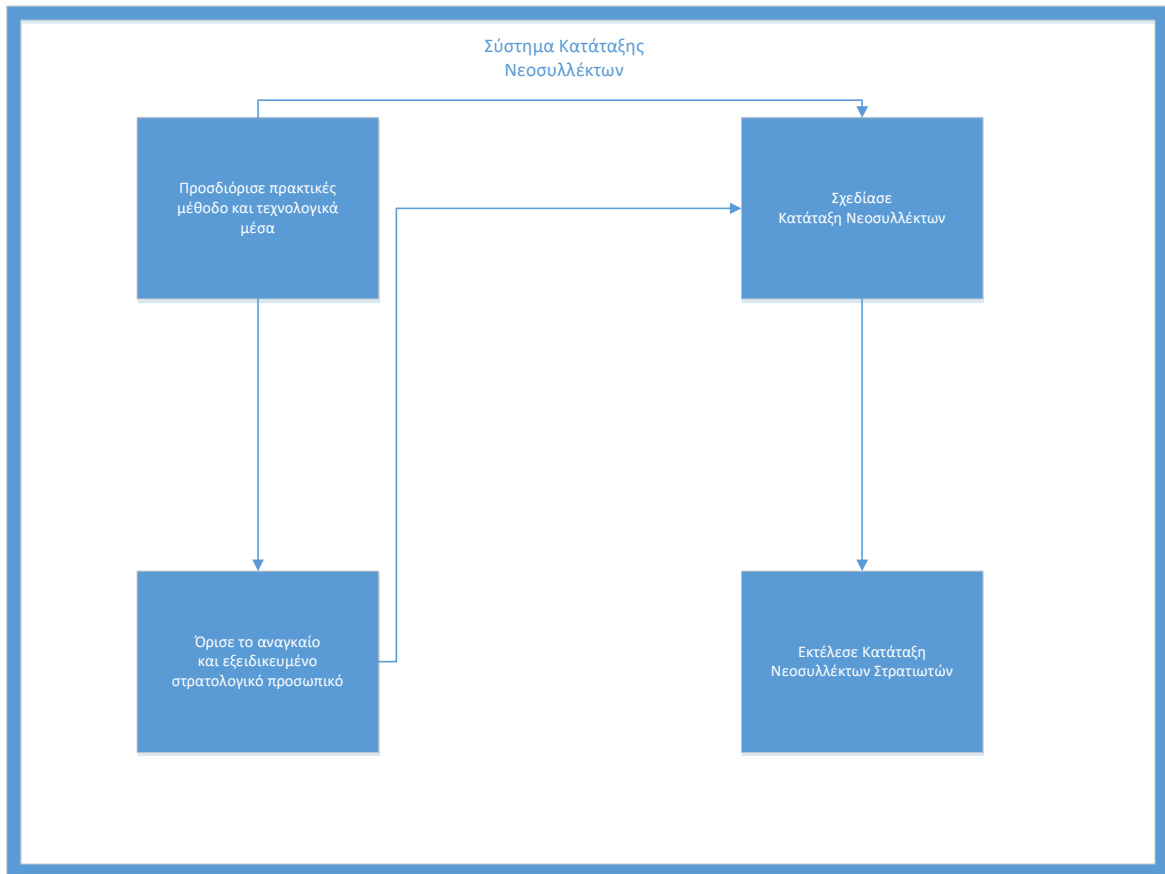
Το σύστημα αυτό (βλ. διάγραμμα 5) θα συλλέξει όλους τους πόρους που είναι διαθέσιμοι (υλικοί και άυλοι) όπως χρήματα, προσωπικό, μέσα κ.α., θα τους καταμετρήσει, θα τους οργανώσει και θα προσπαθήσει να τους καταμερίσει με τον βέλτιστο τρόπο.



Διάγραμμα 5. Μοντέλο του (υπό)συστήματος απόκτησης οργάνωσης και διοίκησης πόρων σε μεγαλύτερο επίπεδο λεπτομέρειας.

3.4.4 «Σύστημα Κατάταξης Νεοσυλλέκτων»

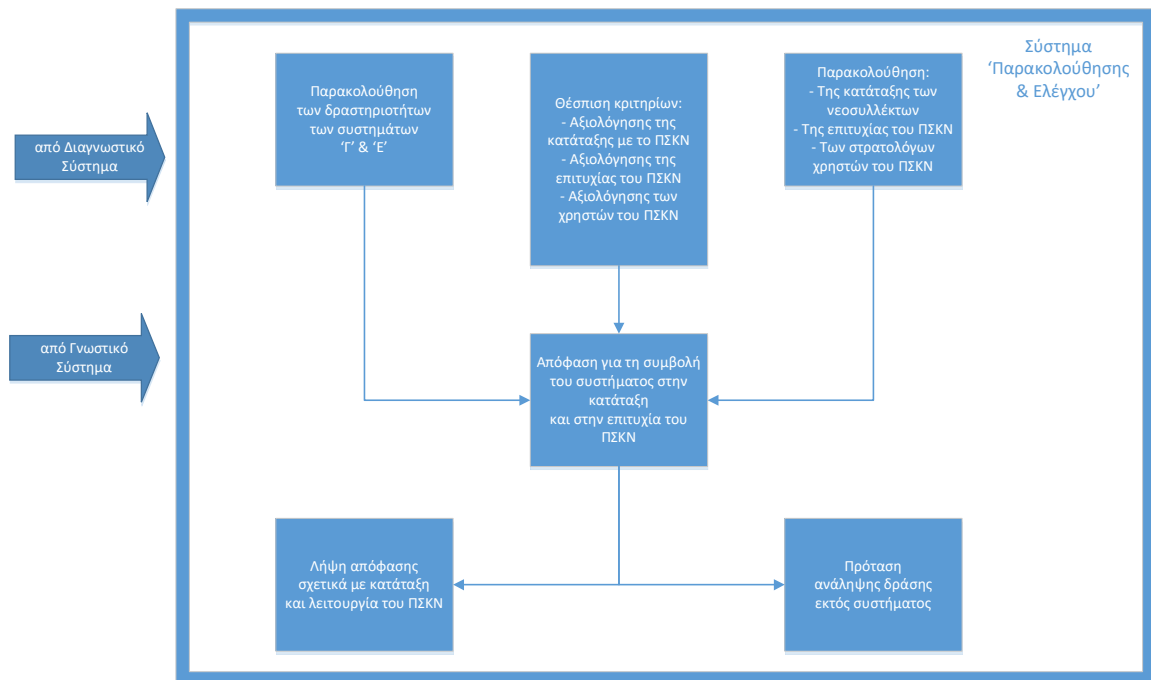
Το σύστημα αυτό θα έχει διαθέσιμη τη γνώση που αποκτάται στο γνωστικό σύστημα, τις ανάγκες κατάταξης των νεοσυλλέκτων του ΠΣΚΝ και την οργάνωση των πόρων, ενώ οι κύριες δραστηριότητές του είναι να σχεδιάζει και να εκτελεί το πρόγραμμα κατάταξης χρησιμοποιώντας το κατάλληλο προσωπικό, τις κατάλληλες πρακτικές και μεθόδους, καθώς και την κατάλληλη ψηφιακή τεχνολογία.



