



Πανεπιστήμιο Πειραιώς

Τμήμα Ψηφιακών Συστημάτων

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΔΙΔΑΚΤΙΚΗ ΤΗΣ ΤΕΧΝΟΛΟΓΙΑΣ ΚΑΙ ΨΗΦΙΑΚΑ ΣΥΣΤΗΜΑΤΑ

Κατεύθυνση: Δικτυοκεντρικά Συστήματα

ΜΕΘΟΔΟΛΟΓΙΑ SCRUM ΓΙΑ ΑΝΑΠΤΥΞΗ ΣΥΣΤΗΜΑΤΩΝ

Κακοσίμος Σπυρίδων, ΑΜ: 10083

Μεταπτυχιακή Διπλωματική Εργασία

Επιβλέπων: Ανδριάννα Πρέντζα, Επίκουρη Καθηγήτρια

Πειραιάς 2013



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΤΜΗΜΑ: ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
Π.Μ.Σ. ΔΙΚΤΥΟΚΕΝΤΡΙΚΑ ΣΥΣΤΗΜΑΤΑ

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ' ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή της για μη κερδοσκοπικό σκοπό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα.

(Υπογραφή)

.....

ΣΠΥΡΙΔΩΝ Ε. ΚΑΚΟΣΙΜΟΣ

Ευχαριστίες

Καταρχήν θα ήθελα να ευχαριστήσω θερμά όλους όσους συνέβαλαν με οποιονδήποτε τρόπο στην επιτυχή εκπόνηση της διπλωματικής μου εργασίας. Θα πρέπει να ευχαριστήσω ιδιαίτερα την καθηγήτρια κα. Ανδριάννα Πρέντζα για την επίβλεψη της εργασίας. Ήταν πάντοτε διαθέσιμη να προσφέρει τη γνώση και την εμπειρία της για τη βαθύτερη κατανόηση της Scrum μεθοδολογίας και οι συμβουλές όπως επίσης και η καθοδήγησή της ήταν καθοριστικές για την επιτυχή ολοκλήρωση της διπλωματικής μου εργασίας.

Στη συνέχεια θα ήθελα να ευχαριστήσω όλους τους καθηγητές του Π.Μ.Σ. Δικτυοκεντρικών Συστημάτων που με καθοδήγησαν τα τελευταία δύο χρόνια και με τις γνώσεις τους και το ενδιαφέρον τους ενίσχυσαν τη βαθύτερη γνώση και την ολοκλήρωση των μεταπτυχιακών μου σπουδών.

Τέλος, θα ήθελα να αναφέρω ανθρώπους εκτός του ακαδημαϊκού περιβάλλοντος, που με στήριξαν ιδιαίτερα στην προσπάθεια απόκτησης του μεταπτυχιακού μου τίτλου. Ένα μεγάλο ευχαριστώ λοιπόν στην Αλεξούλα, στους γονείς μου και την αδερφή μου που πίστεψαν στις δυνατότητές μου και με την ψυχολογική τους κυρίως υποστήριξη με οδήγησαν στην εκπλήρωση των στόχων μου.

Περίληψη

Παρόλο που στις ημέρες μας υπάρχουν αρκετές διαφορετικές προσεγγίσεις για την ανάπτυξη λογισμικού, μπορούμε με βεβαιότητα να ισχυριστούμε πως μόνο ορισμένες από αυτές μπορούν να χρησιμοποιηθούν με επιτυχία στην ανάπτυξη Πληροφοριακών Συστημάτων. Αυτό που είναι εμφανές είναι ότι οι μεθοδολογίες που έχουν καταφέρει να επιβιώσουν και να χρησιμοποιηθούν ευρέως είναι οι ευέλικτες μεθοδολογίες (agile methodologies), μιας και κατάφεραν να ενσωματώσουν εγγυημένες πρακτικές που ενισχύουν την ανάπτυξη ποιοτικού λογισμικού.

Μια από τις ευέλικτες μεθοδολογίες είναι και η Scrum στην οποία αναφερόμαστε και αναλύουμε εκτενώς στα κεφάλαια που ακολουθούν. Κύριο χαρακτηριστικό της αποτελεί το γεγονός ότι βασίζεται κατά κύριο λόγο στην ομαδική εργασία και εγγυάται ποιοτικότερο λογισμικό εντός συγκεκριμένου χρονικού διαστήματος. Όπως και οι υπόλοιπες agile methodologies έτσι και η Scrum είναι προσανατολισμένη στους ανθρώπους και όχι στις διαδικασίες.

Οι παραδοσιακές μεθοδολογίες σχεδιάστηκαν για να ανταποκρίνονται στη μη προβλεψιμότητα του εξωτερικού περιβάλλοντος στο ξεκίνημα κάθε κύκλου εξέλιξης. Προσεγγίσεις όπως η ελικοειδής μεθοδολογία και οι παρόμοιές της περιορίζουν την ικανότητά τους να ανταποκρίνονται στις αλλαγές των απαιτήσεων όταν το πληροφοριακό σύστημα έχει ξεκινήσει να υλοποιείται. Αντιθέτως, η μεθοδολογία Scrum έχει σχεδιαστεί έτσι ώστε να παρέχει ευελιξία σε όλη τη διάρκεια του κύκλου ανάπτυξης ενός πληροφοριακού συστήματος. Χρησιμοποιεί ένα σύνολο δοκιμασμένων πρακτικών που υπόσχονται την ολοκλήρωση του έργου μέσα σε καθορισμένα χρονικά όρια. Επίσης παρέχει πρακτικές για το σχεδιασμό μιας νέας έκδοσης του προϊόντος και στη συνέχεια διαχειρίζεται τις πρακτικές αυτές καθώς η εφαρμογή εξελίσσεται.

Σκοπός της συγκεκριμένης διπλωματικής εργασίας είναι η λεπτομερής μελέτη και αξιολόγηση της μεθοδολογίας Scrum καθώς επίσης και η ανάλυση των πρακτικών

που χρησιμοποιεί. Αρχικά παρουσιάζονται οι κυριότερες παραδοσιακές προσεγγίσεις και στη συνέχεια εμβαθύνουμε στην ευέλικτη ανάπτυξη λογισμικού με τη χρήση της Scrum μεθοδολογίας. Στο τελευταίο κεφάλαιο της διπλωματικής εργασίας χρησιμοποιούμε τη μεθοδολογία Scrum για να δείξουμε πώς θα μπορούσε να υλοποιηθεί ένα Πληροφοριακό Σύστημα για την ανάπτυξη του οποίου χρησιμοποιήθηκε η προσέγγιση του eXtreme Programming.

Λέξεις κλειδιά: Scrum Master, Scrum Ομάδα, Sprint, συνάντηση σχεδιασμού sprint (sprint planning meeting), ανεκτέλεστο υπόλοιπο (backlog), καθημερινές Scrum συναντήσεις (Daily Scrums), απολογισμός sprint (sprint review).

ΠΕΡΙΕΧΟΜΕΝΑ

1	ΕΙΣΑΓΩΓΗ	11
1.1	ΟΡΙΣΜΟΣ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ.....	11
1.2	ΔΟΜΗ ΤΗΣ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ	11
1.3	ΣΥΝΕΙΣΦΟΡΑ ΤΗΣ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ	13
2	ΟΙ ΠΑΡΑΔΟΣΙΑΚΕΣ ΠΡΟΣΕΓΓΙΣΕΙΣ ΣΤΗΝ ΑΝΑΠΤΥΞΗ ΣΥΣΤΗΜΑΤΩΝ.....	14
2.1	ΜΕΘΟΔΟΛΟΓΙΕΣ ΑΝΑΠΤΥΞΗΣ ΣΥΣΤΗΜΑΤΩΝ	16
2.2	ΕΥΕΛΙΚΤΕΣ ΠΡΟΣΕΓΓΙΣΕΙΣ ΣΤΗΝ ΑΝΑΠΤΥΞΗ ΣΥΣΤΗΜΑΤΩΝ.....	20
2.3	ΟΙ ΕΥΕΛΙΚΤΕΣ ΜΕΘΟΔΟΛΟΓΙΕΣ ΜΕ ΜΙΑ ΔΕΥΤΕΡΗ ΜΑΤΙΑ	24
2.3.1	Η ΜΕΘΟΔΟΛΟΓΙΑ RATIONAL UNIFIED PROCESS (RUP)	27
2.3.2	Η ΜΕΘΟΔΟΛΟΓΙΑ EXTREME PROGRAMMING.....	31
2.4	ΠΕΡΙΠΤΩΣΕΙΣ ΧΡΗΣΗΣ (USE CASES) ΚΑΙ ΑΠΑΙΤΗΣΕΙΣ ΧΡΗΣΤΩΝ.....	32
2.5	ΓΙΑΤΙ ΠΡΕΠΕΙ ΝΑ ΠΡΟΤΙΜΟΥΝΤΑΙ ΟΙ ΕΥΕΛΙΚΤΕΣ ΜΕΘΟΔΟΛΟΓΙΕΣ	34
3	ΕΙΣΑΓΩΓΗ ΣΤΗ SCRUM	37
3.1	ΙΣΤΟΡΙΚΗ ΕΞΕΛΙΞΗ ΤΗΣ SCRUM – ΕΤΑΙΡΙΕΣ ΠΟΥ ΒΑΣΙΣΤΗΚΑΝ ΠΑΝΩ ΤΗΣ.....	40
3.2	ΚΥΡΙΕΣ ΦΑΣΕΙΣ ΤΗΣ ΜΕΘΟΔΟΛΟΓΙΑΣ.....	42
3.3	ΚΥΡΙΕΣ ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ ΤΗΣ SCRUM	44
3.3.1	ΚΑΘΟΡΙΣΜΟΣ ΤΟΥ ΜΕΓΕΘΟΥΣ ΤΟΥ SPRINT.....	48
3.3.2	ΚΑΘΟΡΙΣΜΟΣ ΤΟΥ ΣΤΟΧΟΥ ΤΟΥ SPRINT.....	49
3.4	ΚΥΡΙΑ ΑΝΤΙΚΕΙΜΕΝΑ ΕΝΟΣ SCRUM PROJECT	50
3.5	ΜΗΧΑΝΙΣΜΟΙ ΕΛΕΓΧΟΥ ΣΤΗ ΜΕΘΟΔΟΛΟΓΙΑ SCRUM	51
4	ΓΕΝΙΚΟ ΠΛΑΙΣΙΟ - ΡΟΛΟΙ ΤΗΣ SCRUM	53
4.1	Ο ΙΔΙΟΚΤΗΤΗΣ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ	56
4.2	Ο SCRUM MASTER	57
4.3	Η ΟΜΑΔΑ SCRUM	60
4.3.1	ΧΕΙΡΙΣΜΟΣ ΠΟΛΛΑΠΛΩΝ ΟΜΑΔΩΝ ΣΕ ΕΝΑ SCRUM PROJECT.....	62
4.3.2	ΠΙΝΑΚΕΣ ΚΑΘΗΚΟΝΤΩΝ (TASK BOARDS)	68
5	ΟΙ ΠΡΑΚΤΙΚΕΣ ΤΗΣ SCRUM ΜΕΘΟΔΟΛΟΓΙΑΣ.....	70
5.1	ΑΝΕΚΤΕΛΕΣΤΟ ΥΠΟΛΟΙΠΟ ΤΟΥ ΠΡΟΪΟΝΤΟΣ (PRODUCT BACKLOG).....	70
5.1.1	ΚΥΡΙΑ ΠΕΔΙΑ ΤΗΣ ΛΙΣΤΑΣ ΤΟΥ ΑΝΕΚΤΕΛΕΣΤΟΥ ΥΠΟΛΟΙΠΟΥ ΣΥΣΤΗΜΑΤΟΣ	73
5.2	ΣΥΝΑΝΤΗΣΗ ΣΧΕΔΙΑΣΗΣ SPRINT.....	75
5.2.1	VELOCITY: ΜΙΑ ΧΡΗΣΙΜΗ ΠΑΡΑΜΕΤΡΟΣ ΓΙΑ ΑΚΡΙΒΕΙΣ ΥΠΟΛΟΓΙΣΜΟΥΣ.....	79
5.2.2	ΛΙΣΤΕΣ ΕΡΓΑΣΙΩΝ	82
5.2.3	ΠΡΟΤΕΡΑΙΟΤΗΤΕΣ ΣΤΗ ΣΥΝΑΝΤΗΣΗ ΣΧΕΔΙΑΣΜΟΥ	82
5.2.4	ΚΕΝΑ ΔΙΑΣΤΗΜΑΤΑ ΜΕΤΑΞΥ ΤΩΝ SPRINTS	83

5.3	ΤΟ ΑΝΕΚΤΕΛΕΣΤΟ ΥΠΟΛΟΙΠΟ ΤΟΥ SPRINT	84
5.4	ΟΙ ΚΑΘΗΜΕΡΙΝΕΣ SCRUM ΣΥΝΑΝΤΗΣΕΙΣ.....	87
5.4.1	ΚΑΘΟΡΙΣΜΟΣ ΤΟΥ ΤΟΠΟΥ ΚΑΙ ΤΟΥ ΧΡΟΝΟΥ ΤΩΝ SCRUM ΣΥΝΑΝΤΗΣΕΩΝ	88
5.5	ΣΥΝΑΝΤΗΣΗ ΣΧΕΔΙΑΣΜΟΥ ΕΚΔΟΣΗΣ ΛΟΓΙΣΜΙΚΟΥ	89
6	ΕΚΤΙΜΗΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ ΑΠΟ ΤΗΝ ΕΦΑΡΜΟΓΗ ΤΩΝ SCRUM ΤΕΧΝΙΚΩΝ.....	93
6.1	ΑΠΟΛΟΓΙΣΜΟΣ SPRINT	93
6.2	ΑΝΑΚΕΦΑΛΑΙΩΣΗ SPRINT (RETROSPECTIVE).....	94
6.3	ΔΙΑΓΡΑΜΜΑ SPRINT BURNDOWN.....	96
6.4	ΔΙΑΓΡΑΜΜΑ RELEASE BURNDOWN.....	99
6.5	ΟΛΟΚΛΗΡΩΣΗ ΕΡΓΑΣΙΩΝ	100
7	ΤΟ ΔΟΚΙΜΑΣΤΙΚΟ ΣΤΑΔΙΟ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ ΣΤΗ ΜΕΘΟΔΟΛΟΓΙΑ SCRUM	102
8	ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΜΕΘΟΔΟΛΟΓΙΑΣ SCRUM.....	107
9	ΣΥΓΚΡΙΤΙΚΗ ΑΝΑΛΥΣΗ ΤΟΥ ΧΡ ΚΑΙ ΤΗΣ SCRUM.....	110
9.1	ΠΡΩΤΗ ΔΙΑΣΤΑΣΗ: ΧΑΡΑΚΤΗΡΙΣΜΟΣ ΤΟΥ ΠΕΔΙΟΥ ΔΡΑΣΗΣ ΤΗΣ ΜΕΘΟΔΟΥ.....	110
9.2	ΔΕΥΤΕΡΗ ΔΙΑΣΤΑΣΗ: ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΗΣ ΕΥΕΛΙΞΙΑΣ.....	111
9.3	ΤΡΙΤΗ ΔΙΑΣΤΑΣΗ: ΧΑΡΑΚΤΗΡΙΣΜΟΣ ΤΩΝ ΕΥΕΛΙΚΤΩΝ ΤΙΜΩΝ.....	112
9.4	ΤΕΤΑΡΤΗ ΔΙΑΣΤΑΣΗ: ΧΑΡΑΚΤΗΡΙΣΜΟΣ ΤΗΣ ΔΙΑΔΙΚΑΣΙΑΣ ΛΟΓΙΣΜΙΚΟΥ.....	113
9.5	ΣΥΓΚΡΙΣΗ ΧΡ & SCRUM ΜΕ ΒΑΣΗ ΤΙΣ ΔΙΑΣΤΑΣΕΙΣ ΤΟΥ ΕΡΓΑΛΕΙΟΥ 4 - DAT.....	113
10	ΜΕΛΕΤΗ ΠΕΡΙΠΤΩΣΗΣ (CASE STUDY)	120
10.1	ΣΤΑΔΙΑ ΑΝΑΠΤΥΞΗΣ ΕΦΑΡΜΟΓΩΝ ΛΟΓΙΣΜΙΚΟΥ.....	120
10.2	ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΠΟΥ ΥΛΟΠΟΙΗΘΗΚΕ.....	123
10.3	ΤΟ ΠΛΗΡΟΦΟΡΙΑΚΟ ΣΥΣΤΗΜΑ ΜΕ ΧΡΗΣΗ ΤΗΣ ΜΕΘΟΔΟΛΟΓΙΑΣ SCRUM.....	125
11	ΣΥΜΠΕΡΑΣΜΑΤΑ	135
12	ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΑΝΑΦΟΡΕΣ	137

- Εικόνα 1: Αλληλεξαρτήσεις των εμπλεκομένων μερών ενός Π.Σ.
- Εικόνα 2: Συσχέτιση μεταξύ μεθόδων, τεχνικών και συστήματος.
- Εικόνα 3: Επίπεδα Σχεδίασης εφαρμογής.
- Εικόνα 4: Σχεδιασμός επαναλήψεων στη Scrum μεθοδολογία.
- Εικόνα 5: Αξιολόγηση επαναλήψεων.
- Εικόνα 6: Κύκλος ζωής RUP προσέγγισης.
- Εικόνα 7: eXtreme Programming.
- Εικόνα 8: Πολλαπλές ομάδες με επικαλυπτόμενα sprints.
- Εικόνα 9: Πολλαπλές ομάδες με συγχρονισμένα sprints.
- Εικόνα 10: Πολλαπλές ομάδες σε ένα Scrum project.
- Εικόνα 11: Πολλαπλές ομάδες & ανεκτέλεστο υπόλοιπο.
- Εικόνα 12: Πολλαπλές ομάδες & ανεκτέλεστο υπόλοιπο.
- Εικόνα 13: Πολλαπλές ομάδες & ανεκτέλεστο υπόλοιπο.
- Εικόνα 14: Κλιμάκωσης Scrum ομάδας
- Εικόνα 15: Λίστα εργασιών ανά προτεραιότητα.
- Εικόνα 16: Παράδειγμα πίνακα καθηκόντων 1.
- Εικόνα 17: Παράδειγμα πίνακα καθηκόντων 2.
- Εικόνα 18: Burndown Chart 1.
- Εικόνα 19: Burndown Chart 2.
- Εικόνα 20: Αφαίρεση μέρους εργασιών στο Burndown Chart.
- Εικόνα 21: Προσθήκη μέρους εργασιών στο Burndown Chart.
- Εικόνα 22: Release Burndown Chart.
- Εικόνα 23: Δοκιμαστική φάση αποδοχής.
- Εικόνα 24: Προσθήκη Tester στην ομάδα Scrum.
- Εικόνα 25: Συσσωρευμένα σφάλματα κατά την υλοποίηση των sprints.
- Εικόνα 26: Release εκδόσεις μεταξύ των sprints.
- Εικόνα 27: Bug fixes προηγούμενων sprints.
- Εικόνα 28: Βαθμός ευελιξίας στο XP και στη Scrum συγκεντρωτικά.

- Πίνακας 1: Χαρακτηριστικά εμπειρικών και θεωρητικών μεθόδων.
- Πίνακας 2: Release Plan.
- Πίνακας 3: Δραστηριότητες και φάσεις ανάπτυξης RUP.
- Πίνακας 4: Παράδειγμα περίπτωσης χρήσης.
- Πίνακας 5: Ενδεικτικές πληροφορίες τρέχοντος Sprint.
- Πίνακας 6: Πίνακας καθηκόντων.
- Πίνακας 7: Παράδειγμα ανεκτέλεστου υπόλοιπου προϊόντος.
- Πίνακας 8: Πεδία του ανεκτέλεστου υπόλοιπου.
- Πίνακας 9: Σωστή πρακτική ανακεφαλαίωσης και σχεδιασμού sprint.
- Πίνακας 10: Ιδανική πρακτική ανακεφαλαίωσης και σχεδιασμού sprint.
- Πίνακας 11: Παράδειγμα ανεκτέλεστου υπόλοιπου sprint.
- Πίνακας 12: Σύγκριση Scrum και μη ευέλικτων μεθοδολογιών.
- Πίνακας 13: Η πρώτη διάσταση του εργαλείου 4 – DAT.
- Πίνακας 14: Η δεύτερη διάσταση του εργαλείου 4 – DAT
- Πίνακας 15: Η τρίτη διάσταση του εργαλείου 4 – DAT.
- Πίνακας 16: Η τέταρτη διάσταση του εργαλείου 4 – DAT.
- Πίνακας 17: Αξιολόγηση του πεδίου εφαρμογής του XP και της Scrum.
- Πίνακας 18: Βαθμός ευελιξίας στο XP.
- Πίνακας 20: Βαθμός ευελιξίας στο XP και στη Scrum συγκεντρωτικά.
- Πίνακας 19: Βαθμός ευελιξίας στη Scrum.
- Πίνακας 21: Βαθμός ευελιξίας στο XP και στη Scrum για την 3^η διάσταση.
- Πίνακας 22: Βαθμός ευελιξίας στο XP και στη Scrum για την 4^η διάσταση.
- Πίνακας 23: Πίνακας Εργασιών (Βέλτιστος, εκτιμώμενος, και χειρότερος χρόνος).
- Πίνακας 24: Το Ανεκτέλεστο Υπόλοιπο του Προϊόντος.
- Πίνακας 25: Το Ανεκτέλεστο Υπόλοιπο του 1^{ου} Sprint.
- Πίνακας 26: Δοκιμή της πρώτης επανάληψης της εφαρμογής.
- Πίνακας 27: Επιθυμητή και Πραγματική πρόοδος.
- Πίνακας 28: Το Ανεκτέλεστο Υπόλοιπο του 2^{ου} Sprint.
- Πίνακας 29: Δοκιμή της δεύτερης επανάληψης της εφαρμογής.
- Πίνακας 30: Επιθυμητή και Πραγματική πρόοδος.

Πίνακας 31: Το Ανεκτέλεστο Υπόλοιπο του 3^{ου} Sprint.

Πίνακας 32: Δοκιμή της τρίτης επανάληψης της εφαρμογής.

Πίνακας 33: Επιθυμητή και Πραγματική πρόοδος.

1 ΕΙΣΑΓΩΓΗ

1.1 ΟΡΙΣΜΟΣ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ

Ο τομέας της ανάπτυξης λογισμικού έχει αναπτυχθεί ιδιαίτερα μέσα στις τελευταίες δεκαετίες και σε καμία περίπτωση δεν μπορούμε να αναφέρουμε ότι υστερεί στην εισαγωγή νέων μεθοδολογιών. Τα τελευταία χρόνια ένας μεγάλος αριθμός διαφορετικών προσεγγίσεων έχουν προταθεί, αλλά μόνο λίγες κατάφεραν να χρησιμοποιηθούν μέχρι και σήμερα. Οι υπεύθυνοι στον τομέα της ανάπτυξης λογισμικού φαίνεται ότι είναι ιδιαίτερα δύσπιστοι στις νέες προσεγγίσεις κυρίως όταν αυτές είναι δύσκολο να κατανοηθούν ή να ενσωματωθούν με επιτυχία στη διαδικασία ανάπτυξης ενός συστήματος.

Αυτό το κλίμα συνήθως επικρατούσε πριν από την εμφάνιση των ευέλικτων μεθοδολογιών ανάπτυξης λογισμικού. Πλέον οι ευέλικτες μεθοδολογίες ενσωματώνουν ένα σύνολο δοκιμασμένων τεχνικών που βοηθούν στην ανάπτυξη ποιοτικού λογισμικού.

Συγκεκριμένα η μεθοδολογία Scrum αναπτύχθηκε και χρησιμοποιείται προκειμένου να αντιμετωπίσει αποτελεσματικά οποιουδήποτε κινδύνους (risks) μπορεί να προκύψουν κατά την ανάπτυξη του λογισμικού. Χρησιμοποιεί ένα σύνολο δοκιμασμένων πρακτικών που υπόσχονται την ολοκλήρωση του έργου μέσα σε καθορισμένα χρονικά όρια ενώ ταυτόχρονα επιτρέπουν στην ομάδα ανάπτυξης του συστήματος να ανταποκρίνεται και να προσαρμόζεται με επιτυχία στο διαρκώς μεταβαλλόμενο επιχειρησιακό περιβάλλον. Ενθαρρύνει επίσης τη συμμετοχή όλων των μελών της ομάδας, σε όλες τις φάσεις της υλοποίησης, από τη θέσπιση των προδιαγραφών, μέχρι την εγκατάσταση και την τελική κυκλοφορία του συστήματος.

1.2 ΔΟΜΗ ΤΗΣ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Στο επόμενο κεφάλαιο της παρούσας εργασίας κάνουμε μια αναφορά τόσο στις παραδοσιακές προσεγγίσεις όσο και στις βασικότερες ευέλικτες μεθοδολογίες (RUP,

eXtreme Programming αναφέροντας τα βασικότερα χαρακτηριστικά τους), πλην βέβαια της Scrum μεθοδολογίας την οποία αναλύουμε λεπτομερώς στα επόμενα κεφάλαια. Στο τρίτο και τέταρτο κεφάλαιο ξεκινάμε τη μελέτη της μεθοδολογίας Scrum αναλύοντας τις κύριες φάσεις της μεθοδολογίας, τις δραστηριότητες, το γενικό πλαίσιο και τους ρόλους της.

Στη συνέχεια εμβαθύνουμε ακόμη περισσότερο στη μεθοδολογία Scrum μελετώντας τις πρακτικές που χρησιμοποιεί όπως το ανεκτέλεστο υπόλοιπο προϊόντος (product backlog), το ανεκτέλεστο υπόλοιπο sprint (sprint backlog), τη συνάντηση σχεδιασμού του sprint (sprint planning meeting), τις καθημερινές Scrum συναντήσεις (daily Scrums) και τη συνάντηση σχεδιασμού έκδοσης λογισμικού (release planning meeting).

Το έκτο κεφάλαιο της εργασίας αναφέρεται στους μηχανισμούς οι οποίοι χρησιμοποιούνται στη Scrum μεθοδολογία για την εκτίμηση των αποτελεσμάτων από τη χρήση των παραπάνω Scrum τεχνικών. Έτσι μελετάμε τα διαγράμματα sprint και release burndown charts και αναλύουμε επίσης τον απολογισμό του sprint (sprint review) και την ανακεφαλαίωση του sprint (sprint retrospective).

Στη συνέχεια αναφερόμαστε στο δοκιμαστικό στάδιο του συστήματος και εξηγούμε για ποιο λόγο το στάδιο αυτό είναι τόσο σημαντικό όχι μόνο στη μεθοδολογία Scrum, αλλά γενικότερα στην ανάπτυξη λογισμικού. Στο επόμενο κεφάλαιο αναλύουμε να βασικότερα πλεονεκτήματα της Scrum μεθοδολογίας σε σχέση με τις παραδοσιακές μεθοδολογίες.

Στο ένατο κεφάλαιο γίνεται συγκριτική ανάλυση της μεθοδολογίας Scrum με αυτή του eXtreme Programming χρησιμοποιώντας το εργαλείο 4-DAT (Dimension Analytical Tool) όπου και καταλήγουμε στο συμπέρασμα ότι η μεθοδολογία Scrum είναι πιο ευέλικτη όσο αφορά τις πρακτικές που χρησιμοποιεί.

Στο προτελευταίο κεφάλαιο της παρούσας εργασίας μέσω της μελέτης περίπτωσης χρήσης (case study) η μεθοδολογία Scrum εφαρμόζεται για την ανάπτυξη

συστήματος καθώς επίσης και για τη συγκριτική αξιολόγηση σε σχέση με τη μεθοδολογία που χρησιμοποιήθηκε όταν υλοποιήθηκε το εν λόγω σύστημα. Το τελευταίο κεφάλαιο της παρούσας εργασίας, αναφέρει τα συμπεράσματα που μπορούμε να εξάγουμε για τη Scrum μεθοδολογία γενικά, αλλά και συγκριτικά σε σχέση με τις υπόλοιπες μεθοδολογίες, ευέλικτες και παραδοσιακές.

1.3 ΣΥΝΕΙΣΦΟΡΑ ΤΗΣ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Στόχος της παρούσας εργασίας δεν αποτελεί η ευρεία αποδοχή και εδραίωση της Scrum μεθοδολογίας στη διαχείριση και στην ανάπτυξη έργων λογισμικού. Συνεισφορά της συγκεκριμένης διπλωματικής εργασίας αποτελεί η λεπτομερής μελέτη και αξιολόγηση της μεθοδολογίας Scrum καθώς επίσης και η εις βάθος ανάλυση των πρακτικών που χρησιμοποιεί. Μελετώντας τις πρακτικές της καταλαβαίνουμε ότι πρόκειται για ένα θεωρητικό υπόβαθρο που αντιμετωπίζει με διαφορετικό τρόπο τη δομή και την παρακολούθηση των έργων λογισμικού.

Επίσης μέσω της μελέτης περίπτωσης χρήσης (case study) η μεθοδολογία Scrum προσφέρεται για τη συγκριτική αξιολόγηση και ανάλυση, επισημαίνοντας τα δυνατά και τα αδύνατα σημεία της.

Γενικότερα, η αφομοίωση και η κατανόηση της Scrum μεθοδολογίας όπως επίσης των ευέλικτων μεθοδολογιών αλλά και του ευέλικτου τρόπου σκέψης από φορείς και οργανισμούς, οδηγεί στην ανάπτυξη και την παράδοση ποιοτικού λογισμικού on time.

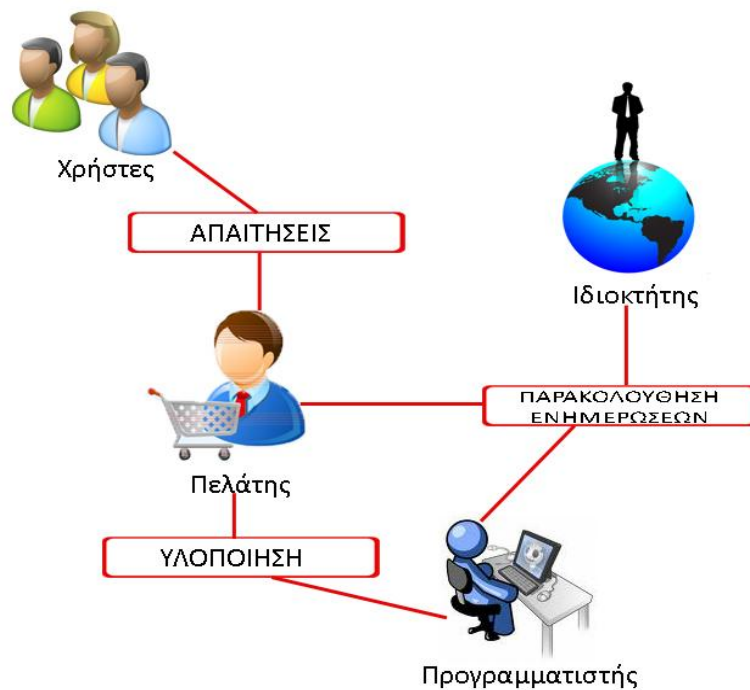
Τα δεδομένα στις τυπικές μέχρι τώρα προσεγγίσεις για την ανάπτυξη των πληροφοριακών συστημάτων είναι τα έγγραφα των απαιτήσεων του συστήματος και περιλαμβάνουν τις πλήρεις προδιαγραφές του προϊόντος που πρόκειται να αναπτυχθεί. Από τις προδιαγραφές αυτές μπορούν να εξαχθούν άμεσα συμπεράσματα και για τις απαιτούμενες προδιαγραφές από πλευράς λογισμικού μιας και αυτές αναγράφονται στο αρχικό έγγραφο με κάθε λεπτομέρεια. Στη συνέχεια οι απαιτήσεις αυτές μετατρέπονται σε ένα πλάνο ανάπτυξης, κυρίως με τη χρήση διαγραμμάτων Gantt και υπολογιστικών φύλλων έτσι ώστε να έχουμε μια εκτίμηση για τις εργασίες που θα πρέπει να ολοκληρωθούν. Η πρόοδος των εργασιών μετράται σε σχέση με την ολοκλήρωση των καθηκόντων για κάθε υπολογιστικό πόρο με βάση το αρχικό πλάνο σχεδίασης.