Διάγραμμα 6. Μοντέλο του (υπό)συστήματος Κατάταξης Νεοσυλλέκτων σε μεγαλύτερο επίπεδο λεπτομέρειας.

3.4.5 «Σύστημα Παρακολούθησης και Ελέγχου».

Προκειμένου να διασφαλιστεί ότι η κατάταξη των νεοσυλλέκτων του ΠΣΚΝ είναι τόσο αποτελεσματική και αποδοτική, ώστε το σύστημα να συμβάλει θετικά στην επιτυχία του ΠΣΚΝ, είναι απαραίτητη η παρακολούθηση και ο έλεγχος των λειτουργιών του συστήματος αναφορικά με την παρεχόμενη ποιότητα κατάταξης, την πορεία εξέλιξης του ΠΣΚΝ και την επιτυχία του ΠΣΚΝ, με βάση αντίστοιχα σύνολα κριτηρίων, αλλά και η αξιολόγηση των βοηθητικών υπηρεσιών που λαμβάνουν χώρα προκειμένου η κατάταξη να διεξάγεται επιτυχώς.



Διάγραμμα 7. Μοντέλο του (υπό)συστήματος «Σύστημα Παρακολούθησης και Ελέγχου» σε μεγαλύτερο επίπεδο λεπτομέρειας.

3.5 Φάση 5^η. Σύγκριση των ιδεατών εννοιολογικών μοντέλων με την παρούσα κατάσταση

Στη φάση 5 τα εννοιολογικά μοντέλα που κατασκευάστηκαν στη φάση 4 με βάση τη συστημική σκέψη μεταφέρονται στον πραγματικό κόσμο και συγκρίνονται με την προβληματική κατάσταση με σκοπό την ανάπτυξη διαλόγου μεταξύ όλων των ενδιαφερομένων μερών, δηλαδή των αναλυτών και των εκπροσώπων του συστήματος, προκειμένου να εντοπιστούν λύσεις που είναι εφικτές και επιθυμητές (Βασιλακόπουλος, 2016).

Πρέπει να γίνει κατανοητό ότι ο παρατηρητής στην παρούσα φάση έχει στη διάθεσή του δύο διαφορετικά στοιχεία. Την πλούσια εικόνα, που είναι το αποτέλεσμα του δεύτερου σταδίου και ανταποκρίνεται στην πραγματικότητα και το εννοιολογικό μοντέλο που παρουσιάζει την ιδανική μορφή που θα έπρεπε να έχει το σύστημα που μελετά.

Ο Checkland (1981) υποστηρίζει ότι η φάση της σύγκρισης είναι ένα στάδιο της μεθοδολογίας μαλακών συστημάτων που τη διαχωρίζει από τις «σκληρές» συστημικές προσεγγίσεις, αφού δεν υφίσταται η αντίστοιχη διαδικασία σε αυτές.

Επομένως, στο στάδιο της σύγκρισης, αυτό που ουσιαστικά πρέπει να κάνουμε είναι να διαπιστώσουμε «*εάν οι προτάσεις του εννοιολογικού μοντέλου αντιπροσωπεύονται με κάποιον τρόπο στην πραγματικότητα και εάν ναι, τότε να ερευνήσουμε εάν ο τρόπος αυτός μπορεί να αντικατασταθεί με κάποιον καταλληλότερο*» (Ledington, 1999).

Βάσει των παραπάνω λοιπόν τα συστήματα συνοψίζονται :

3.5.1 «Γνωστικό Σύστημα»

Το υφιστάμενο γνωστικό σύστημα διαφέρει σε κάποιο βαθμό από το σύστημα του μοντέλου, καθώς η υπάρχουσα εφαρμογή κατάταξης είναι γραμμένη σε γλώσσα Access με όλους τους περιορισμούς που αυτή συνεπάγεται, ενώ στη διαδικασία υπάρχει αλληλομεταφορά αρχείων μεταξύ Διεύθυνσης Οπλιτών Θητείας και ΚΕΠΥΕΣ καθώς και Κέντρων Κατάταξης και ΚΕΠΥΕΣ. Επίσης, υπάρχουν περιορισμοί που απορρέουν από τη νομοθεσία, όπως για παράδειγμα η νομοθεσία περί Προστασίας Προσωπικών Δεδομένων με αποτέλεσμα η εφαρμογή να μην είναι δυνατόν να υπάρχει και να λειτουργεί στο διαδίκτυο, αλλά σε ιδιόκτητο δίκτυο του Ε.Σ.

3.5.2 «Σύστημα Κατάταξης Νεοσυλλέκτων»

Το υφιστάμενο σύστημα *κατάταξης* των νεοσυλλέκτων είναι σε κάποιο μικρό βαθμό παρόμοιο με το αντίστοιχο του μοντέλου, καθώς οι περισσότερες διαδικασίες που ακολουθούνται είναι βάση διαταγών και νόμων. Ωστόσο, στο υφιστάμενο σύστημα υπάρχουν πολλές διαδικασίες που δυσχεραίνουν το έργο της κατάταξης. Για παράδειγμα, ο στρατολόγος πριν την κατάταξη θα πρέπει να παραλάβει ένα αρχείο από το ΚΕΠΥΕΣ με προσυμπληρωμένα στοιχεία των στρατιωτών, να το περάσει τοπικά στην Access εφαρμογή και στη συνέχεια με το πέρας της κατάταξης να εξάγει ένα νέο αρχείο από το τοπικό δίκτυο και να το αποστείλει πίσω στο ΚΕΠΥΕΣ. Τούτο, επιβραδύνει τη διαδικασία, αλλά ταυτόχρονα αυξάνει και την πιθανότητα σφαλμάτων και δυσλειτουργιών. Στο σύστημα του μοντέλου, όλα λειτουργούν σε ενιαίο ιδιόκτητο δίκτυο με web εφαρμογή και δεν υπάρχει η μεταφορά αρχείων και η χειροκίνητη φόρτωση αυτών στην εφαρμογή. Επίσης, χρησιμοποιούνται όλες οι νέες τεχνολογίες των web εφαρμογών με όλα τα πλεονεκτήματα που αυτές προσφέρουν στο

σύγχρονο ψηφιακό περιβάλλον, τόσο από πλευράς απόδοσης, συντήρησης αλλά και ασφάλειας.

3.5.3 Σύστημα «απόκτησης οργάνωσης και οργάνωσης πόρων»

Αναφορικά με το σύστημα *απόκτησης και οργάνωσης πόρων*, παρατηρούμε ότι κάποιες από τις διαδικασίες που περιλαμβάνονται στο μοντέλο, τηρούνται από την αντίστοιχη υφιστάμενη δομή, καθώς υπάρχει ήδη η εφαρμογή σε Access. Ωστόσο, δεν είναι αυτοματοποιημένες οι διαδικασίες πλήρως, γεγονός που επιφέρει ενδεχόμενες χρονικές καθυστερήσεις, σφάλματα, δυσλειτουργίες και απασχόληση επιπλέον προσωπικού.