Δεν είναι πάντοτε όμως τόσο ευέλικτο να αλλάξουμε το πλάνο των εργασιών σε περίπτωση που κάποια από αυτές καθυστερήσει να ολοκληρωθεί. Πολλές φορές η επικοινωνία με όλα τα συμβαλλόμενα μέρη είναι αρκετά δύσκολη και πολύπλοκη και επομένως δεν είναι εύκολο να έχουμε μια πλήρη εικόνα για το στάδιο ανάπτυξης στο οποίο βρισκόμαστε. Πολύ συχνά μπορεί να αναφέρουμε ότι έχουμε καθυστερήσει την ανάπτυξη ενός συστήματος, αλλά είναι δύσκολο να καθορίσουμε πόσο ακριβώς έχουμε καθυστερήσει. Αυτό γίνεται συνήθως αντιληπτό όσο πλησιάζουμε προς την καταληκτική ημερομηνία παράδοσης του project. Η τμηματική παράδοση του προϊόντος δεν είναι πολλές φορές εφικτή καθώς ο φόρτος των εργασιών που πρέπει να ολοκληρωθεί δεν επαρκεί με βάση το χρονικό διάστημα το οποίο έχει αρχικά οριστεί. Προβλήματα που μπορεί να προκύψουν στην πορεία και δεν έχουν αρχικά προβλεφθεί, μπορεί να διακόψουν τη ροή του αρχικού πλάνου και έτσι μπορεί να έχουμε χαμηλή ποιότητα ολοκλήρωσης. Επίσης με τις παραδοσιακές προσεγγίσεις έμφαση δίνεται στο σημείο στο οποίο θα θέλαμε να βρισκόμασταν, και όχι στο πού είμαστε τώρα ή πού θα μπορούσαμε να ήμασταν (wishful planning). [Jamie Allsop, “Foundations of agile development”, 2008, p 4]

Ο σκοπός της δημιουργίας μιας εφαρμογής είναι η παροχή λύσεων και διευκολύνσεων στα προβλήματα των χρηστών που θα τη χρησιμοποιήσουν. Αυτό απαιτεί τη δημιουργία ποιοτικού κώδικα από την ομάδα ανάπτυξης του έργου και με όσο το δυνατό μικρότερο κόστος για την επιχείρηση που θα χρησιμοποιήσει το σύστημα. Ο συνδυασμός των δύο παραπάνω παραγόντων (υψηλή ποιότητα υπηρεσιών με τη μικρότερη δυνατή οικονομική επιβάρυνση) οδηγεί τις περισσότερες φορές σε αποτυχία των πληροφοριακών συστημάτων.

Το πρόβλημα όσο αφορά την τεχνολογική διάσταση σχετίζεται με τρεις παράγοντες. Πρώτον, τον καθορισμό των απαιτήσεων κατά τη διάρκεια του οποίου ορίζουμε τη φύση της εφαρμογής και το βαθμό στον οποίο θα ικανοποιεί τις ανάγκες των χρηστών. Δεύτερον, το στάδιο της ανάπτυξης όπου ξεκινάει η υλοποίηση της εφαρμογής η οποία θα πρέπει να καλύπτει τις απαιτήσεις που προσδιορίστηκαν και τέλος το στάδιο της συντήρησης όπου τροποποιούμε το σύστημα ώστε να καλύπτει τις νέες απαιτήσεις που μπορεί να προκύψουν διατηρώντας ταυτόχρονα σε υψηλό επίπεδο τη λειτουργικότητα του συστήματος. [Dan Rawsthorne, “Comparing/Combining RUP, XP, and Scrum – mixing the Process Cocktail”, 2003, p 4]

Από οικονομικής άποψης τα προβλήματα που μπορεί να προκύψουν μπορεί να σχετίζονται με το ύψος του ποσού που η επιχείρηση είναι διατεθειμένη να ξοδέψει ώστε να επιλύσει τα προβλήματα των χρηστών, με το βαθμό στον οποίο οι developers της εφαρμογής μπορούν να επιλύσουν τα παραπάνω προβλήματα και με το αν η υλοποίηση της εφαρμογής προχωρά με βάση το χρονοδιάγραμμα που είχε αρχικά σχεδιαστεί ή σπαταλιέται άσκοπα χρόνος και χρήματα. Επίσης μας ενδιαφέρει η ποιότητα και η αρτιότητα του συστήματος καθώς επίσης και ο βαθμός στον οποίο τα ενδιαφερόμενα μέρη είναι ικανοποιημένα από την υλοποίηση και τη λειτουργία του. Οι σχέσεις μεταξύ των εμπλεκόμενων μερών κατά την ανάπτυξη ενός νέου πληροφοριακού συστήματος απεικονίζεται στην παρακάτω εικόνα. [Dan Rawsthorne, “Comparing/Combining RUP, XP, and Scrum – mixing the Process Cocktail”, 2003, p 4]



Εικόνα 1: Αλληλεξαρτήσεις των εμπλεκόμενων μερών ενός Π.Σ. [Dan Rawsthorne, “Comparing/Combining RUP, XP, and Scrum – mixing the Process Cocktail”, 2003 p 5]

Αυτό που γίνεται εύκολα κατανοητό με βάση την παραπάνω εικόνα είναι ότι για την ανάπτυξη ενός Π.Σ. συμμετέχουν ενεργά όλα τα εμπλεκόμενα μέρη. Για παράδειγμα οι τελικοί χρήστες της εφαρμογής με τις περιγραφές τους δημιουργούν τις απαιτήσεις του συστήματος οι οποίες στη συνέχεια θα βοηθήσουν τον προγραμματιστή στην υλοποίηση της εφαρμογής. Από την άλλη πλευρά, ο προγραμματιστής θα πρέπει να βρίσκεται σε συνεχή επικοινωνία με τον ιδιοκτήτη του συστήματος ενημερώνοντάς τον όταν έχει έτοιμο ένα μέρος από τη συνολική εφαρμογή. Ο ιδιοκτήτης του συστήματος μπορεί να παρακολουθεί οποιαδήποτε χρονική στιγμή την υλοποίηση των εργασιών της εφαρμογής καθώς επίσης και οποιεσδήποτε αλλαγές πραγματοποιούνται σε αυτό.

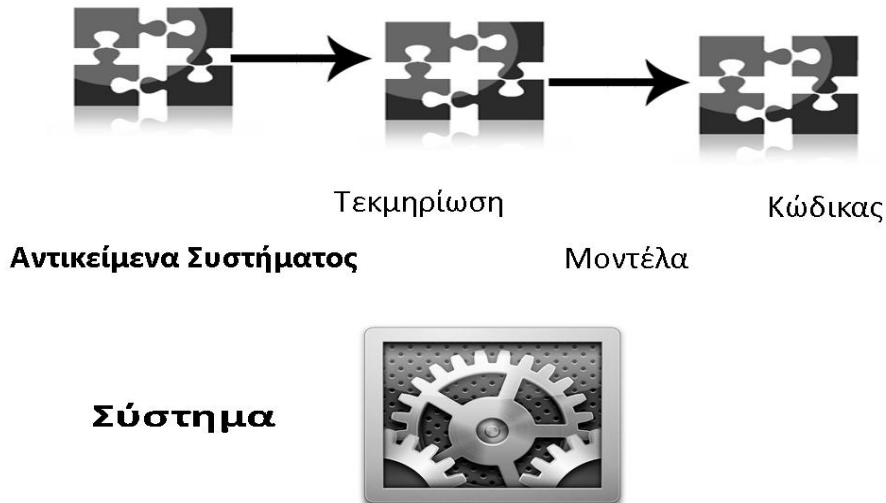
2.1 ΜΕΘΟΔΟΛΟΓΙΕΣ ΑΝΑΠΤΥΞΗΣ ΣΥΣΤΗΜΑΤΩΝ

Η ανάπτυξη ενός συστήματος είναι το αποτέλεσμα της δημιουργίας μιας λογικής δομής η οποία υλοποιείται σε ηλεκτρονικούς υπολογιστές. Η λογική αυτή δομή,

αποτελείται από εισόδους, διαδικασίες και εξόδους τόσο για ενδιάμεσα στάδια της εφαρμογής όσο και για το σύνολό της, το οποίο είναι γνωστό και ως ολοκληρωμένο πληροφοριακό σύστημα.

Πολλές τεχνικές εφαρμόζονται κατά την ανάπτυξη ενός συστήματος και μπορεί να χρησιμοποιούνται για τον έλεγχο του ανταγωνισμού και τη δημιουργία ενός δοκιμαστικού προϊόντος. Οι τεχνικές αποτελούνται από έγγραφα, μοντέλα, προγράμματα, δοκιμαστικές περιπτώσεις και άλλα παραδοτέα που δημιουργούνται με σκοπό την ανάπτυξη ενός συστήματος. Η προσέγγιση που χρησιμοποιείται για την ανάπτυξη ενός συστήματος είναι γνωστή ως μέθοδος και περιγράφει τις ενέργειες που περιλαμβάνονται για την υλοποίηση του συστήματος.

Μία μέθοδος έχει μικρο- και μακρο- συστατικά. Τα μακρο-συστατικά καθορίζουν τη συνολική ροή και το χρονικό πλαίσιο για την εκτέλεση μιας εργασίας. Τα μικρο-συστατικά περιλαμβάνουν τους γενικούς κανόνες σχεδίασης και μοτίβα για την επίτευξη ή την αποφυγή ενεργειών κατά τη διάρκεια του σχεδιασμού ή γενικές προσεγγίσεις που θα εφαρμοστούν κατά την ανάπτυξη του συστήματος. Τα μοτίβα είναι λύσεις που μπορούν να εφαρμοστούν στα προβλήματα που μπορεί να εμφανιστούν. Η παρακάτω εικόνα αποτυπώνει τις σχέσεις μεταξύ των μεθόδων, των τεχνικών και του συστήματος. [Ken Schwaber, "Scrum Development Process", p 18]



Εικόνα 2: Συσχέτιση μεταξύ μεθόδων, τεχνικών και συστήματος [Ken Schwaber, Scrum Development Process, p 19]

Η δομή ενός συστήματος αποτελείται από εισόδους διαδικασίες και εξόδους. Κατά την ανάπτυξη ενός οποιουδήποτε συστήματος δημιουργούνται αντικείμενά του, τα οποία χρησιμοποιούνται ως οδηγοί για τον έλεγχο και το βαθμό ολοκλήρωσης του συστήματος. Κύρια αντικείμενα ενός συστήματος μπορεί να αποτελεί η περιγραφή των απαιτήσεων, η τεκμηρίωσή του, ο κώδικας της εφαρμογής καθώς επίσης και οι περιπτώσεις χρήσεις με τη βοήθεια των οποίων καταλήγουμε στην ολοκλήρωση του συστήματος που υλοποιείται.

Οι μέθοδοι που χρησιμοποιούνται για τον έλεγχο των διαδικασιών σε πληροφοριακά που αναπτύσσονται διακρίνονται σε δύο κατηγορίες, τις θεωρητικές και τις εμπειρικές. Η σωστή κατηγοριοποίηση των μεθόδων είναι ιδιαίτερα σημαντική, η κατάλληλη δομή μιας μεθόδου που χρησιμοποιείται σε μία εφαρμογή εξαρτάται από το αν η μέθοδος είναι θεωρητική ή εμπειρική. Τα κυριότερα χαρακτηριστικά των εμπειρικών και θεωρητικών μεθόδων περιγράφονται στον πίνακα που ακολουθεί. [Ken Schwaber, Scrum Development Process, p 20]

Θεωρητική Μοντελοποίηση	Εμπειρική μοντελοποίηση
Συνήθως απαιτούνται λιγότερες μετρήσεις, απαιτεί σχετικό	Απαιτεί εκτεταμένες μετρήσεις μιας και βασίζεται ολοκληρωτικά στον

πειραματισμό μόνο για τον υπολογισμό άγνωστων παραμέτρων.	πειραματισμό των μοντέλων υλοποίησης.
Παρέχει πληροφορίες για την εσωτερική κατάσταση μιας διαδικασίας.	Παρέχει πληροφορίες μόνο για το τμήμα της διαδικασίας που επηρεάζεται από τη διαδικασία ελέγχου.
Υποστηρίζει τη θεμελιώδη κατανόηση των εσωτερικών λειτουργιών μιας διαδικασίας.	Μεταχειρίζεται τη διαδικασία ως ένα χρονοκύβο.
Απαιτεί ακριβή και ολοκληρωμένη γνώση της διαδικασίας.	Δεν απαιτεί τόσο λεπτομερή γνώση, παρά μόνο για τα δεδομένα εξόδου που επηρεάζονται από τις αλλαγές στα δεδομένα εισόδου.
Δεν παρέχει ικανοποιητική κατανόηση των σύνθετων διαδικασιών.	Είναι η μοναδική εναλλακτική λύση για την κατανόηση των σύνθετων διαδικασιών.
Παρέχει τόσο γραμμική όσο και μη γραμμική μοντελοποίηση διαδικασιών.	Απαιτεί συγκεκριμένες μεθόδους για την παραγωγή μη γραμμικής μοντελοποίησης.

Πίνακας 1: Χαρακτηριστικά εμπειρικών και θεωρητικών μεθόδων

Επομένως καταλήγουμε στο συμπέρασμα ότι η διαδικασία υλοποίησης των πληροφοριακών συστημάτων είναι εμπειρική. Ωστόσο οι μεθοδολόγοι πιστεύουν ότι δεν μπορεί μια μέθοδος να αναλύσει λεπτομερώς τον τρόπο με τον οποίο θα πρέπει να γίνονται οι διαδικασίες. Η ανάπτυξη λογισμικού είναι μια δημιουργική διαδικασία, ενώ μια μέθοδος παρέχει ένα γενικό πλαίσιο το οποίο περιγράφει το σημείο όπου πρέπει να είμαστε δημιουργικοί. Πολλά προβλήματα στην ανάπτυξη των συστημάτων υπήρξαν εξαιτίας λανθασμένης κατηγοριοποίησης. Όταν μια διαδικασία βασιζόμενη σε χρονοκύβους θεωρείται ως μια πλήρως καθορισμένη διαδικασία τότε μη προβλέψιμες καταστάσεις μπορούν να συμβούν και οι έλεγχοι δεν μπορούν να μετρήσουν και να αντιμετωπίσουν τη μη προβλεψιμότητα.

Στην πράξη, πολλές εταιρίες και σε πολλά projects η μεθοδολογία Scrum έχει παρατηρηθεί ότι παρέχει μια βιώσιμη λύση σε πολλά προβλήματα. Τα συστήματα παραδίδονται εντός του χρονικού προγραμματισμού και σε πολλές περιπτώσεις ξεπερνάνε τις προσδοκίες τόσο των χρηστών όσο και της διοίκησης του οργανισμού. Παρόλο που είναι ιδιαίτερα απαιτητικό για μια ομάδα να εργάζεται σε ένα Scrum project, οι προγραμματιστές ανταμείβονται από το υψηλό ομαδικό πνεύμα και μια αίσθηση ότι η υλοποίηση ενός πληροφοριακού συστήματος μπορεί να είναι μια ευχάριστη και ικανοποιητική εμπειρία. [Ken Schwaber, “Scrum Development Process”, p 21]

2.2 ΕΥΕΛΙΚΤΕΣ ΠΡΟΣΕΓΓΙΣΕΙΣ ΣΤΗΝ ΑΝΑΠΤΥΞΗ ΣΥΣΤΗΜΑΤΩΝ

Η παρουσία των ευέλικτων μεθοδολογιών αλλάζει τον παραδοσιακό τρόπο ανάπτυξης των πληροφοριακών συστημάτων αλλάζοντας τη διαδικασία που περιγράφηκε παραπάνω και απλουστεύοντας τις αλληλεξαρτήσεις των εμπλεκόμενων μερών. Οι κύριες ευέλικτες (agile) μεθοδολογίες είναι οι ακόλουθες: [Dan Rawsthorne, “Comparing/Combining RUP, XP, and Scrum – mixing the Process Cocktail”, 2003, p 6,7]

- Rational Unified Process (RUP).
- eXtreme Programming.
- Scrum.