3.5.4 «Σύστημα Παρακολούθησης και Ελέγχου»

Τέλος, παρατηρείται ότι το υφιστάμενο σύστημα *παρακολούθησης και ελέγχου* δε λειτουργεί σε ικανοποιητικό βαθμό, ενώ κάποιες από τις διαδικασίες που εμφανίζονται στο μοντέλο, δεν υπάρχουν καθόλου στο υφιστάμενο, όπως για παράδειγμα ο έλεγχος του χρόνου διάρκειας κατάταξης σε κάθε κέντρο, ο ακριβής χρόνος που καταχωρείται κάθε νεοσυλλεκτος, αλλά και ο αριθμός των νεοσυλλέκτων που καταγράφονται σε κάθε κέντρο σε πραγματικό χρόνο, καθώς και στατιστικά στοιχεία, επίσης, σε πραγματικό χρόνο.

3.6 Φάση 6^η . Πραγμάτωση εφικτών και επιθυμητών αλλαγών

Οι συγκρίσεις της φάσης 5 θα αποφέρουν ένα σύνολο συστάσεων για αλλαγή οι οποίες θεωρούνται αποδεδειγμένα επιθυμητές. Όμως, είναι γεγονός ότι δεν θα είναι αποδεκτές όλες οι αλλαγές του συνόλου. (Βασιλακόπουλος, 2016)

Στη φάση 6 προσδιορίζονται οι δυνατές (εφικτές και επιθυμητές) αλλαγές στα πλαίσια μια διαδικασίας συμβιβασμού και συναίνεσης.

Αυτές οι αλλαγές προσδιορίζονται ως εξής: Ανάπτυξη πληροφοριακού συστήματος (ΠΣ), το οποίο μπορεί να συμπεριλάβει στοιχεία των υφιστάμενων διαδικασιών που σχετίζονται με την κατάταξη των νεοσυλλέκτων, έτσι ώστε να αποφευχθούν μη αυτοματοποιημένες και χρονοβόρες διαδικασίες, με επιδίωξη να μειωθεί παράλληλα και ένα μέρος του επιπλέον προσωπικού που απασχολείται στην κατάταξη. Τέλος, στο ίδιο ΠΣ είναι δυνατόν να

συμπεριληφθούν και οι δυνατότητες που περιγράφονται στο σύστημα παρακολούθησης και ελέγχου, από το οποίο επωφελείται ο ιδιοκτήτης του, η Διεύθυνση Οπλιτών Θητείας και ο Ελληνικός Στρατός, παρακολουθώντας τη διαδικασία της κατάταξης σε πραγματικό χρόνο και με μεγαλύτερη λεπτομέρεια, έτσι ώστε να είναι σε θέση να εξάγει ασφαλέστερα συμπεράσματα.

3.7 Φάση 7^η. Υλοποίηση των αλλαγών

Η φάση 7 αναφέρεται στη δράση που πρέπει να αναληφθεί για τη βελτίωση της προβληματικής κατάστασης. Η ανάληψη αυτής της δράσης προσδιορίζει ένα νέο πρόβλημα που θα μπορούσε να αντιμετωπισθεί με τη βοήθεια της ΜΜΣ: το πρόβλημα της πραγμάτωσης των προκριθέντων αλλαγών (*implementation*).

4. Συμπεράσματα – Προοπτικές

Στην παρούσα μελέτη, δόθηκε μια γενική περιγραφή των συστημικών εννοιών και εισήχθη η έννοια του συστήματος ανθρώπινης δραστηριότητας. Συγκεκριμένα, αναλύθηκε το Πληροφοριακό Σύστημα Κατάταξης Νεοσυλλέκτων (ΠΣΚΝ) με βάση τη Μεθοδολογία Μαλακών Συστημάτων (ΜΜΣ). Το σύνολο των δραστηριοτήτων (ή υποσυστημάτων) που αναπαριστούν το σύστημα που περιγράφεται στο θεμελιακό ορισμό αποτελείται από τις διαρκείς δραστηριότητες που πρέπει να υπάρχουν προκειμένου να πραγματοποιηθεί η συγκεκριμένη άποψη και σκοπός του συστήματος. Εν συνεχεία, αναλύθηκαν τα υποσυστήματα του συστήματος ΠΣΚΝ, έγινε η σύγκριση του ιδεατού με το υπάρχον μοντέλο και τελικώς καταλήξαμε στην πραγμάτωση των επιθυμητών και εφικτών αλλαγών.

5. Βιβλιογραφία

1. Checkland P. (1981), *Systems Thinking, Systems Practice*, John Wiley & Sons, New York.
2. Daves , Ledington. *Information in action: Soft Systems Methodology*.
3. Haperen KV. *Working towards Information Superiority: Superiority: Application Coherence for Digitisation Programmes – A Method for Coherently Defining Requirements for Future Command and Control Information Systems*. 2006.
4. Βασιλακόπουλος Γεώργιος, 2016
5. Basden A, Wood-Harper A.T. (2006), 'A Philosophical Discussion of the Root Definition in Soft Systems Thinking: An Enrichment of CATWOE'. *Systems Research and Behavioral Science*, no. 23, pp. 61 – 87.
6. Flood R. L. και Jackson M. C. (1996), *Δημιουργική Επίλυση Οργανωσιακών Προβλημάτων*. Εκδόσεις Παπαζήση, Μετάφραση: Κικής Τ., Επιμέλεια: Τσιβάκου Ι.
7. Williams M. C. (1999), 'Rich Pictures on the Path Towards Systemic Being', *Systems Research and Behavioral Science*, no. 16, pp. 369-373.
8. Κιουντούζης Ε. (2002), *Μεθοδολογίες Ανάλυσης και Σχεδιασμού Πληροφοριακών Συστημάτων*, Εκδόσεις Ε. Μπένου, Αθήνα.
9. Schoderbek, P.P., Schoderbek, C.G., Kefalas, A.G. (1990), *Management Systems: Conceptual Considerations*, Irwin, Boston, MA
10. Καλλιόπη Πεπόνη, Ανάλυση και σύγκριση συστημικών μεθοδολογιών οργανωσιακής σχεδίασης, διπλωματική εργασία, Πανεπιστήμιο Αιγαίου
11. Basden A, Wood-Harper A.T. (2006), 'A Philosophical Discussion of the Root Definition in Soft Systems Thinking: An Enrichment of CATWOE'. *Systems Research and Behavioral Science*, no. 23, pp. 61 – 87.
12. Armson R. (2001), 'A Quick Sprint through Soft Systems Methodology'.

Παράρτημα Β

Στο παρόν παράρτημα παρατίθεται ο αρχικός κύκλος ζωής που εκπονήθηκε για το Πληροφοριακό Σύστημα Κατάταξης Νεοσυλλέκτων στον Στρατό Ξηράς με βάση τη μεθοδολογία του «Καταρράκτη».

Σημειώνεται ότι οι ημερομηνίες που αναγράφονται παρακάτω δεν είναι πραγματικές, αλλά ενδεικτικές.

Περιεχόμενα

1. ΔΙΑΧΕΙΡΙΣΗ ΕΡΓΟΥ Β-2
2. ΔΙΕΡΕΥΝΗΤΙΚΗ ΜΕΛΕΤΗ (ΠΡΟΜΕΛΕΤΗ) Β-3
3. ΜΕΛΕΤΗ ΕΦΙΚΤΟΤΗΤΑΣ Β-4
4. ΑΝΑΛΥΣΗ ΑΠΑΙΤΗΣΕΩΝ Β-5
5. ΛΟΓΙΚΟΣ ΣΧΕΔΙΑΣΜΟΣ ΣΥΣΤΗΜΑΤΟΣ Β-6
6. ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΥΣΤΗΜΑΤΟΣ Β-7
7. ΠΡΑΓΜΑΤΩΣΗ ΣΥΣΤΗΜΑΤΟΣ Β-8
8. ΔΟΚΙΜΗ ΣΥΣΤΗΜΑΤΟΣ Β-9
9. ΧΡΟΝΟΔΙΑΓΡΑΜΜΑ ΕΚΤΕΛΕΣΗΣ ΕΡΓΟΥ Β-11

1. ΔΙΑΧΕΙΡΙΣΗ ΕΡΓΟΥ

Διάρκεια: 16 Εβδομάδες

Περίοδος: 1/9/2017- 6/1/2018

Συμμετέχοντες Φορείς: Αρμόδιοι Φορείς , Ομάδα Ανάπτυξης Πληροφοριακού Συστήματος

Περιγραφή: Εδώ περιλαμβάνονται όλες οι απαραίτητες ενέργειες διοικητικό-οικονομο-τεχνικού χαρακτήρα που θα διασφαλίσουν την ομαλή εξέλιξη των επιμέρους εργασιών του έργου. Κατά τη διάρκεια του έργου θα διεξάγονται ελεγκτικές διαδικασίες με τη συμμετοχή εκπροσώπων από τους διάφορους φορείς. Με βάση τις εκτιμήσεις των φορέων θα δίδονται οδηγίες προς την ομάδα έργου για τις απαραίτητες τεχνικές βελτιώσεις και θα επαναπροσδιορίζονται τα σχέδια υλοποίησης του έργου ανάλογα με τις ανάγκες του. Επιπλέον, η εργασία αυτή αφορά τον έλεγχο των ενδιάμεσων και των τελικών παραδοτέων του έργου σε αριθμό και ποιότητα. Στη *Διαχείριση του έργου* περιλαμβάνονται οι ακόλουθες επιμέρους εργασίες:

- Χάραξη στρατηγικής Έργου.
- Εντοπισμός και επίλυση προβλημάτων.
- Συντονισμός και έλεγχος ροής των εργασιών.
- Έλεγχος παραδοτέων

Παραδοτέο:

1.1 Εκθέσεις προόδου του έργου.

2. ΔΙΕΡΕΥΝΗΤΙΚΗ ΜΕΛΕΤΗ (ΠΡΟΜΕΛΕΤΗ)

Διάρκεια: 2 Εβδομάδες

Περίοδος: 1/9/2017- 15/9/2017

Συμμετέχοντες Φορείς: Ομάδα Ανάπτυξης Πληροφοριακού Συστήματος

Περιγραφή: Σκοπός είναι να προσδιοριστεί με σχετική ακρίβεια το πρόβλημα που επιθυμούμε να λύσουμε να εξεταστεί ποιες εναλλακτικές λύσεις θα μπορούσαν να αποτελέσουν λύση στο πρόβλημα και να καθορισθεί ποια από αυτές τις λύσεις φαίνεται να μας ενδιαφέρει περισσότερο.

Στη *Διερευνητική Μελέτη* περιλαμβάνονται οι ακόλουθες επιμέρους εργασίες:

- Εντοπισμός και ορισμός των προβληματικών καταστάσεων.
- Επίλυση προβλημάτων, συντονισμός και έλεγχος ροής των εργασιών.

Παραδοτέο:

2.1 Προμελέτη Έργου.

3. ΜΕΛΕΤΗ ΕΦΙΚΤΟΤΗΤΑΣ

Διάρκεια: 1 Εβδομάδα

Περίοδος: 16/9/2017- 24/9/2017

Συμμετέχοντες Φορείς: Ομάδα Ανάπτυξης Πληροφοριακού Συστήματος

Περιγραφή: Η *Μελέτη Εφικτότητας* έχει σκοπό να αξιολογήσει τις λύσεις που διατυπώθηκαν κατά τη διερευνητική μελέτη με κριτήρια:

- τον χρόνο, π.χ. πόσο χρόνο χρειάζεται η υλοποίηση κάθε λύσης
- την τεχνολογία, π.χ. αν είναι τεχνολογικά εφικτή η λύση
- την τεχνογνωσία, π.χ. αν υπάρχει τεχνογνωσία, αν θα υπάρξουν δυσκολίες
- το κόστος, π.χ. οικονομικό ή και κοινωνικό κόστος κάθε λύσης
- το όφελος, π.χ. ποια είναι τα μακροπρόθεσμα ή βραχυπρόθεσμα
- οικονομικά ή επιχειρησιακά οφέλη
- τον συνδυασμό των παραπάνω, π.χ. προσδιορισμός δεικτών για τη βαθμολόγηση κάθε λύσης.

Παραδοτέο:

3.1 Προσδιορισμός Λειτουργικής Εφικτότητας (Functional Feasibility).

3.2 Προσδιορισμός Τεχνικής Εφικτότητας (Technical Feasibility).

3.3 Προσδιορισμός Οικονομικής Εφικτότητας (Economic Feasibility).

4. ΑΝΑΛΥΣΗ ΑΠΑΙΤΗΣΕΩΝ

Διάρκεια: 1 Εβδομάδα

Περίοδος: 25/9/2017-30/9/2017

Συμμετέχοντες Φορείς: Ομάδα Ανάπτυξης Πληροφοριακού Συστήματος

Περιγραφή: Κατά τη διάρκεια της *Ανάλυσης των Απαιτήσεων* γίνεται λεπτομερής προσδιορισμός των λειτουργικών και τεχνικών απαιτήσεων, των απαιτήσεων ασφάλειας και των απαιτήσεων δεδομένων του συστήματος, των επιδόσεών του (π.χ. αριθμός ταυτόχρονων χρηστών, ανοχή παύσης λειτουργίας, ταχύτητας λειτουργίας, ποιότητας δεδομένων), των εισροών και εκροών δεδομένων και πληροφοριών, των επεξεργασιών και των αποτελεσμάτων.

Σ' αυτήν τη φάση γίνεται ανάλυση της διαδικασίας και προσδιορίζονται οι λειτουργικές και πληροφοριακές απαιτήσεις του συστήματος, καθώς και οι απαιτήσεις ασφάλειας πληροφοριών. Συνοπτικά, οι λειτουργίες (διεργασίες) του συστήματος, όπως αυτές θα σχεδιαστούν και υλοποιηθούν (προβλεπόμενη κατάσταση) είναι οι ακόλουθες:

1. Ο στρατολόγος εισαγάγει τα στοιχεία του νεοσυλλέκτου κατά την κατάταξη.
2. (Λοιπά προς μελέτη)

Παραδοτέο:

4.1 Προσδιορισμός λειτουργικών απαιτήσεων.

4.2 Προσδιορισμός τεχνικών απαιτήσεων.