Δεν είναι οι μοναδικές αλλά είναι οι πιο γνωστές και οι πιο διαδεδομένες, κάποιες άλλες ευέλικτες προσεγγίσεις αλλά όχι τόσο ευρέως χρησιμοποιούμενες είναι οι εξής:

- Dynamic System Development Method (DSDM).
- Crystal Methodologies.
- Feature Driven Development (FDD).
- Adaptive software Development.

Οι βασικές διαδικασίες στις ευέλικτες μεθοδολογίες καθορίζουν τι πρέπει να κάνουμε και πότε πρέπει να το κάνουμε.

- Plan driven (οδηγούμενες με βάση σχέδιο), καθορίζουν εκ των προτέρων τι θα πρέπει να κάνουμε.
- Evaluation driven (οδηγούμενες με βάση αξιολόγηση), οποιαδήποτε χρονική στιγμή καθορίζουμε το επόμενο βήμα.
- Συνδυασμός των δύο παραπάνω, έχουμε ένα αρχικό πλάνο και συνεχώς το ενημερώνουμε όποτε χρειάζεται.

Οι παραπάνω διαδικασίες είναι απαραίτητες επειδή οι ιδιοκτήτες του συστήματος έχουν υψηλές προσδοκίες για την εφαρμογή και χρειάζονται κάποιες εγγυήσεις ότι θα καλυφθούν όλες οι περιγραφόμενες απαιτήσεις. Αυτό που παίζει καθοριστικό ρόλο για τις ευέλικτες μεθοδολογίες είναι η συνοχή της ομάδας και συγκεκριμένα όσο πιο πειθαρχημένη είναι η ομάδα ανάπτυξης που θα υλοποιήσει την εφαρμογή, τόσο πιο απλή γίνεται μια διαδικασία. Τις περισσότερες φορές όμως τα μέλη μιας ομάδας δεν είναι πειθαρχημένα, έτσι οι διαδικασίες είναι απαραίτητες και χρήσιμες για τον καθορισμό του λόγου για τον οποίο είναι απαραίτητη η ολοκλήρωση μιας συγκεκριμένης διαδικασίας.

Η ευελιξία είναι απαραίτητη σε όλα τα στάδια ανάπτυξης των πληροφοριακών συστημάτων. Ευελιξία είναι η ικανότητα να προλαβαίνουμε τις εξελίξεις οι οποίες μπορεί να έχουν δυσμενείς συνέπειες για την εφαρμογή μας, καθώς επίσης και η ικανότητα να συμβαδίζουμε με τις σχετικές αλλαγές των απαιτήσεων, των γνώσεων του συστήματος και του περιβάλλοντος στο οποίο αυτό αναπτύσσεται.

Οι ευέλικτες προσεγγίσεις (agile development) βασίζονται σε ένα σύνολο ευέλικτων αρχών και πρακτικών, που σκοπό τους έχουν την ενθάρρυνση των αλλαγών των απαιτήσεων που μπορεί να προκύψουν σε ένα project. Επομένως οι όποιες αλλαγές είναι ευπρόσδεκτες ως αποτέλεσμα μιας καλύτερης κατανόησης των επιχειρησιακών αναγκών. Η πρόοδος μετράται σε σχέση με το παραγόμενο προϊόν που παραδίδεται στον πελάτη, ελαχιστοποιώντας τη σπατάλη χρόνου για τις

αλλαγές που μπορεί να προκύψουν. Η πρόοδος των εργασιών που ολοκληρώθηκαν χρησιμοποιούνται για την πρόβλεψη του χρόνου ολοκλήρωσης των μελλοντικών εργασιών. Οι βασικότερες αρχές των ευέλικτων μεθοδολογιών είναι οι παρακάτω:

- Ενθάρρυνση των αλλαγών των απαιτήσεων.
- Συνεχής παράδοση τμημάτων του προϊόντος που έχουν ολοκληρωθεί.
- Αυτό – οργανωσιακές ομάδες.
- Συνεργασία όλων των μελών της ομάδας (π.χ. ιδιοκτήτες, προγραμματιστές).
- Εμπιστοσύνη στα κίνητρα των μελών της ομάδας.
- Το λογισμικό που λειτουργεί είναι το πρωταρχικό μέτρο προόδου.
- Τεχνική αρτιότητα και καλός σχεδιασμός.
- Η απλότητα είναι απαραίτητη.

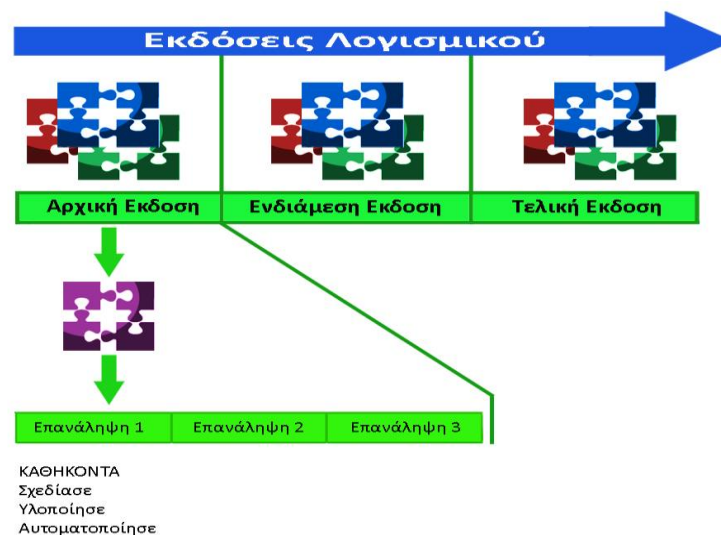
Πελάτης του συστήματος στην ευέλικτη μεθοδολογία μπορεί να θεωρηθεί οποιοσδήποτε χρήστης μπορεί να χρησιμοποιήσει την εφαρμογή. Η ομάδα ανάπτυξης μπορεί να θεωρηθεί ως πελάτης τεχνικής υποστήριξης του συστήματος. Γενικότερα ο καθορισμός των πραγματικών πελατών δεν είναι πάντοτε εύκολη διαδικασία, παρόλο που οι πελάτες και οι χρήστες είναι πολλές φορές ίδιοι, εφόσον χρησιμοποιούν την εφαρμογή. Τυπικά ο όρος πελάτης έχει ευρύτερη έννοια ώστε να συμπεριλάβει όλα τα ενδιαφερόμενα μέρη που θα χρησιμοποιήσουν το σύστημα. Οι περιγραφές των χρηστών αποτυπώνουν τα χαρακτηριστικά του συστήματος από τη σκοπιά του χρήστη και τις λειτουργικές του απαιτήσεις με μοιόμορφο αλλά ταυτόχρονα ανεπίσημο τρόπο. Τυπικά παραδείγματα χρηστών ενός συστήματος μπορεί να είναι ο system administrator, ο security administrator, ο τελικός χρήστης και ο υπεύθυνος τεχνικής υποστήριξης.

Η πρόοδος των εργασιών βασίζεται στη χρήση χρονοκύβων (time boxes) και γίνεται ξεκινώντας έναν αριθμό επαναλήψεων (iterations), οι οποίες έχουν προκαθορισμένη χρονική διάρκεια. Η ομάδα ανάπτυξης επικεντρώνεται στο σκοπό της κάθε επανάληψης που είναι σχετικός με τις απαιτήσεις των χρηστών, ενώ

ταυτόχρονα παρατηρεί την πρόοδο που έχει συντελεστεί με βάση το σύνολο των απαιτήσεων. Η χρήση των time boxes μας επιτρέπει να υπολογίσουμε το χρόνο που απαιτείται για να τελειώσουν οι μη ολοκληρωμένες εργασίες μέσα σε κάθε επανάληψη. Ολοκλήρωση μιας επανάληψης σημαίνει ενσωμάτωση με τις προηγούμενες διαδικασίες, δοκιμή και τεκμηρίωση, δηλαδή κάθε εργασία θα πρέπει να τελειώνει με ένα πλήρως λειτουργικό τμήμα λογισμικού.

Ο σχεδιασμός μιας εφαρμογής στις ευέλικτες μεθοδολογίες γίνεται σε διαφορετικά επίπεδα και ενημερώνεται συνεχώς όπως φαίνεται και στην ακόλουθη εικόνα. Έτσι μπορούμε να μιλάμε για:

- Release Planning, που πραγματοποιείται μεταξύ όλων των επαναλήψεων (iterations).
- Iteration Planning, που πραγματοποιείται σε κάθε επανάληψη.
- Daily Planning, που πραγματοποιείται κάθε ημέρα ώστε να υπάρχει συντονισμός των εργασιών.



Εικόνα 3: Επίπεδα Σχεδίασης εφαρμογής [Jamie Jamie Allsop, "Foundations of agile development", 2008, p 15]

Η Release πρόοδος μετράται σε σχέση με το σύνολο των απαιτήσεων των χρηστών που ολοκληρώθηκαν, ενώ η πρόοδος σε κάθε επανάληψη μετράται σε σχέση με τις εργασίες που ολοκληρώθηκαν μέσα στην επανάληψη. Εκεί που δίνεται ιδιαίτερη

προσοχή είναι στο τι πρέπει στη συνέχεια να γίνει και όχι ο χρόνος που πέρασε μέχρι να φτάσουμε στο σημείο που βρισκόμαστε τώρα. Στις ευέλικτες μεθοδολογίες απαραίτητο είναι να ενημερώνονται τα πλάνα εργασιών ώστε να αποτυπώνονται οι νέες απαιτήσεις που μπορεί να προέκυψαν και όχι να προστατεύουμε τον αρχικό μας σχεδιασμό από νέες αλλαγές. Επειδή σαν πρώτο στόχο οι ευέλικτες προσεγγίσεις έχουν την ικανοποίηση του πελάτη, θα πρέπει συγκρίνουμε σε κάθε επανάληψη αν η πρόοδος που έχει γίνει είναι ανάλογη με την αναμενόμενη, καθώς επίσης και να σκεφτόμαστε πώς θα προχωρήσουμε στη συνέχεια. Η ανακεφαλαίωση μετά από κάθε iteration είναι ζωτικής σημασίας και δεν θα πρέπει να αποφεύγεται, αν θέλουμε να έχουμε σωστή ανατροφοδότηση επί της συνολικής διαδικασίας. Τυπικά θα πρέπει να κάνουμε μία αλλαγή σε κάθε επανάληψη και να δώσουμε χρόνο ώστε να αξιολογήσουμε τις επιπτώσεις τους. Ανατροφοδότηση από τις αλλαγές που έγιναν θα πάρουμε στις επόμενες επαναλήψεις. [Jamie Allsop, “Foundations of agile development”, 2008, p 16]

2.3 ΟΙ ΕΥΕΛΙΚΤΕΣ ΜΕΘΟΔΟΛΟΓΙΕΣ ΜΕ ΜΙΑ ΔΕΥΤΕΡΗ ΜΑΤΙΑ

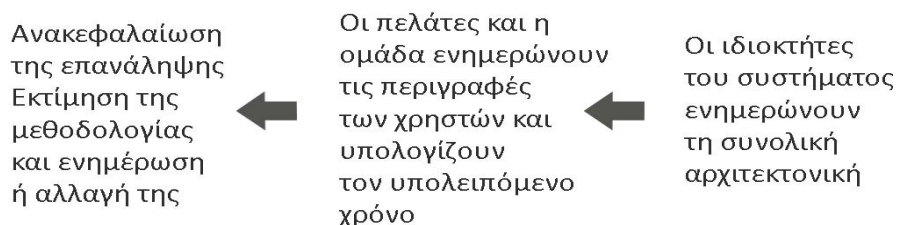
Στο ξεκίνημα μιας ευέλικτης μεθοδολογίας και εφόσον πρόκειται για τη δημιουργία νέου προϊόντος θα πρέπει να παράγουμε νέες ιδέες και να εισάγουμε το πρώτο σύνολο των απαιτήσεων των χρηστών και τα αναμενόμενα χρονοδιαγράμματα. Όταν πρόκειται για υπάρχοντα προϊόντα, συγκεντρώνουμε ένα σύνολο περιγραφών για έναν αριθμό επαναλήψεων. Όπως αναφέραμε οι επαναλήψεις γίνονται σε timeboxes σταθερού μήκους (τις περισσότερες φορές διάρκειας έως και τρεις εβδομάδες). Για το σχεδιασμό των επαναλήψεων ο πελάτης βάζει σε σειρά προτεραιότητας τις περιγραφές και η ομάδα υπολογίζει αν μπορεί να δεσμευτεί για την υλοποίησή τους. Στη συνέχεια ο πελάτης και η ομάδα καθορίζουν από τις απαιτήσεις τις περιπτώσεις χρήσεις (use cases) και αξιολογούν τις περιπτώσεις χρήσης. Η ομάδα ενημερώνει τη συνολική αρχιτεκτονική και καθορίζει καθήκοντα με βάση τις απαιτήσεις και τις περιπτώσεις χρήσεις. Αν όλα πάνε καλά τότε η ομάδα προχωράει στην υλοποίηση για τη συγκεκριμένη επανάληψη. Όλα αυτά απεικονίζονται και στην παρακάτω εικόνα.



Εικόνα 4: Σχεδιασμός επαναλήψεων στη Scrum μεθοδολογία [Jamie Allsop, “Foundations of agile development”, 2008, p 20]

Στη φάση της αξιολόγησης της επανάληψης, οι πελάτες και η ομάδα ενημερώνουν τις απαιτήσεις, προσθέτοντας καινούργιες όπου αυτό απαιτείται. Χωρίζουν μεγάλες περιγραφές χρηστών σε μικρότερες, ενημερώνουν τη λίστα προτεραιοτήτων και υπολογίζουν τον υπολειπόμενο χρόνο για την ολοκλήρωση του έργου. Διαγραμματικά όλες αυτές τις ενέργειες αποτυπώνονται και στην εικόνα που ακολουθεί.








ΑΞΙΟΛΟΓΗΣΗ ΕΠΑΝΑΛΗΨΗΣ



Εικόνα 5: Αξιολόγηση επαναλήψεων [Jamie Allsop (2008), Foundations of Agile Development p 21]

Το release plan μπορεί να χαρακτηριστεί ως οδικός χάρτης με βάση τον οποίο βαδίζουμε. Το πλάνο αυτό υπολογίζει τις αλλαγές που πρέπει να γίνουν καθώς οι

απαιτήσεις ολοκληρώνονται όπως επίσης και την πρόοδο ολοκλήρωσης των εργασιών όπως φαίνεται παρακάτω.

Ορόσημο	Περιγραφές	Υπολειπόμενος φόρτος	Συνολικό μέγεθος	Ημερομηνία ολοκλήρωσης	Πρόοδος
1	9	-	26	Ολοκληρώθηκε	
2	13	-	103	Ολοκληρώθηκε	
3	40	-	219	Ολοκληρώθηκε	
4	56	330	498	20/07/2012	
5	15	100	100	30/07/2012	
6	9	200	200	10/08/2012	
7	27	103	200	25/08/2012	

Πίνακας 2: Release Plan [Jamie Allsop, “Foundations of agile development”, 2008, p 22]

Για παράδειγμα βλέπουμε ότι για το 2^ο ορόσημο δεν απομένει υπολειπόμενος φόρτος εργασιών μιας και όλες οι απαιτήσεις για τις περιγραφές των χρηστών έχουν ολοκληρωθεί. Αυτό ακριβώς υποδηλώνει και το πράσινο χρώμα στη στήλη της προόδου. Αντίστοιχα το ορόσημο 6 δεν έχει ξεκινήσει να υλοποιείται ακόμη μιας και ο υπολειπόμενος φόρτος εργασιών είναι ίδιος με το συνολικό μέγεθος των εργασιών. Για το λόγο αυτό το χρώμα στη στήλη προόδου είναι κόκκινο.

Όσο αφορά τις απαιτήσεις των χρηστών υπάρχουν δύο κατηγορίες, οι ολοκληρωμένες και οι μη ολοκληρωμένες. Τα μη ολοκληρωμένα καθήκοντα μπορεί να έχουν επιλεγεί ώστε να ολοκληρωθούν κατά τη διάρκεια της τρέχουσας επανάληψης, να εκτιμάται ότι θα τελειώσουν μέχρι το τέλος της επανάληψης ή να μην έχουν επιλεγεί να ολοκληρωθούν την παρούσα χρονική στιγμή. Αντίθετα οι απαιτήσεις θεωρούνται ολοκληρωμένες ή μη ανάλογα με το στάδιο της υλοποίησης στο οποίο βρίσκονται, την τεκμηρίωσή τους, τις περιπτώσεις χρήσης που καλύπτουν και το πόσο αποτελεσματικά καλύπτουν τις περιγραφές των χρηστών. [Jamie Allsop, “Foundations of agile development”, 2008, p 16]

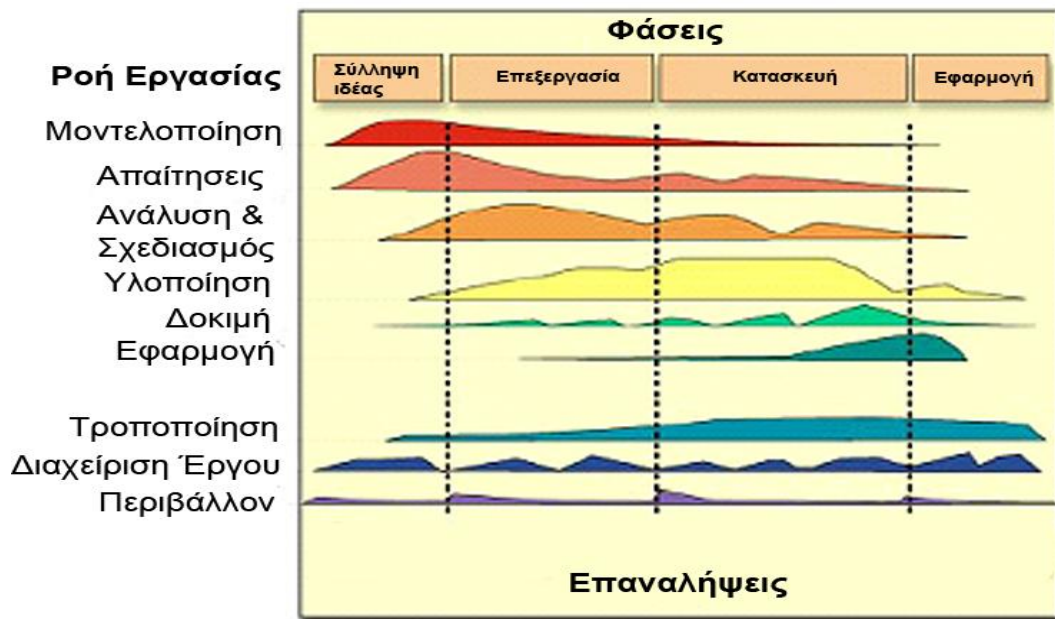
2.3.1 Η ΜΕΘΟΔΟΛΟΓΙΑ RATIONAL UNIFIED PROCESS (RUP)

Η μεθοδολογία RUP αναπτύχθηκε με σκοπό να αποφύγουμε όσο το δυνατόν περισσότερο τα σφάλματα που μπορεί να συμβούν στην ανάπτυξη ενός πληροφοριακού συστήματος, κάτι που μπορεί να επιβαρύνει τόσο το συνολικό κόστος της εφαρμογής, όσο και το χρόνο παράδοσής της. Όλα τα συστήματα απαιτείται να έχουν και μια σταθερή αρχιτεκτονική. Επίσης με τη RUP έχουμε τη δυνατότητα να παρακολουθήσουμε το στάδιο ανάπτυξης της εφαρμογής και να είμαστε εντός χρονοπρογραμματισμού. Με τη μεθοδολογία RUP η διοίκηση ενός οργανισμού θεωρεί ότι έχουμε ευκολότερη διαχείριση των πόρων και των διαδικασιών. Οι ενέργειες που πρέπει να γίνουν είναι γνωστές σε όλους, όπως επίσης και ο τρόπος με τον οποίο πρέπει να γίνουν. Οι προγραμματιστές από τη δική τους πλευρά θεωρούν ότι η συγκεκριμένη μεθοδολογία είναι βαρετή, σπατάλη χρόνου και καθυστερεί την ανάπτυξη ενός συστήματος. Πιο συγκεκριμένα τα κύρια χαρακτηριστικά της RUP είναι τα εξής: [Dan Rawsthorne, “Comparing/Combining RUP, XP, and Scrum – mixing the Process Cocktail”, 2003, p 15]

- Καθοδηγούμενη από τις περιπτώσεις χρήσεις (use cases).
- Είναι architecture centric μεθοδολογία.
- Κάνει χρήση της UML.
- Μπορεί να προσαρμοστεί στις ανάγκες ενός έργου.
- Αναπτύσσει το λογισμικό σε μια σειρά επαναλήψεων.
- Διαχειρίζεται τις απαιτήσεις.
- Βεβαιώνει την ποιότητα του συστήματος.
- Ελέγχει τις αλλαγές του λογισμικού.

Στη RUP προσέγγιση ο κύκλος ζωής ενός συστήματος αποτελείται από κύκλους, τον κύκλο ανάπτυξης, βελτίωσης και συντήρησης. Οι φάσεις που μπορεί να διακρίνει κανείς είναι οι φάσεις της έναρξης, επεξεργασίας, κατασκευής και εφαρμογής του συστήματος. Το έργο υλοποιείται μετά από μια σειρά επαναλήψεων οι οποίες λαμβάνουν χώρα κατά τη διάρκεια της ανάλυσης, του καθορισμού των απαιτήσεων,

του σχεδιασμού, της υλοποίησης και της δοκιμής, όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 6: Κύκλος ζωής RUP προσέγγισης [Dan Rawsthorne, "Comparing/Combining RUP, XP, and Scrum – mixing the Process Cocktail", 2003, p 16]

Η κύρια πρακτική της μεθοδολογίας είναι ότι τα περισσότερα επαφίενται στον developer της εφαρμογής, ωστόσο υπάρχουν πολλά UML διαγράμματα τα οποία απεικονίζουν τις δραστηριότητες και τη σειρά με την οποία αυτές θα πρέπει να εκτελεστούν. Μερικοί από τους κύριους ρόλους στη μεθοδολογία RUP είναι οι επόμενοι: [Dan Rawsthorne, "Comparing/Combining RUP, XP, and Scrum – mixing the Process Cocktail", 2003, p 21]

- Project Manager
- Database Designer
- User Interface Designer
- System Analyst
- Architect
- Implementator
- Code Reviewer
- Test Designer

- Integration Tester
- Performance Tester

Οι δραστηριότητες στη RUP ανάλογα με τη φάση ανάπτυξης του συστήματος απεικονίζονται στον πίνακα που ακολουθεί.

	Έναρξη	Επεξεργασία	Κατασκευή
<i>Project Management</i>			
Καθορισμός κινδύνων	*****	*	*
Ανάπτυξη πλάνου	*****	*	*
Ανάπτυξη πλάνου επαναλήψεων	*	*****	*****
Εκτίμηση επαναλήψεων	*	***	*****
<i>Business Modeling</i>			
Καταγραφή UC	*****		
Εύρεση actors και UC	*****		
Περιγραφή UC	*****	*	
Επισκόπηση UC	*****	*	
<i>Requirements</i>			
Χειρισμός εξαρτήσεων	**	*****	
Προτεραιότητα UC	***	***	
Λεπτομέρειες στις UC	**	***	**
Μοντελοποίηση UI	**	***	*
Ανασκόπηση απαιτήσεων	***	***	***
<i>Analysis and Design</i>			
Ανάλυση UC	*	*****	***

Σχεδιασμός Κλάσεων		*	*****
Σχεδιασμός Β.Δ.		*	*****
Σχεδιασμός UC		**	*****
Ανασκόπηση σχεδιασμού		*	*****
<i>Implementation</i>			
Μοντέλο υλοποίησης		*	*****
Ολοκλήρωση συστήματος		*	*****
Διόρθωση Ελαττωμάτων			*****
Ανασκόπηση κώδικα			*****
<i>Test</i>			
Σχεδιασμός δοκιμής		*	*****
Εκτέλεση δοκιμής			*****
Αξιολόγηση δοκιμής			*****

Πίνακας 3: Δραστηριότητες και φάσεις ανάπτυξης RUP [Dan Rawsthorne (2003)
Comparing/Combining RUP, XP, and Scrum – mixing the Process Cocktail p 18-20]

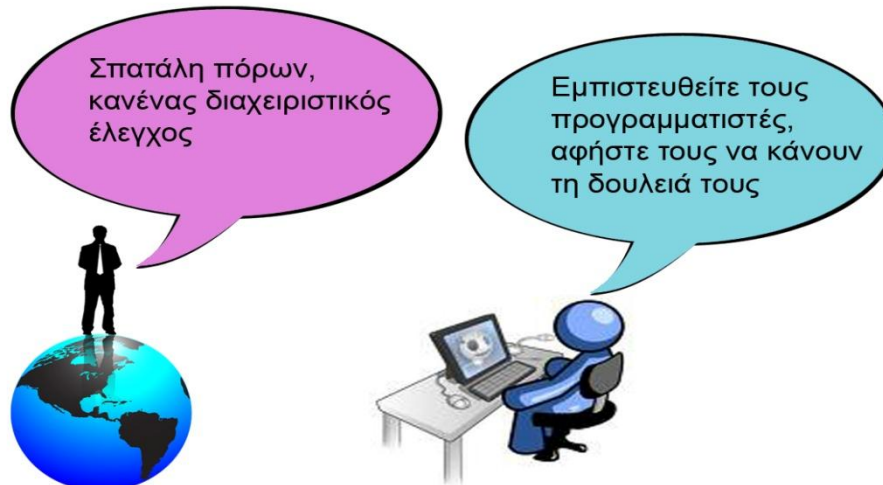
Όπως προαναφέραμε ο παραπάνω πίνακας αναπαριστά τις δραστηριότητες στη RUP ανάλογα με τη φάση ανάπτυξης του συστήματος. Όταν για παράδειγμα βρισκόμαστε στη φάση του business modeling δίνουμε μεγαλύτερη έμφαση στην καταγραφή των περιπτώσεων χρήσης πριν το σύστημα αρχίσει να υλοποιείται. Αντίστοιχα, όταν βρισκόμαστε στο δοκιμαστικό στάδιο της εφαρμογής περισσότερη προσοχή δίνουμε στην εκτέλεση της δοκιμής όταν η εφαρμογή έχει ξεκινήσει να υλοποιείται. Επομένως οι αστερίσκοι σε κάθε στήλη του πίνακα υποδηλώνουν τη βαρύτητα της κάθε δραστηριότητας ανάλογα με τη φάση ανάπτυξης στην οποία βρισκόμαστε.

Τον κυρίαρχο ρόλο στην προσέγγιση του eXtreme Programming (XP) έχουν οι developers του συστήματος. Όσο πιο καλογραμμένος είναι ο κώδικας της εφαρμογής, τόσο πιο εύκολα μπορούν να γίνουν διορθώσεις και αλλαγές σε περίπτωση που απαιτηθεί. Οι πελάτες είναι αυτοί που διαμορφώνουν τις επιχειρηματικές αποφάσεις και οι προγραμματιστές αναλαμβάνουν εξ' ολοκλήρου το τεχνικό κομμάτι. Οι επαναλήψεις είναι μικρές και σύντομες έτσι ώστε ο πελάτης να έχει γρήγορη ανατροφοδότηση των ολοκληρωμένων τμημάτων του συστήματος. Όπως και στις υπόλοιπες μεθοδολογίες, έτσι και στη μεθοδολογία XP η διαδικασία ξεκινάει από τον καθορισμό των απαιτήσεων. Στη συνέχεια έχουμε τη φάση του σχεδιασμού και των επαναλήψεων, το στάδιο της υλοποίησης και τέλος τη φάση της δοκιμής. Οι κύριοι ρόλοι στο eXtreme Programming είναι ο προγραμματιστής, ο πελάτης, ο manager και ο tester της εφαρμογής.

Μια περίπτωση χρήσης στη μεθοδολογία XP περιγράφει τον τρόπο με τον οποίο ένας δράστης αλληλεπιδρά με το σύστημα για την επίτευξη ενός στόχου και περιγράφει τη συνολική διαδικασία. Οι απαιτήσεις ενός συστήματος προσθέτουν επιπλέον λειτουργικότητα στην επιχειρησιακή διαδικασία που θα υλοποιηθεί μέσα στο επόμενο χρονικό διάστημα. Οι απαιτήσεις επικεντρώνονται στην υλοποίηση και στη μετέπειτα παραγωγή της εφαρμογής και περιγράφουν ένα μέρος της συνολικής διαδικασίας.

Η διοίκηση ενός οργανισμού θεωρεί την προσέγγιση του eXtreme Programming ως μια μεθοδολογία από την οποία απουσιάζει ο διαχειριστικός έλεγχος, είναι πολύ πιθανό για το λόγο αυτό να κατασπαταλώνται πόροι κάτι που έχει άμεσες οικονομικές συνέπειες για την επιχείρηση. Επίσης επειδή στο extreme programming δεν δίνεται ιδιαίτερη έμφαση στη φάση της ανάλυσης και του σχεδιασμού (κυρίαρχο ρόλο έχει η φάση της υλοποίησης και ο ρόλος του προγραμματιστή, όπως φαίνεται και στην παρακάτω εικόνα), έχει παρατηρηθεί καθυστερημένη παράδοση των πληροφοριακών συστημάτων τα οποία βασίστηκαν στη συγκεκριμένη μεθοδολογία, καθώς επίσης και παράδοση συστημάτων που δεν ανταποκρίνονταν

πλήρως στις απαιτήσεις του πελάτη. Η προσέγγιση του eXtreme Programming μπορεί παρόλα αυτά να εφαρμοστεί με επιτυχία σε πληροφοριακά συστήματα στα οποία δεν υπάρχει μεγάλη πολυπλοκότητα διαδικασιών και συγκεκριμένα μπορεί να αποδειχθεί και πολύ πιο γρήγορη στην παράδοση συστημάτων μικρής κλίμακας.



Εικόνα 7: eXtreme Programming [Dan Rawsthorne, "Comparing/Combining RUP, XP, and Scrum – mixing the Process Cocktail", 2003, p 24]

2.4 ΠΕΡΙΠΤΩΣΕΙΣ ΧΡΗΣΗΣ (USE CASES) ΚΑΙ ΑΠΑΙΤΗΣΕΙΣ ΧΡΗΣΤΩΝ

Οι περιπτώσεις χρήσης είναι μια τεχνική που εξετάζει ένα ή περισσότερα σενάρια από την πλευρά του χρήστη στην προσπάθειά του να ολοκληρώσει ένα συγκεκριμένο στόχο. Πιο συγκεκριμένα:

- Λαμβάνει χώρα σε ένα συγκεκριμένο και σχεδιασμένο σκοπό.
- Εφαρμόζεται ώστε να επιτευχθεί ένας στόχος αφού ολοκληρωθεί κάποιο σενάριο.
- Παρέχει εγγυήσεις για την ολοκλήρωση ενός σεναρίου και προσδιορίζει κάποια επιπλέον σενάρια σε περίπτωση τυχόν παρεκκλίσεων.