4.3 Προσδιορισμός απαιτήσεων δεδομένων.

4.4 Προσδιορισμός απαιτήσεων ασφάλειας.

5. ΛΟΓΙΚΟΣ ΣΧΕΔΙΑΣΜΟΣ ΣΥΣΤΗΜΑΤΟΣ

Διάρκεια: 1 Εβδομάδα

Περίοδος: 1/10/2017-8/10/2017

Συμμετέχοντες Φορείς: Ομάδα Ανάπτυξης Πληροφοριακού Συστήματος

Περιγραφή: Κατά τη διάρκεια του *Λογικού Σχεδιασμού* θα πραγματοποιηθεί

- ♦ Η εννοιολογική αναπαράσταση των διεργασιών του υπό ανάπτυξη συστήματος χρησιμοποιώντας διάφορες τεχνικές για τις διάφορες όψεις του συστήματος
- ♦ Η εννοιολογική αναπαράσταση της βάσης δεδομένων

Επίσης, θα περιγραφούν τα υποσυστήματα (προς μελέτη ποια ακριβώς θα είναι αυτά) και θα περιγραφούν οι διεπαφές μεταξύ τους. Για το λογικό σχεδιασμό των διαδικασιών του συστήματος θα χρησιμοποιηθούν διαγράμματα ροής δεδομένων (data flow) και για τον λογικό σχεδιασμό των βάσεων δεδομένων θα χρησιμοποιηθούν διαγράμματα οντοτήτων-συσχετίσεων (entity-relationship). Επιπλέον, θα προσδιοριστούν τα δεδομένα που θα αποθηκεύονται στην κεντρική βάση δεδομένων και θα σχεδιαστεί η πολιτική ασφάλειας πληροφοριών του συστήματος. Στο *Λογικό Σχεδιασμό Συστήματος* περιλαμβάνονται οι ακόλουθες επιμέρους εργασίες:

- Λογικός σχεδιασμός διαδικασιών.
- Λογικός σχεδιασμός βάσεων δεδομένων.

Παραδοτέο:

5.1 Λογικός σχεδιασμός συστήματος.

6. ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΥΣΤΗΜΑΤΟΣ

Διάρκεια: 1 Εβδομάδα

Περίοδος: 9/10/2017-15/10/2017

Συμμετέχοντες Φορείς: Ομάδα Ανάπτυξης Πληροφοριακού Συστήματος

Περιγραφή: Κατά τη διάρκεια της φάσης της *Αρχιτεκτονικής Συστήματος* θα περιγραφεί η αρχιτεκτονική του συστήματος. Το σύστημα αποτελείται από μία κύρια διαδικασία: της κατάταξης του νεοσυλλέκτου από τον στρατολόγο. -Το σύστημα θα υλοποιηθεί σε περιβάλλον Internet/Web. -Κάθε χρήστης (π.χ. στρατολόγος) συνδέεται μέσω του δικτύου (π.χ. μέσω Ίντερνετ) με τον Web Server του ΚΕΠΥΕΣ και ακολουθεί η διαδικασία αυθεντικοποίησης αυτού και του νεοσυλλέκτου. Η πρώτη γίνεται μέσω της πληκτρολόγησης USERNAME και PASSWORD και η δεύτερη μέσω της ταυτοποίησης του αριθμού μητρώου του νεοσυλλέκτου. Σε περίπτωση επιτυχούς αυθεντικοποίησης του χρήστη, του παρέχεται πρόσβαση στα σχετικά στοιχεία του νεοσυλλέκτου. Οι ενέργειες που εκτελεί ο κάθε χρήστης του συστήματος (π.χ. συμπλήρωση στοιχείων στο σύστημα) πραγματοποιούνται μέσω Web σελίδων που φορτώνονται στον Web Browser του υπολογιστή του. Στην *Αρχιτεκτονική Συστήματος* περιλαμβάνονται οι ακόλουθες επιμέρους εργασίες:

- Προσδιορισμός διαδικασιών.
- Προσδιορισμός υποσυστημάτων.
- Προσδιορισμός ασφάλειας συστήματος

Παραδοτέο:

6.1 Αρχιτεκτονική συστήματος.

7. ΠΡΑΓΜΑΤΩΣΗ ΣΥΣΤΗΜΑΤΟΣ

Διάρκεια: 8 Εβδομάδες

Περίοδος: 16/10/2017-17/12/2017

Συμμετέχοντες Φορείς: Ομάδα Ανάπτυξης Πληροφοριακού Συστήματος

Περιγραφή: Στη *Πραγμάτωση του Συστήματος* θα γίνει η πλήρης ανάπτυξη (codification) και ολοκλήρωσή του μαζί με τις αντίστοιχες δοκιμές σε επίπεδο μονάδας και συστήματος και θα υλοποιηθούν οι διαδικασίες ασφάλειας πληροφοριών του ολοκληρωμένου συστήματος.

Παραδοτέο:

7.1 Ολοκληρωμένο σύστημα διαχείρισης ψηφιακού βιβλιαρίου ασθενών.

8. ΔΟΚΙΜΗ ΣΥΣΤΗΜΑΤΟΣ

Διάρκεια: 2 Εβδομάδα

Περίοδος: 18/12/2017-6/1/2018

Συμμετέχοντες Φορείς: Ομάδα Ανάπτυξης Πληροφοριακού Συστήματος

Περιγραφή: Σκοπός της *Δοκιμής του Συστήματος* είναι να γίνουν οι αναγκαίες τροποποιήσεις για την εύρυθμη λειτουργία του συστήματος και την ικανοποίηση των απαιτήσεων, όπως ο εντοπισμός των αποκλίσεων του υλοποιημένου προϊόντος από τις προδιαγραφές, ο εντοπισμός σφαλμάτων (αλγοριθμικά σφάλματα, σφάλματα υπερφόρτωσης του συστήματος, σφάλματα απόδοσης (π.χ. η απόκριση του συστήματος), σφάλματα ορίων).

Η διαδικασία θα γίνεται παράλληλα με την ανάπτυξη τουλάχιστον για τις επιμέρους μονάδες του συστήματος.

-Για τον σκοπό αυτόν θα δοκιμασθεί η επικοινωνία και ροή πληροφοριών μεταξύ των υποσυστημάτων, καθώς και το ολοκληρωμένο σύστημα με εικονικά σενάρια ελέγχου, έτσι ώστε να υπάρχει μια ολοκληρωμένη εικόνα λειτουργίας σταθερότητας του συστήματος. Μετά τη σύμφωνη γνώμη και την αποδοχή του συστήματος από τον Εργοδότη θα γίνει η παράδοση και η εγκατάστασή του. Στη *Δοκιμή Συστήματος* περιλαμβάνονται οι ακόλουθες επιμέρους εργασίες:

- Προετοιμασία δοκιμαστικής λειτουργίας του συστήματος.
- Black Box – White Box.
- Unit Tests.
- Integration Tests
- Δοκιμαστική λειτουργία συστήματος.

Παραδοτέο:

8.1 Τελική έκδοση συστήματος.