Στον πίνακα που ακολουθεί το όνομα της περίπτωσης χρήσης προσδιορίζει το λόγο ύπαρξής της, το κύριο σενάριο της επιτυχίας θα πρέπει να περιλαμβάνει τουλάχιστον δύο βήματα αλλά όχι πολύ περισσότερα. Ο σκοπός και το επίπεδο του

στόχου είναι πολύ σημαντικά και θα πρέπει να αναπαριστώνται με εικονίδια για γρήγορη αναζήτηση όταν χρειάζεται. Ο αρχικός actor είναι κάποιος από τους προαναφερθέντες χρήστες, γεφυρώνοντας το χάσμα μεταξύ των απαιτήσεων και των περιπτώσεων χρήσης. Το πλαίσιο τεχνολογία και παραλλαγές δεδομένων χρησιμοποιείται για να περιγράψει κάποια στοιχεία τα οποία μπορεί να διαφέρουν, ενώ οι επεκτάσεις προσδιορίζουν το τι είναι διαφορετικό σε σχέση με το αρχικό σενάριο. Αναφέρονται σε συγκεκριμένα βήματα του αρχικού σεναρίου και συγκεκριμένα περισσότερα από τέσσερα βήματα σε βάθος προσδιορίζουν την ύπαρξη μιας άλλης περίπτωσης χρήσης. [Jamie Allsop, “Foundations of agile development”, 2008, p 40]

Περίπτωση χρήσης xxx	Σκοπός σχεδίασης	Τίτλος Περίπτωσης χρήσης	Επίπεδο σκοπού
Σκοπός			
Επίπεδο			
Κύριος δράστης			
Ενδιαφερόμενα μέρη			
Συνθήκες			
Εγγυήσεις			
Κύριο σενάριο			
1			
n			
Επεκτάσεις			
1a			
Na			
Τεχνολογία και παραλλαγές δεδομένων			
1			
n			
Σχόλια			

Πίνακας 4: Παράδειγμα περίπτωσης χρήσης [Jamie Allsop (2008), Foundations of Agile Development

Στις περιπτώσεις χρήσεις έμφαση δίνεται κυρίως στο τι επιτυγχάνεται και όχι στο πώς επιτυγχάνεται, ο τρόπος με τον οποίο πετυχαίνουμε κάτι περιγράφεται στην τεκμηρίωση του χρήστη. Οι use cases είναι ιδιαίτερα χρήσιμες όταν προσπαθούμε να περιγράψουμε περίπλοκες αλληλεπιδράσεις μεταξύ των χρηστών.

Ποια όμως είναι πραγματικά η σχέση μεταξύ μιας περίπτωσης χρήσης και της περιγραφής ενός χρήστη; Η αλήθεια είναι ότι οι περιπτώσεις χρήσεις βασίζονται στις απαιτήσεις των χρηστών. Πιο συγκεκριμένα μια περίπτωση χρήσης περιγράφει ένα στόχο καθοδηγούμενο από κάποιο λειτουργικό σενάριο (functional scenario), καθορίζει ένα ή περισσότερα σενάρια επιτυχίας, και παρέχει το γενικό πλαίσιο εντός του οποίου θεσπίζονται οι απαιτήσεις των χρηστών. Αντίθετα οι απαιτήσεις προσδιορίζουν κάποια ενέργεια (βήμα) που πρέπει να γίνει μέσα σε ένα σενάριο. Οι απαιτήσεις μπορεί επίσης να καθορίζουν ένα μη λειτουργικό παράγοντα ενός συστήματος από την οπτική του χρήστη. Όπως καταλαβαίνουμε οι περιπτώσεις χρήσης και οι περιγραφές των χρηστών είναι στενά συνδεδεμένες. Μια περίπτωση χρήσης μπορεί να παρέχει το περιεχόμενο για αρκετές απαιτήσεις χρηστών, ενώ οι απαιτήσεις μπορούν να συνεπάγονται αρκετές περιπτώσεις χρήσης.

Οι λειτουργικές απαιτήσεις τυπικά περιγράφουν καθήκοντα τα οποία ένας χρήστης μπορεί να πραγματοποιήσει, ενώ οι μη λειτουργικές περιγράφουν μελλοντικές καταστάσεις που ένας χρήστης μπορεί να αντιμετωπίσει (πρόκειται δηλαδή για ένα τρόπο σύλληψης ιδεών σχετικά με τους περιορισμούς και τις μεταβλητές του συστήματος). Ιδιαίτερη έμφαση δίνεται στο ποιος χρήστης θα επωφεληθεί από τις μη λειτουργικές απαιτήσεις, ή ποιος χρήστης μπορεί να επηρεαστεί περισσότερο από μια μεταβλητή.

2.5 ΓΙΑΤΙ ΠΡΕΠΕΙ ΝΑ ΠΡΟΤΙΜΟΥΝΤΑΙ ΟΙ ΕΥΕΛΙΚΤΕΣ ΜΕΘΟΔΟΛΟΓΙΕΣ

Στις κοινές μεθοδολογίες ανάπτυξης συστημάτων οι τεχνικές που χρησιμοποιούνται για τον υπολογισμό και την καταγραφή της προόδου εφαρμόζονται πιστά σε κάθε στάδιο του κύκλου ανάπτυξης και απαιτείται αρκετή προσπάθεια για να

δημιουργηθούν πίνακες που απεικονίζουν την πρόοδο των εργασιών και την ενημέρωση των πινάκων. Πολλές φορές ο φόρτος είναι τόσο μεγάλος που ένα άτομο στην ομάδα καταλήγει να κάνει αποκλειστικά αυτή τη δουλειά. Αν όλα τα παραπάνω εμπόδια ξεπεραστούν, η ακρίβεια των δεδομένων παραμένει σε χαμηλά επίπεδα και τις περισσότερες φορές βασίζεται στη διαίσθηση. Στις ευέλικτες μεθοδολογίες ανάπτυξης συστημάτων οι παραπάνω υπολογισμοί και καταγραφές παράγονται αυτόματα ως αποτέλεσμα της εφαρμογής της μεθοδολογίας. Τα παραγόμενα δεδομένα είναι όσο πιο ακριβή γίνεται καθώς η εξέλιξη της προόδου βασίζεται σε πραγματικά στοιχεία με βάση την πρόοδο που μέχρι τώρα έχει επιτευχθεί. Ο υπολογισμός του χρόνου ολοκλήρωσης γίνεται με βάση την πρόοδο των εργασιών και όχι με βάση τις προβλέψεις των developers της εφαρμογής, οι οποίοι δεν μπορούν να παρέχουν σωστές εκτιμήσεις για το χρόνο του υπολειπόμενου έργου. Σημεία της εφαρμογής που θεωρούνται δύσκολα να υλοποιηθούν διακρίνονται εύκολα μιας και παρουσιάζουν πολύπλοκες απαιτήσεις. Επομένως το ρίσκο μπορεί να αντιμετωπιστεί χωρίζοντας τις απαιτήσεις σε περισσότερες και μικρότερες εργασίες που είναι πιο εύκολο να υλοποιηθούν. Επίσης η χρήση περιγραφών των χρηστών παρέχει μια ανεπίσημη μέθοδο καταγραφής των λειτουργικών και μη λειτουργικών απαιτήσεων. Με τον τρόπο αυτό είναι ευκολότερο για τον ιδιοκτήτη της εφαρμογής να ταξινομήσει τις επιχειρησιακές ανάγκες.

Οι ευέλικτες μεθοδολογίες είναι πετυχημένες για τους ακόλουθους λόγους:

- Χρησιμοποιούν επαναλήψεις σταθερού μήκους.
- Η διάρκειά των επαναλήψεων είναι μέχρι και τρεις εβδομάδες ώστε να αποφεύγονται μακροσκελής εργασίες και το έργο να παραδίδεται εντός του χρονικού ορίου που έχει αρχικά προβλεφθεί.
- Η αξιολόγηση για το σχεδιασμό και την υλοποίηση των εργασιών γίνεται σε κάθε μία από τις επαναλήψεις ξεχωριστά.
- Τα καθήκοντα εντός των επαναλήψεων είναι σαφώς καθορισμένα και ολοκληρώνονται πάντοτε on time.

- Χρησιμοποιούνε μηχανισμούς που καθορίζουν πότε οι υποσχόμενες απαιτήσεις θα έχουν ολοκληρωθεί.

Τα συστήματα αναπτύσσονται σε ένα ιδιαίτερα πολύπλοκο περιβάλλον, ωστόσο απαιτείται η μέγιστη ευελιξία και ο κατάλληλος έλεγχος. Η πρόοδος ευνοεί εκείνους που εκτίθενται στις αλλαγές με όσο το δυνατόν μεγαλύτερη ευελιξία και δεν ευνοεί όσους δεν επιδέχονται τις νέες καταστάσεις και δεν έχουν βρει τους κατάλληλους μηχανισμούς για να ενσωματώσουν την καινοτομία στις εφαρμογές τους. Έτσι απαιτείται μια μεθοδολογία που διευκολύνει την ομάδα ανάπτυξης ενός project να λειτουργήσει εύρυθμα σε ένα περίπλοκο περιβάλλον χρησιμοποιώντας ακριβείς και καλά καθορισμένες διαδικασίες.

Η Scrum είναι μια μεθοδολογία διαχείρισης, βελτίωσης και συντήρησης ενός υπάρχοντος ή ενός νέου συστήματος. Οι διάφορες εκδόσεις λογισμικού σχεδιάζονται με βάση τις παρακάτω παραμέτρους.

- Τις απαιτήσεις του πελάτη.
- Το χρονοδιάγραμμα παράδοσης του έργου.
- Τον ανταγωνισμό.
- Την παρεχόμενη ποιότητα.
- Τους πόρους

Οι παράμετροι αυτές αποτελούν το πλάνο του αρχικού σχεδιασμού για τη δημιουργία ή τη βελτίωση του προϊόντος, όμως μπορεί να αλλάξουν κατά την ανάπτυξη του συστήματος. Μια πετυχημένη μεθοδολογία θα πρέπει να λάβει υπόψη της αυτές τις παραμέτρους καθώς επίσης και όποιες αλλαγές μπορεί να προκύψουν στις παραμέτρους αυτές.

Η σωστή μεθοδολογία μπορεί να είναι ο σημαντικότερος παράγοντας για τον καθορισμό της επιτυχίας και πιο συγκεκριμένα μεθοδολογίες που ενθαρρύνουν την ευελιξία μέσα σε ένα οργανισμό έχουν μεγαλύτερο βαθμό ανοχής στις αλλαγές και τις νέες εξελίξεις. Παρόλο που μπορεί να θεωρήσει κάποιος ότι οι ευέλικτες

μεθοδολογίες είναι απρόβλεπτες λόγω των αλλαγών που επιτρέπουν στην ανάπτυξη μιας εφαρμογής, ενσωματώνουν παρόλο αυτά τεχνικές διαχείρισης της μη προβλεψιμότητας. Η βασική διαφορά της Scrum με άλλες μεθοδολογίες είναι ότι θεωρεί ότι υπάρχει κάποιος βαθμός μη προβλεψιμότητας στη φάση της ανάλυσης, του σχεδιασμού και της υλοποίησης. Χρησιμοποιεί όμως ένα μηχανισμό για τη διαχείριση των απρόβλεπτων καταστάσεων και γι' αυτό η ευελιξία, η ανταπόκριση και η αξιοπιστία είναι τα αποτελέσματα από τη χρήση της μεθόδου.

Η Scrum είναι μια ευέλικτη μεθοδολογία που χρησιμοποιείται συχνά για την ανάπτυξη ενός προϊόντος και κυρίως την ανάπτυξη λογισμικού. Η Scrum παρέχει ένα πλαίσιο διαχείρισης γενικού σκοπού και μπορεί να εφαρμοστεί σε εφαρμογές με πιεστικές ημερομηνίες παράδοσης, με σύνθετες απαιτήσεις και συγκεκριμένες ιδιαιτερότητες. Είναι μια ευέλικτη προσέγγιση στην ανάπτυξη λογισμικού, ένα γενικό πλαίσιο και όχι μια πλήρης διαδικασία ή μεθοδολογία. Έτσι αντί να παρέχουμε ολοκληρωμένες και λεπτομερείς περιγραφές για το πώς θα πρέπει να γίνουν τα πάντα σε ένα έργο, το μεγαλύτερο βάρος ανατίθεται στην ομάδα ανάπτυξης του συστήματος. Αυτό γίνεται γιατί η ομάδα γνωρίζει καλύτερα πώς να επιλύσει το πρόβλημα που παρουσιάστηκε. Αυτός είναι ο λόγος για παράδειγμα που μία συνάντηση σχεδίασης (planning meeting) περιγράφεται με βάση το επιθυμητό αποτέλεσμα (μια δέσμευση για να τεθούν τα χαρακτηριστικά που πρέπει να αναπτυχθούν μέχρι την επόμενη συνάντηση) και όχι ένα σύνολο εισαγωγικών κριτηρίων, καθορισμοί καθηκόντων, κριτήρια επικύρωσης και κριτήρια εξόδου όπως θα παρέχονταν στις περισσότερες μεθοδολογίες.

[<http://www.mountangoatsoftware.com/topics/scrum>]

Η Scrum στηρίζεται σε μια αυτό-οργανωτική και διατμηματική ομάδα, στην οποία δεν υπάρχει ένας γενικός ηγέτης που επιλέγει το άτομο που θα εργαστεί σε συγκεκριμένο έργο ή που θα αποφασίσει για το πώς θα λυθεί ένα πιθανό πρόβλημα. Όλα αυτά είναι ζητήματα που αποφασίζονται από την ομάδα ως σύνολο. Όπως αναφέραμε η ομάδα είναι διατμηματική έτσι ώστε ο καθένας που θα εμπλακεί να πρέπει να αναλάβει αρμοδιότητες από τη σύλληψη της ιδέας μέχρι και την εφαρμογή της. [<http://www.mountangoatsoftware.com/topics/scrum>]

Αυτές οι ευέλικτες ομάδες ανάπτυξης υποστηρίζονται από δύο συγκεκριμένα άτομα, τον Scrum Master και τον ιδιοκτήτη του προϊόντος. Ο Scrum Master μπορεί να θεωρηθεί ως ο προπονητής για την ομάδα, βοηθώντας τα μέλη της να χρησιμοποιήσουν το Scrum πλαίσιο έτσι ώστε να αποδώσουν στο υψηλότερο δυνατό επίπεδο. Ο ιδιοκτήτης του προϊόντος αντιπροσωπεύει την επιχείρηση, τους πελάτες ή τους χρήστες και καθοδηγεί την ομάδα προς το «χτίσιμο» της σωστής εφαρμογής. [<http://www.mountangoatsoftware.com/topics/scrum>]

Τα Scrum projects σημειώνουν πρόοδο των εργασιών σε μια σειρά από επαναλήψεις βασιζόμενες στο χρόνο, που ονομάζονται sprints. Ο χρόνος οριοθετείται σε κύβους και τα sprints επαναλαμβάνονται σε χρονικό διάστημα όχι μεγαλύτερο από ένα μήνα. Κατά την έναρξη ενός sprint, τα μέλη της ομάδας δεσμεύονται να παρέχουν κάποια χαρακτηριστικά που αναφέρονται στο ανεκτέλεστο υπόλοιπο των τμημάτων του έργου. Στο τέλος του sprint, αυτά τα χαρακτηριστικά ολοκληρώνονται, κωδικοποιούνται, δοκιμάζονται και ενσωματώνονται στο υπό εξέλιξη προϊόν ή σύστημα. Στο τέλος του sprint, γίνεται μια επανεξέταση κατά την οποία η ομάδα παρουσιάζει την πρόοδο στον ιδιοκτήτη του προϊόντος καθώς επίσης και στα υπόλοιπα ενδιαφερόμενα μέρη που παρέχουν ανατροφοδότηση και θα μπορούσαν να επηρεάσουν το επόμενο sprint.

[<http://www.mountangoatsoftware.com/topics/scrum>]

Με λίγα λόγια τα βασικά χαρακτηριστικά της διαδικασίας Scrum είναι τα ακόλουθα:

- Η Scrum είναι μια ευέλικτη μεθοδολογία που μας επιτρέπει να επικεντρωθούμε στην παροχή της υψηλότερης επιχειρηματικής αξίας στο συντομότερο χρονικό διάστημα.
- Το προϊόν αναπτύσσεται σταδιακά σε μια σειρά από sprints.
- Οι απαιτήσεις καταγράφονται ως στοιχεία σε μια λίστα του «ανεκτέλεστων προϊόντος» (product backlog).
- Οι ομάδες αυτό-οργανώνονται για να καθορίσουν τον καλύτερο τρόπο παράδοσης των υψηλότερης προτεραιότητας χαρακτηριστικών.

- Χρησιμοποιεί ένα σύνολο πρακτικών για να δημιουργήσει ένα ευέλικτο περιβάλλον παράδοσης των έργων.
- Δεν υπάρχουν προβλεπόμενες ειδικές πρακτικές ανάπτυξης.
- Κάθε δύο εβδομάδες έως ένα μήνα ο καθένας μπορεί να δει μια πραγματική έκδοση λειτουργικού λογισμικού και να αποφασίσει αν θα την αποδεχθεί, όπως είναι ή αν θα συνεχίσει τη βελτίωσή της για μια ακόμη περίοδο sprint.
- Οι φάσεις του σχεδιασμού και της ολοκλήρωσης αποτελούνται από διαδικασίες, εισόδους και εξόδους οι οποίες είναι καλά καθορισμένες. Η γνώση για τον τρόπο που εκτελούνται οι διαδικασίες είναι σαφής και η ροή γραμμική.
- Το sprint είναι μια εμπειρική διαδικασία στη διάρκεια του οποίου πολλά καθήκοντα είναι απροσδιόριστα και μη ελεγχόμενα. Αντίστοιχα έλεγχοι που περιλαμβάνουν τη διαχείριση επικινδυνότητας εφαρμόζονται σε κάθε επανάληψη του sprint ώστε να αποφύγουμε δυσμενής καταστάσεις και μεγιστοποιώντας την ευελιξία.
- Τα sprints είναι μη γραμμικά και ευέλικτα. Όπου είναι εφικτό χρησιμοποιείται ήδη αποκτημένη γνώση, ενώ σε άλλες περιπτώσεις εφαρμόζονται δοκιμαστικές εκδόσεις της εφαρμογής ώστε να αποκτηθεί καινούργια γνώση. Τα sprints χρησιμοποιούνται για να εξελίξουν το τελικό προϊόν.
- Τα τελικό προϊόν μπορεί να αλλάξει οποιαδήποτε στιγμή κατά τη φάση του σχεδιασμού και κατά τη διάρκεια του sprint. Το σύστημα παραμένει ανοιχτό στην πολυπλοκότητα του περιβάλλοντος, του ανταγωνισμού, του χρονικού και οικονομικού περιορισμού.

3.1 ΙΣΤΟΡΙΚΗ ΕΞΕΛΙΞΗ ΤΗΣ SCRUM – ΕΤΑΙΡΙΕΣ ΠΟΥ ΒΑΣΙΣΤΗΚΑΝ ΠΑΝΩ ΤΗΣ

Από τους χιλιάδες ανθρώπους που συνέβαλαν στη Scrum, θα πρέπει να ξεχωρίσουμε εκείνους που έπαιξαν καθοριστικό ρόλο στα πρώτα δέκα χρόνια. Πρώτος ήταν ο Jeff Sutherland, σε συνεργασία με τον Jeff McKenna, και ο Ken Schwaber με τον Mike Smith και τον Chris Martin. Η Scrum για πρώτη

φορά παρουσιάστηκε επισήμως και δημοσιεύθηκε στην OOPSLA το 1995. Κατά τη διάρκεια των επόμενων ετών ο Mike Beedle και ο Martine Devos έκαναν αρκετές συνεισφορές. Στη συνέχεια υπήρξαν πολλοί άλλοι, χωρίς τη βοήθεια των οποίων η Scrum δεν θα είχε εξελιχθεί όπως είναι σήμερα. Η Scrum έχει χρησιμοποιηθεί κατά καιρούς για:

- Εμπορικό λογισμικό
- Εσωτερική ανάπτυξη
- Οικονομικές εφαρμογές
- Εφαρμογές πιστοποιημένες με ISO 9001
- Ανάπτυξη παιχνιδιών βίντεο
- Συστήματα κρίσιμα για την ζωή, εγκεκριμένα από τον οργανισμό τροφίμων και φαρμάκων FDA
- Λογισμικό ελέγχου δορυφόρων
- Διαδικτυακούς τόπους
- Λογισμικό φορητών μονάδων
- Κινητά τηλέφωνα
- Εφαρμογές δικτυακής δρομολόγησης

[<http://www.mountangoatsoftware.com/scrum/a-reusable-scrum-presentation>]

Την καθοριστική σημασία της Scrum συνειδητοποίησαν από νωρίς αρκετές γνωστές εταιρίες οι οποίες την ενσωμάτωσαν κατά τη διαδικασία παραγωγής τους για την ανάπτυξη νέων προϊόντων και εφαρμογών τους. Μερικές από τις εταιρίες αυτές παρουσιάζονται ενδεικτικά παρακάτω.

- Microsoft
- Yahoo
- Google
- Electronic Arts
- Philips
- Siemens

- Nokia
- BBC
- John Deere

[<http://www.mountangoatsoftware.com/scrum/a-reusable-scrum-presentation>]

3.2 ΚΥΡΙΕΣ ΦΑΣΕΙΣ ΤΗΣ ΜΕΘΟΔΟΛΟΓΙΑΣ

Η μεθοδολογία Scrum αποτελείται από τρεις φάσεις, την pregame (περιλαμβάνει τις επιμέρους φάσεις του σχεδιασμού και της αρχιτεκτονικής του συστήματος), την game (περιλαμβάνει τη φάση της ανάπτυξης) και την postgame (περιλαμβάνει τη φάση της ολοκλήρωσης). Κατά τη διάρκεια του σχεδιασμού γίνεται μια εκτίμηση για το χρόνο ολοκλήρωσης και το συνολικό κόστος της εφαρμογής. Αν πρόκειται για την ανάπτυξη ενός νέου συστήματος, η φάση σχεδιασμού περιλαμβάνει επίσης και τη φάση της ανάλυσης. Όσο αφορά την αρχιτεκτονική του συστήματος, κατά τη διάρκειά της μελετάται το πώς θα υλοποιηθεί το ανεκτέλεστο υπόλοιπο του προϊόντος. Αυτή η φάση περιέχει επίσης και την τροποποίηση της αρχιτεκτονικής του συστήματος σε περίπτωση που χρειαστεί, καθώς επίσης και σχεδιασμό του συστήματος σε υψηλό επίπεδο (high level). Κατά τη διάρκεια της υλοποίησης του συστήματος θα πρέπει να ληφθούν υπόψη και να αντιμετωπιστούν διάφορες μεταβλητές όπως οι απαιτήσεις, η ποιότητα, το κόστος, ο ανταγωνισμός και ο χρόνος ολοκλήρωσης. Η αλληλεπίδραση με αυτές τις μεταβλητές καθορίζει και το τέλος της φάσης αυτής μετά από μια σειρά επαναλήψεων sprint που χρησιμοποιούνται για την εξέλιξη του συστήματος. Τέλος στη φάση της ολοκλήρωσης γίνεται η προετοιμασία για την παράδοση του συστήματος, την τεκμηρίωσή του και τη δοκιμαστική εφαρμογή του. Κάθε μία από τις παραπάνω φάσεις περιλαμβάνει τα ακόλουθα βήματα [Ken Schwaber, “Scrum Development Process”, p 12-14]:

A. Σχεδιασμός

- Ανάπτυξη μιας κατανοητής λίστας ανεκτέλεστου υπόλοιπου συστήματος.

- Προσδιορισμός της ημερομηνίας παράδοσης και της λειτουργικότητας της εφαρμογής.
- Προσδιορισμός των ομάδων του project.
- Προσδιορισμός κατάλληλων risk controls και αξιολόγηση της επικινδυνότητας.
- Ανασκόπηση και προσαρμογή των ανεκτέλεστων καθηκόντων.
- Επικύρωση και επιλογή του εργαλείου ανάπτυξης.
- Εκτίμηση του συνολικού κόστους.
- Επιβεβαίωση της έγκρισης της διοίκησης και της χρηματοδότησης.

B. Αρχιτεκτονική

- Ανασκόπηση στο ανεκτέλεστο υπόλοιπο που έχει ανατεθεί.
- Προσδιορισμός των απαραίτητων αλλαγών στο ανεκτέλεστο υπόλοιπο.
- Καθορισμός της κατάλληλης αρχιτεκτονικής που υποστηρίζει τις απαιτήσεις του συστήματος.
- Προσδιορισμός προβλημάτων υλοποίησης και εφαρμογή των απαραίτητων αλλαγών.
- Σχεδιασμός της συνάντησης ανασκόπησης.

Γ. Υλοποίηση

Η φάση της υλοποίησης είναι ένας επαναλαμβανόμενος κύκλος ενεργειών. Η διοίκηση του οργανισμού καθορίζει την ποιότητα και τη λειτουργικότητα της εφαρμογής και η ομάδα ανάπτυξης μετά από ένα αριθμό επαναλήψεων ολοκληρώνει το σύστημα. Τα βήματα της συγκεκριμένης φάσης είναι τα ακόλουθα:

- Συναντήσεις με τις υπόλοιπες ομάδες για την ανασκόπηση των τμημάτων της εφαρμογής που έχουν ολοκληρωθεί.
- Προσαρμογή των προτύπων με τα οποία το σύστημα θα πρέπει να συμφωνεί.
- Επαναλαμβανόμενα sprints μέχρι το προϊόν να ολοκληρωθεί και να είναι έτοιμο να διανεμηθεί.

Κάθε sprint αποτελείται από διάφορα στάδια ολοκλήρωσης. Το στάδιο της ανάπτυξης (development), κατά το οποίο καθορίζονται οι απαιτούμενες αλλαγές για την υλοποίηση των ανεκτέλεστων απαιτήσεων, την ανάλυση, τη δοκιμή και την τεκμηρίωση των αλλαγών. Το στάδιο του τυλίγματος (wrapping) όπου δημιουργούμε ένα εκτελέσιμο τμήμα της εφαρμογής, το στάδιο της ανασκόπησης κατά τη διάρκεια της οποίας όλες οι ομάδες συναντώνται και συζητούν για την πρόοδο που δημιουργήθηκε ενημερώνοντας το ανεκτέλεστο υπόλοιπο του συστήματος. Τέλος στο στάδιο της προσαρμογής συγκεντρώνουμε όλες τις πληροφορίες από τις συναντήσεις που πραγματοποιήθηκαν συμπεριλαμβανομένου και των αλλαγών και των νέων ιδιοτήτων που προέκυψαν μετά τη συνάντηση των ομάδων.

Δ. Ολοκλήρωση

Όταν η ομάδα θεωρήσει ότι οι μεταβλητές του χρόνου, του κόστους, του ανταγωνισμού, των απαιτήσεων και της ποιότητας έχουν ολοκληρωθεί, δηλαδή πληρούνται όλες οι προδιαγραφές για τη δημιουργία μιας νέας έκδοσης λογισμικού, τότε δηλώνουν την εφαρμογή «closed». Κατά τη διάρκεια αυτής της φάσης προετοιμάζεται το προϊόν που πρόκειται να παραδοθεί για την τελική του κυκλοφορία. Η φάση της ολοκλήρωσης περιλαμβάνει τα παρακάτω βήματα.

- Δοκιμή του συστήματος.
- Τεκμηρίωση.
- Εμπορική προετοιμασία του συστήματος.
- Διανομή.

3.3 ΚΥΡΙΕΣ ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ ΤΗΣ SCRUM

Το sprint είναι η κύρια αρμοδιότητα του Scrum project. Η Scrum είναι μια επαναληπτική και σταδιακή διαδικασία και έτσι το project χωρίζεται σε μια σειρά διαδοχικών sprints οριοθετημένα σε κύβους. Κάθε κύβος είναι συνήθως μεταξύ μιας εβδομάδας και ενός μήνα. Μια πρόσφατη έρευνα έδειξε ότι το πιο

διαδομένο μήκος sprint είναι δύο εβδομάδες. Κατά τη διάρκεια αυτής της περιόδου, η ομάδα κάνει τα πάντα για να πάρει ένα μικρό δείγμα από το σύνολο των χαρακτηριστικών της εφαρμογής (ιδέα, κωδικοποίηση, δοκιμή).

[<http://www.mountangoatsoftware.com/topics/scrum>]

Στη διάρκεια ενός sprint ο Scrum Master βεβαιώνεται ότι καμία αλλαγή και τίποτε δεν θα επηρεάσει το στόχο του sprint. Η σύνθεση της ομάδας και οι αρχικοί στόχοι παραμένουν σταθεροί σε όλη τη διάρκεια του sprint. Τα sprints αποτελούνται και περιέχουν τη συνάντηση σχεδιασμού (Sprint Planning Meeting), τον απολογισμό (Sprint Review), την ανακεφαλαίωση (Sprint Retrospective) και τις εργασίες υλοποίησης. Τα sprints εκτελούνται συνεχόμενα το ένα μετά το άλλο χωρίς να μεσολαβήσει κανένα χρονικό διάστημα μεταξύ τους.

Η πρώτη δραστηριότητα του sprint, είναι η συνάντηση σχεδιασμού. Κατά τη συνάντηση αυτή, ο ιδιοκτήτης και η ομάδα συζητούν για τα στοιχεία εκείνα που έχουν την υψηλότερη προτεραιότητα στη λίστα των ανεκτέλεστων λειτουργιών του έργου. Τα μέλη της ομάδας συνειδητοποιούν ποιες λειτουργίες μπορούν να εκτελεστούν άμεσα και ποια καθήκοντα είναι εκείνα που θα πρέπει να εκτελεστούν κατά τη διάρκεια του sprint.

Σε κάθε ημέρα του sprint, μια καθημερινή συνάντηση Scrum παρακολουθούν όλα τα μέλη της ομάδας, συμπεριλαμβανομένου και του Scrum Master και του ιδιοκτήτη του προϊόντος. Η συνάντηση αυτή χρονομετρείται σε κύβους και έχει διάρκεια έως και δεκαπέντε λεπτά. Κατά τη διάρκεια αυτή, τα μέλη της ομάδας μοιράζονται τα στοιχεία που εργάστηκαν στην προηγούμενη ημέρα, τα στοιχεία που θα εργαστούν σήμερα και εντοπίζουν τυχόν εμπόδια στην πρόοδο της διαδικασίας. Οι καθημερινές Scrum συναντήσεις χρησιμεύουν στο συγχρονισμό των ενεργειών των μελών της ομάδας καθώς και στη λεπτομερή ανάλυση του έργου του sprint.

[<http://www.mountangoatsoftware.com/topics/scrum>]

Στο τέλος της διάρκειας ενός sprint η ομάδα διεξάγει μια αξιολόγηση για το συγκεκριμένο sprint. Κατά την αξιολόγηση η ομάδα αποδεικνύει τη λειτουργικότητα

που προστέθηκε στην εφαρμογή κατά τη διάρκεια αυτού του sprint. Ο στόχος της αξιολόγησης είναι να πάρουμε τις ανάλογες πληροφορίες και την κατάλληλη ανατροφοδότηση από τον ιδιοκτήτη του προϊόντος ή από οποιουδήποτε άλλους χρήστες που μπορεί να προσκλήθηκαν για την αξιολόγηση. Η ανάδραση αυτή μπορεί να οδηγήσει σε αλλαγές στη λειτουργικότητα της εφαρμογής που παραδόθηκε πρόσφατα. Μπορεί όμως εξίσου να οδηγήσει σε αναθεώρηση ή σε προσθήκη στοιχείων στο ανεκτέλεστο τμήμα της εφαρμογής.