9. ΧΡΟΝΟΔΙΑΓΡΑΜΜΑ ΕΚΤΕΛΕΣΗΣ ΕΡΓΟΥ

ΠΑΚΕΤΟ ΕΡΓΑΣΙΑΣ	ΠΑΡΑΔΟΤΕΑ	ΕΒΔΟΜΑΔΕΣ																ΕΝΑΡΞΗ	ΛΗΞΗ	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16			
1. Διαχείριση Έργου	Εκθέσεις προόδου	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	01/9/2017	6/01/2018
2. Διερευνητική Μελέτη	Προμελέτη Έργου	■	■															01/9/2017	15/9/2017	
3. Μελέτη Εφικτότητας	Προσδιορισμός σκοπιμότητας			■														16/9/2017	24/9/2017	
4. Ανάλυση Απαιτήσεων	Προσδιορισμός απαιτήσεων.				■													25/9/2017	30/9/2017	
5. Λογικός Σχεδιασμός	Λογικός σχεδιασμός συστήματος					■												1/10/2017	8/10/2017	
6. Αρχιτεκτονική Συστήματος	Αρχιτεκτονική συστήματος.						■											9/10/2017	15/10/2017	
7. Πραγμάτωση Συστήματος	Ολοκληρωμένο σύστημα							■	■	■	■	■	■	■	■	■		16/10/2017	17/12/2017	
8. Δοκιμή Συστήματος	Τελική έκδοση συστήματος																■	18/12/2017	6/01/2018	

Παράρτημα Γ

Στο παρόν παράρτημα παρουσιάζονται εν συντομία οι τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη του Πληροφοριακού Συστήματος Κατάταξης Νεοσυλλέκτων του Στρατού Ξηράς, καθώς και μία περιορισμένη έκταση παρουσίαση του.

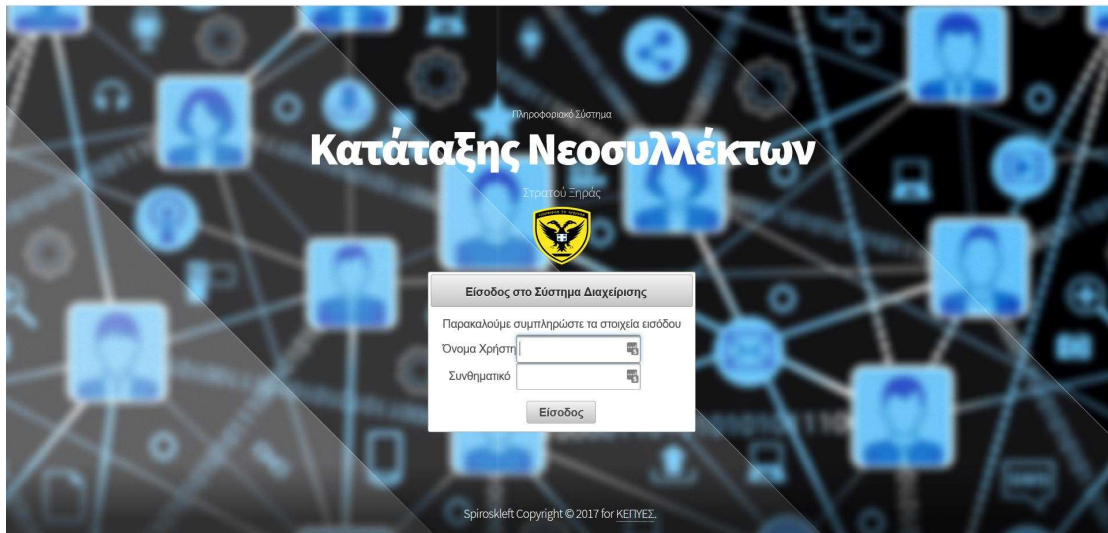
Τεχνολογίες Ανάπτυξης Πληροφοριακού Συστήματος

Η ανάπτυξη της εφαρμογής πραγματοποιήθηκε με χρήση της γλώσσας προγραμματισμού Java και πιο συγκεκριμένα των πλατφορμών (frameworks) διαδικτυακού προγραμματισμού JSF, JPA και Primefaces, χρησιμοποιώντας την MySQL ως σύστημα διαχείρισης βάσεων δεδομένων. Η επικοινωνία του προγραμματιστικού επιπέδου με τη βάση δεδομένων έχει γίνει μέσω του framework Hibernate, το οποίο είναι μια βιβλιοθήκη απεικόνισης αντικειμενοστραφούς μοντέλου στα σχεσιακά δεδομένα της βάσης δεδομένων. Η εφαρμογή που έχει αναπτυχθεί είναι διαδικτυακή, αυτό σημαίνει ότι κάθε χρήστης της υπηρεσίας έχει δυνατότητα πρόσβασης στην εφαρμογή μέσω του φυλλομετρητή (browser) του υπολογιστή. Ως εξυπηρετητής εφαρμογής (Application Server) έχει χρησιμοποιηθεί ο Wildfly-10.1.0.Final.

Στην συνέχεια ακολουθεί μία σύντομη παρουσίαση του πληροφοριακού συστήματος που αναπτύχθηκε με τις κύριες οθόνες και λειτουργίες του. Υπενθυμίζεται ότι δεν αποτελεί την τελική έκδοση του πληροφοριακού συστήματος, καθώς και ότι δεν παρουσιάζονται όλες οι λειτουργίες του παρά μόνο ένα μικρό μέρος αυτών.

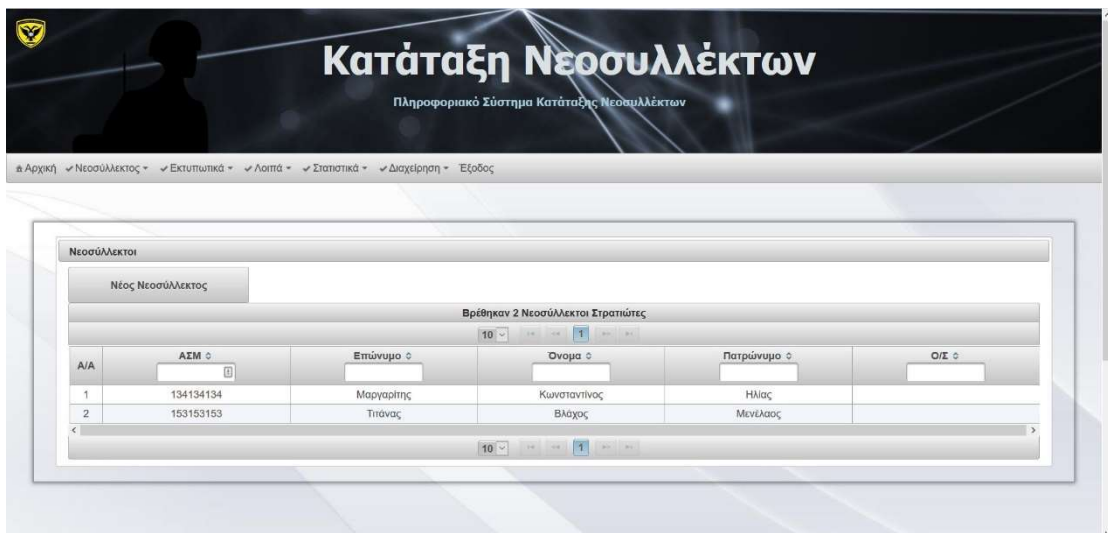
Είσοδος στην Εφαρμογή

Πληκτρολογώντας σε έναν φυλλομετρητή (browser) τη διεύθυνση της εφαρμογής εμφανίζεται η παρακάτω οθόνη εισόδου (login) στην οποία ο χρήστης πληκτρολογεί τα διαπιστευτήρια που του έχουν δοθεί προκειμένου να εισέλθει στο πληροφοριακό σύστημα.



Σελίδα Αναζήτησης Νεοσυλλέκτου

Στη σελίδα που ακολουθεί αυτή της εισόδου, ο χρήστης μπορεί να αναζητήσει, με βάση όποιο από τα κύρια χαρακτηριστικά επιθυμεί, τον νεοσύλλεκτο που πρόκειται να καταχωρήσει στη διαδικασία της κατάταξης. Επίσης, μέσω του μενού στο πάνω μέρος της οθόνης μπορεί να μεταφερθεί σε διάφορες λειτουργίες που παρέχονται από το πληροφοριακό σύστημα.



Προσωπικά Στοιχεία

Με την επιλογή του νεοσυλλέκτου ο χρήστης μεταφέρεται στην πρώτη σελίδα καταχώρησης στοιχείων που αφορά τα προσωπικά στοιχεία του νεοσυλλέκτου. Ενδεικτικά κάποια εξ αυτών φαίνονται στις παρακάτω εικόνες.

Κατάταξη Νεοσυλλέκτων
Πληροφοριακό Σύστημα Κατάταξης Νεοσυλλέκτων

Αρχική | Νεοσυλλέκτος | Εκτυπωτικά | Λοιπά | Στατιστικά | Διαχείριση | Εξόδος

Προσωπικά | Οικογενειακή Κατάσταση | Μοριοδότηση | Υγεία | Γνώσεις | Κατάταξη | Επιλογή άλλου Ν/Σ

Προσωπικά Στοιχεία Νεοσυλλέκτου

Βασικά Στοιχεία

* Επώνυμο Πιάνας * Όνομα Βλάχος * Πατρώνυμο Μενέλαος * Μητρώνυμο Ασπασία
 * ΑΣΜ 163153153 * ΕΞΣΟ 21073 * Ο/Σ Ερευνας Πληροφορικής * Μονάδα Κατάταξης ΚΕΤΘ

Προσωπικά Στοιχεία

Καταγωγή
 * Νομός - Επιλέξτε - * Επαρχία - Επιλέξτε - * Πόλη - Επιλέξτε -

Διαμονή
 * Νομός - Επιλέξτε - * Επαρχία - Επιλέξτε - * Πόλη - Επιλέξτε -

Λοιπά

* Αστυνομικό Τμήμα Α.Τ. Αιμυρού Διεύθυνση Μάνδρας * Τ.Κ. 37900
 Τηλέφωνο Σταθερό 210655655 Τηλέφωνο Κινητό 6936025321 ΑΦΜ 134178999
 ΔΟΥ Αθηνών ΑΜΚΑ IBAN

Στοιχεία Ταυτότητας

* Τόπος Γέννησης Λάρισα * Δημότης Λάρισα * Αριθμός Αστ. Δελτίου Ταυτότητας 12345
 * Σχήμα Προσώπου Δεν κατανοείται? * Χρώμα Μαλλιών Δεν κατανοείται? * Χρώμα Μπαγών Δεν κατανοείται?
 Ονοματεπώνυμο Συζύγου

* Υποχρεωτικά Πεδία

Αποθήκευση

Οικογενειακή Κατάσταση

Η επόμενη σελίδα αφορά την οικογενειακή κατάσταση του νεοσυλλέκτου και καταχωρούνται διάφορα στοιχεία που είναι χρήσιμα στις στρατιωτικές υπηρεσίες. Ενδεικτικά κάποια εξ αυτών φαίνονται στην παρακάτω εικόνα.

Μοριοδότηση

Στην επόμενη σελίδα καταχωρούνται τα στοιχεία εκείνα που αφορούν τη Μοριοδότηση του κάθε νεοσυλλέκτου βάση των νόμων του ελληνικού κράτους. Ενδεικτικά φαίνονται κάποια εξ αυτών στην παρακάτω εικόνα.

Υγεία

Εδώ καταχωρούνται όλα τα στοιχεία εκείνα που έχουν να κάνουν με την υγεία κάθε νεοσυλλέκτου. Μέσα από τη σελίδα αυτή ο χρήστης μπορεί επίσης να αλλάξει και τα στοιχεία Σωματικής Ικανότητας του νεοσυλλέκτου, καθώς και να καταχωρήσει διάφορα στοιχεία προς χρήση από τις υπηρεσίες του στρατού.

Υγειονομική Κατάσταση Νεοσυλλέκτου

Βασικά Στοιχεία

Επίσημο: Τρίπιας Όνομα: Βλάχος Πατρώνυμο: Μενίλαος Μητρώνυμο: Ασπασία
 ΑΣΜ: 153153153 ΕΣΣΟ: 21073 Ο/Σ: Έρευνας Πληροφορικής Μονάδα Κατάταξης: ΚΕΤΟ

Υγειονομικά Στοιχεία

Σωματομετρικά Στοιχεία
 * Ύψος: 192 * Βάρος: 68

Στοιχεία Υγείας

* ΣΙ: H1 Ομάδα Αίματος: AB * Ακοή: Καλή * Όραση: Καλή όραση

Πρόβλημα Οσμίας: Καλή Οσμία * Ψυχολογικές Παθήσεις: Όχι * Νευρωτικές Παθήσεις: Ναι * Αχρωματωσία: Ναι
 * Διασχιματωσία: Όχι * Μεταδοτικές Λοιμώξεις: Ναι Χειρουργικές Επεμβάσεις: Όχι Επέμβαση Ανακτικής Καρδιάς: Ναι
 Βρογχικό Άσθμα: Όχι Επιληψία: Ναι Διόστρες: Ναι Τρυπημένα Αιμά: Ναι
 Ακατάλληλος για Μάγνητος: Ναι Ακατάλληλος για Ο/Α: Ναι Ακατάλληλος για Αεροπορίας: Ναι Κίρσοι: Ναι
 Πλατυποδία: Ναι Κατόρθια: Ναι Κακή Κατηφόρα Ακρων: Ναι Δερμοστίβια: Ναι
 Παχυσαρκία: Όχι ΛΟΙΦ: Ναι

Αλλαγή Σωματικής Ικανότητας Νεοσυλλέκτου: [Αλλαγή Σ.Ι.](#)

Αποθήκευση * Υποχρεωτικά Πεδία

Υγειονομική Κατάσταση Νεοσυλλέκτου

Βασικά Στοιχεία

Επίσημο: Τρίπιας Όνομα: Βλάχος Πατρώνυμο: Μενίλαος Μητρώνυμο: Ασπασία
 ΑΣΜ: 153153153 ΕΣΣΟ: 21073 Ο/Σ: Έρευνας Πληροφορικής Μονάδα Κατάταξης: ΚΕΤΟ

Υγειονομικά Στοιχεία

Σωματομετρικά Στοιχεία
 * Ύψος: 192

Στοιχεία Υγείας

* ΣΙ: H1

Πρόβλημα Οσμίας: Καλή Οσμία * Ψυχολογικές Παθήσεις: Όχι * Νευρωτικές Παθήσεις: Ναι * Αχρωματωσία: Ναι
 * Διασχιματωσία: Όχι * Μεταδοτικές Λοιμώξεις: Ναι Χειρουργικές Επεμβάσεις: Όχι Επέμβαση Ανακτικής Καρδιάς: Ναι
 Βρογχικό Άσθμα: Όχι Επιληψία: Ναι Διόστρες: Ναι Τρυπημένα Αιμά: Ναι
 Ακατάλληλος για Μάγνητος: Ναι Ακατάλληλος για Ο/Α: Ναι Ακατάλληλος για Αεροπορίας: Ναι Κίρσοι: Ναι
 Πλατυποδία: Ναι Κατόρθια: Ναι Κακή Κατηφόρα Ακρων: Ναι Δερμοστίβια: Ναι
 Παχυσαρκία: Όχι ΛΟΙΦ: Ναι

Αλλαγή Σωματικής Ικανότητας Νεοσυλλέκτου: [Αλλαγή Σ.Ι.](#)

Αποθήκευση * Υποχρεωτικά Πεδία

Μεταβολή Σωματικής Ικανότητας Νεοσυλλέκτου

Βασικά Στοιχεία

Επίσημο: Τρίπιας Όνομα: Βλάχος Πατρώνυμο: Μενίλαος Μητρώνυμο: Ασπασία
 ΑΣΜ: 153153153 ΕΣΣΟ: 21073 Ο/Σ: Έρευνας Πληροφορικής Μονάδα Κατάταξης: ΚΕΤΟ

Στοιχεία Σωματικής Ικανότητας

* Σωματική Ικανότητα: H1 * Ημερομηνία Ικανότητας: 15/12/2017 Σχετική Διαταγή: Φ 133/123/126
 Αριθμός Γνωμάτευσης: 1 * Επιτροπή Απολογιών: 401 ΓΣΝΑ * Αιτιολογία: ΠΑ Παράγοντας
 Γνωμάτευση Ιατρού:

Καταχώρηση

Γνώσεις

Στην παρούσα σελίδα καταχωρούνται στοιχεία που αφορούν το μορφωτικό επίπεδο του υπόψιν νεοσυλλέκτου, καθώς και διάφορα άλλα γνωστικά στοιχεία.

Μορφωτικό Επίπεδο

A/A	Σχολή	Χώρα	Σχολές - Χώρες Επίπεδο Μορφώσεως	Δείκτης Μορφώσεως
1	Πανεπιστήμιο Πελαγονίας	Ελλάδα	Μεταπτυχιακό	Πτυχιούχος Ανώτατης Σχολής
2	Πανεπιστήμιο Αθηνών	Ελλάδα	Μεταπτυχιακό	Πτυχιούχος Ανώτατης Σχολής

[+ Προσθήκη Σχολής](#)

Ξένες Γλώσσες

A/A	Ξένη Γλώσσα	Πτυχίο	Επίδοση Πτυχίου
1	Αγγλικά	Proficiency	Επίδοση Βαθμός
2	Γερμανικά	Proficiency	Χρυσά

[+ Προσθήκη Ξένης Γλώσσας](#)

Επαγγελματικά Στοιχεία

* Επάγγελμα: Ιδιωτικός Υπάλληλος Χρόνος Εξασκήσεως Επαγγέλματος: 24 Διαθέτει Άδεια Εξασκήσεως Επαγγέλματος: Ημερομηνία Κτήσης Άδειας Εξασκήσεως Επαγγέλματος: 08/11/2017

Λοιπά Στοιχεία

* Δίπλωμα Οδήγησης: Κατηγορία Β Ημερομηνία Κτήσης Διπλώματος Οδήγησης: 08/11/2016 * Επιβεβαιώνω την γνωστοποίηση των σπουδαστικών και επαγγελματικών του προσόντων στην ηλεκτρονική σελίδα του ΥΠΕΘΑ στο Internet

Αποθήκευση * Υποχρεωτικά Πεδία

Στοιχεία Κατάταξης

Στην παρούσα σελίδα καταχωρούνται στοιχεία που αφορούν την κατάταξη του νεοσυλλέκτου, όπως διάφορες επιθυμίες για ειδικότητα, μεταθέσεις κτλ. Κάποιες εξ αυτών φαίνονται ενδεικτικά στην παρακάτω εικόνα.

Κατάταξη

Στοιχεία Κατάταξης

* Ημερομηνία Κατάταξης: 07/12/2017 * Μονάδα (Τάγμα): ΚΕΤΘ * Κατηγορία Εφέδρου: Κατηγορία Εφέδρου Προτεραιότητα: 186

Ειδικότητα: Τεχνικός Βαθμός: Στρατιωτικής Ημέρες Φυλακής > 00

Επιθυμίες Νεοσυλλέκτου

Επιθυμεί ΟΗΕ: Επιθυμεί ΕΛΔΥΚ: Επιθυμεί ΥΕΑ: Επιθυμεί Στρατιωτική:

Επιθυμεί Ειδικές Δυναμίες: Επιθυμεί ΤΕΝΕ:

Επιθυμίες Τοποθέτησης

Σειρά	Επαρχία
1	Μαγνησία
2	Κέρκυρα
3	Έβρος

Επιθυμίες Μετάθεσης

Σειρά	Πόλη
1	Αθήνα
2	Βόλος
3	Θεσσαλονίκη

Ανασκόπηση Στοιχείων

Στην παρούσα σελίδα ο χρήστης έχει τη δυνατότητα ανασκόπησης όλων των στοιχείων του νεοσυλλέκτου που καταχωρήθηκαν. Στη συνέχεια μπορεί να επιλέξει τον επόμενο προς κατάταξη νεοσύλλεκτο.

Ανασκόπηση

Ανασκόπηση Στοιχείων Νεοσυλλέκτου

Επίσημο: Τίνας Όνομα: Βλάχος ΑΔΜ: 153153153

Για Υποστήριξη επικοινωνήστε μαζί μας
Σταθός Κλεψιδίου, 16613 Π. Γραφείο Κλεψιδίου
ΚΕΤΘ/ΕΛ. Στοιχεία © 2012. All Rights Reserved

Σημαντική Σημείωση

Είναι σημαντικό να αναφερθεί ότι οι παραπάνω εικόνες δεν αποτελούν την τελική παρά μόνο μία αρχική έκδοση του πληροφοριακού συστήματος, το οποίο αναπτύχθηκε στα πλαίσια της παρούσης διπλωματικής εργασίας με τη μεθοδολογία Scrum και αφορά το χρονικό διάστημα και μόνο της διπλωματικής εργασίας. Τέλος, σημειώνεται ότι ο παραπάνω νεοσύλλεκτος αποτελεί φανταστικό πρόσωπο και τα στοιχεία αυτού είναι, επίσης, φανταστικά.