Μια άλλη δραστηριότητα που εκτελείται στο τέλος κάθε sprint είναι ο απολογισμός του. Στη συνάντηση αυτή συμμετέχουν επίσης όλα τα μέλη της ομάδας, συμπεριλαμβανομένου και του Scrum Master και του ιδιοκτήτη του προϊόντος. Η συνάντηση αυτή είναι μια ευκαιρία να ανασκοπήσουμε το sprint που τελειώνει και να προσδιορίσουμε τις ευκαιρίες ώστε να βελτιώσουμε το νέο sprint.

Κάθε σύστημα αποτελείται από ένα προσδιορισμό για το τι πρέπει να υλοποιηθεί, για τις εργασίες που πρέπει να ολοκληρωθούν με βάση ένα συγκεκριμένο πλάνο και ποιο θα πρέπει να είναι το τελικό παραδοτέο προϊόν. Κάθε εφαρμογή έχει έναν ορίζοντα βάση του οποίου μπορούμε να πούμε ότι οι εργασίες που έχουν ολοκληρωθεί είναι ανάλογες με το χρονικό σημείο στο οποίο βρισκόμαστε. Αν ο ορίζοντας είναι πολύ μεγάλος, τότε ο αρχικός προσδιορισμός μπορεί να αλλάξει, νέες μεταβλητές μπορεί να έχουν εμφανιστεί και ο βαθμός επικινδυνότητας μπορεί να είναι πολύ μεγάλος.

Τα sprints μπορούν να ακυρωθούν πριν τη λήξη ενός time box, αυτός που έχει τη δυνατότητα να ακυρώσει ένα sprint είναι ο ιδιοκτήτης του προϊόντος αλλά μπορεί να το κάνει υπό την πίεση της ομάδας ή του Scrum Master. Η περίπτωση για την οποία μπορεί να ακυρωθεί ένα sprint οφείλεται συνήθως στο ότι ο στόχος του είναι πλέον απαρχαιωμένος. Αυτό συμβαίνει κυρίως όταν ο οργανισμός αλλάξει κατεύθυνση ή η τεχνολογική κατάσταση και η αγορά μεταβληθούν σημαντικά. Με λίγα λόγια ένα sprint θα πρέπει να ακυρωθεί αν δεν έχει κανένα βαθμό χρησιμότητας σε σχέση με τις τρέχουσες καταστάσεις. Ωστόσο επειδή τα sprints

έχουν μικρή χρονική διάρκεια σπάνια συντρέχουν λόγοι για τους οποίους θα πρέπει να ακυρωθούν.

Όταν ένα sprint ακυρωθεί οποιεσδήποτε εργασίες είχαν ολοκληρωθεί θεωρούνται «χαμένες», το ανεκτέλεστο υπόλοιπο του προϊόντος θα πρέπει να ενημερωθεί και συγκεκριμένα οι νέες ανεκτέλεστες εργασίες θα πρέπει να ολοκληρωθούν στο χρονικό διάστημα που είχαν αρχικά σχεδιαστεί. Η ακύρωση των sprints προκαλεί κατανάλωση πόρων και είναι ιδιαίτερα επίπονες για τις ομάδες μιας και η συνάντηση σχεδιασμού του sprint θα πρέπει να ξεκινήσει από την αρχή.

Είναι σημαντικό για την ομάδα να είναι ενημερωμένη για το στάδιο στο οποίο βρίσκεται και τις εργασίες που εξελίσσονται, διαφορετικά τα μέλη της μπορεί να υποπέσουν σε λανθασμένες υποθέσεις για το τι πρέπει να υλοποιηθεί. Για το λόγο αυτό χρησιμοποιείται η sprint info page, όπως φαίνεται και στον παρακάτω πίνακα.

Ομάδα 3, Sprint 15

Στόχος του Sprint

Δοκιμαστική έκδοση

Ανεκτέλεστο Υπόλοιπο Sprint

Είσοδος χρήστη

Περιβάλλον διαχειριστή

Πρόγραμμα

Περίοδος Sprint: 11/06/2012 – 24/06/2012

Sprint Demo: 24/06/2012, 13:00

Καθημερινή Scrum: 9:30 – 9:45, στο γραφείο συνεδριάσεων

Μέλη Ομάδας

Σπύρος

Χρήστος

Μαρίνα

Ιωάννα

Χάρης

Τάσος

Πίνακας 5: Ενδεικτικές πληροφορίες τρέχοντος Sprint [Henrik Kniberg, “Scrum and XP from the Trenches”, 2006, p 26]

Το συντομότερο δυνατό και μετά τη συνάντηση σχεδιασμού sprint ο Scrum Master περιγράφει με συντομία τι θα εκτελεστεί στο επόμενο sprint και στη συνέχεια κοινοποιεί την περιγραφή αυτή σε όλα τα μέλη της ομάδας έτσι ώστε να είναι όλοι ενήμεροι για τις εργασίες που θα ολοκληρωθούν.

3.3.1 ΚΑΘΟΡΙΣΜΟΣ ΤΟΥ ΜΕΓΕΘΟΥΣ ΤΟΥ SPRINT

Ένα από τα αποτελέσματα της συνάντησης σχεδιασμού του sprint είναι και ο καθορισμός της ημερομηνίας ολοκλήρωσης ενός τμήματος της εφαρμογής. Αυτό σημαίνει ότι θα πρέπει πρώτα να καθορίσουμε το μήκος του sprint. Τα μικρά σε διάρκεια sprints μπορεί να είναι πιο ευέλικτα, δίνοντας τη δυνατότητα στην ομάδα να κινηθεί προς διαφορετικές κατευθύνσεις, παρέχουν γρηγορότερη ανατροφοδότηση, γίνονται περισσότερες παρουσιάσεις τμημάτων της εφαρμογής και συγκριτικά η ομάδα μπορεί να χάσει λιγότερο χρόνο κινούμενη προς τη λάθος κατεύθυνση. Όμως και τα μεγαλύτερα σε διάρκεια sprint έχουν τα πλεονεκτήματά τους. Η ομάδα έχει περισσότερο χρόνο να ανακάμψει από τυχόν προβλήματα και δυσκολίες και να επιτύχει το στόχο που έχει αρχικά οριστεί. Επίσης υπάρχει λιγότερη πίεση χρόνου σχετικά με την προετοιμασία που πρέπει να κάνει για τη συνάντηση σχεδιασμού ή το τελικό παραδοτέο του sprint [Henrik Kniberg, “Scrum and XP from the Trenches”, 2006, p 14].

Σε γενικές γραμμές μπορούμε να πούμε ότι οι ιδιοκτήτες των πληροφοριακών συστημάτων προτιμούν τα μικρά σε διάρκεια sprints ενώ οι developers των εφαρμογών τα μεγάλα. Έτσι θα πρέπει να καταλήξουμε σε μια συμβιβαστική λύση. Οι περισσότερες ομάδες επιλέγουν sprints μεταξύ 2 – 3 εβδομάδων που είναι ένα αρκετά καλό χρονικό διάστημα που μπορεί να μας παρέχει μια σχετική ευελιξία, ανατροφοδότηση και την ευκαιρία να ανακάμψουμε από τυχόν προβλήματα και δυσκολίες που μπορεί να εμφανιστούν. Γενικότερα είναι καλή πρακτική να πειραματιζόμαστε αρχικά με τη χρονική διάρκεια των sprints. Απλά επιλέγουμε ένα κατάλληλο χρονικό διάστημα για το sprint και το δοκιμάζουμε για τα επόμενα ένα ή δύο sprints. Σε περίπτωση που στη συνέχεια παρατηρήσουμε αποκλίσεις μεταξύ της

υπολογιζόμενης και της πραγματικής διάρκειας τότε μπορούμε να αλλάξουμε το χρονικό υπολογισμό για τα επόμενα sprints.

Έχοντας καταλήξει στο κατάλληλο χρονικό διάστημα του sprint θα πρέπει να μείνουμε αφοσιωμένοι σε αυτό για το υπόλοιπο της εφαρμογής μας. Σε ορισμένες περιπτώσεις μπορεί να μας φανεί λίγο πιεστικό ενώ σε άλλες λίγο μεγάλο. Διατηρώντας το ίδιο μέγεθος sprint, αρχίζει να παγιώνεται μέσα στην ομάδα ως το πιο δεδομένο χρονικό διάστημα. Καθώς εξελίσσεται η εφαρμογή όλοι το αποδέχονται και γνωρίζουν ότι θα πρέπει να παραδίδουν τα τμήματα των εφαρμογών εντός του χρονικού διαστήματος που έχει παγιωθεί.

3.3.2 ΚΑΘΟΡΙΣΜΟΣ ΤΟΥ ΣΤΟΧΟΥ ΤΟΥ SPRINT

Είναι μερικές φορές δύσκολο να ορίσουμε με σαφήνεια ένα καλά καθορισμένο στόχο για κάθε sprint. Ωστόσο είναι απαραίτητο γιατί μας βοηθάει να υλοποιήσουμε το ανεκτέλεστο υπόλοιπο της λίστας των εργασιών και να παραδώσουμε εντός του χρονικού διαστήματος το τμήμα της εφαρμογής για το οποίο έχουμε δεσμευτεί. Το σημαντικότερο είναι ότι ο στόχος του sprint θα πρέπει να δηλώνεται με επιχειρηματικούς όρους και όχι με τεχνικούς που σημαίνει ότι θα πρέπει να είναι κατανοητός και σε άτομα εκτός της ομάδας. Ο στόχος του sprint είναι εκείνος ο οποίος αιτιολογεί και το λόγο ύπαρξης του συγκεκριμένου sprint.

Ο στόχος θα πρέπει να ορίζει κάτι το οποίο δεν έχει προς το παρόν επιτευχθεί. Μπορεί στην αρχή του sprint να φαίνεται κάπως σκηνοθετημένος, αλλά αποδεικνύεται ιδιαίτερα χρήσιμος όταν οι διαδικασίες εξελίσσονται, γίνονται πιο πολύπλοκες και τα μέλη της ομάδας αρχίζουν να μπερδεύονται. Όταν υπάρχουν πολλές ομάδες που δουλεύουν σε διαφορετικά τμήματα της εφαρμογής όπως στις περισσότερες περιπτώσεις είναι απαραίτητο να υπάρχει μια λίστα στόχων κάθε έναν από τους οποίους θα πρέπει να υλοποιήσει και διαφορετική ομάδα. Με τον τρόπο αυτό τόσο οι ομάδες όσο και η διοίκηση του οργανισμού είναι σε θέση να γνωρίζουν το τι θα πρέπει να ολοκληρώσει η κάθε ομάδα και για ποιο λόγο. [Henrik Kniberg, “Scrum and XP from the Trenches”, 2006, p 15].

3.4 ΚΥΡΙΑ ΑΝΤΙΚΕΙΜΕΝΑ ΕΝΟΣ SCRUM PROJECT

Ο πρωταρχικός στόχος ενός Scrum Project είναι φυσικά το ίδιο το προϊόν. Η ομάδα αναμένεται να φέρει το προϊόν ή το σύστημα σε μια κατάσταση δυνητικά εμπορική στο τέλος κάθε sprint.

Το υπόλοιπο τμήμα της εφαρμογής που δεν έχει εκτελεστεί είναι ένας κατάλογος των λειτουργιών που απομένει να ολοκληρωθούν. Το ανεκτέλεστο τμήμα της εφαρμογής ταξινομείται με προτεραιότητα από τον ιδιοκτήτη του συστήματος, έτσι ώστε η ομάδα πάντοτε να αντιμετωπίζει πρώτα τα πιο κρίσιμα σημεία του. Ο πιο επιτυχημένος και δημοφιλής τρόπος για να δημιουργήσουμε μια λίστα μη ολοκληρωμένων ενεργειών είναι να συμπληρωθούν με σύντομες περιγραφές οι ενέργειες από τη σκοπιά του χρήστη ή του πελάτη.

Την πρώτη ημέρα του sprint και κατά τη διάρκεια της συνάντησης σχεδιασμού του sprint, τα μέλη της ομάδας δημιουργούν το ανεκτέλεστο υπόλοιπο sprint το οποίο μπορεί να θεωρηθεί ως η λίστα με το τι θα πρέπει να ολοκληρωθεί στο συγκεκριμένο sprint. Αντιθέτως το ανεκτέλεστο υπόλοιπο του προϊόντος είναι μια λίστα με τα χαρακτηριστικά που θα πρέπει να υλοποιηθούν (συχνά αναφέρονται στη φόρμα των περιγραφών των χρηστών). Το ανεκτέλεστο υπόλοιπο sprint είναι μια συγκεκριμένη λίστα με τα καθήκοντα της ομάδας που πρέπει να εκτελεστούν, ώστε να προσφέρουν την υποσχόμενη λειτουργικότητα στο τέλος της διάρκειας ενός sprint.

Δύο άλλα βασικά χαρακτηριστικά είναι το sprint burndown διάγραμμα και το διάγραμμα απολογισμού, τα οποία θα αναλύσουμε παρακάτω με περισσότερη λεπτομέρεια. Τα διαγράμματα δείχνουν τον όγκο των εργασιών που απομένουν και είναι πολύ αποτελεσματικά εργαλεία για τον προσδιορισμό με μια ματιά αν ένα sprint ή ο απολογισμός είναι εντός χρονοδιαγράμματος και έχουν τελειώσει όλες οι προγραμματισμένες εργασίες εντός της επιθυμητής ημερομηνίας.

Όταν έχουμε να αντιμετωπίσουμε κάποιους παράγοντες όπως η πολυπλοκότητα και η μη προβλεψιμότητα, θα πρέπει να πάρουμε μέτρα ώστε να αντιμετωπίσουμε τους κινδύνους που μπορεί να προκύψουν και να μειώσουμε σε όσο το δυνατόν μεγαλύτερο βαθμό το ρίσκο του συστήματός μας. Η Scrum έχει διάφορα μέτρα και ελέγχους για την αντιμετώπιση των δύσκολων καταστάσεων και είναι τα παρακάτω:

- **Ανεκτέλεστο υπόλοιπο:** Πρόκειται για λειτουργικές απαιτήσεις του συστήματος που δεν έχουν ακόμη ολοκληρωθεί. Το ανεκτέλεστο υπόλοιπο μπορεί επίσης να αναφέρεται σε σφάλματα (bugs) που πρέπει να διορθωθούν, ενέργειες βελτίωσης συγκεκριμένων τμημάτων της εφαρμογής που προέκυψαν μετά από αιτήματα των πελατών και τεχνολογικές αναβαθμίσεις προκειμένου να εξασφαλίσουμε ανταγωνιστικό πλεονέκτημα.
- **Εκδόσεις λογισμικού:** Τμήμα του συνολικού συστήματος που αναπαριστά μέχρι στιγμής μια βιώσιμη εφαρμογή βασιζόμενη στις απαιτήσεις που έχουν ολοκληρωθεί μέχρι στιγμής και του χρονικού διαστήματος που έχει παρέλθει.
- **Αλλαγές:** Οι αλλαγές πρέπει να υπάρχουν σε ένα project για να καλυφθούν τυχόν τροποποιήσεις ή προσθήκες στις απαιτήσεις του συστήματος.
- **Κίνδυνοι:** Αξιολόγηση των κινδύνων πρέπει να γίνεται σε περιπτώσεις που επηρεάζουν την επιτυχία του συστήματος.
- **Λύσεις:** Επίλυση των προβλημάτων και των κινδύνων είναι απαραίτητη για την ολοκλήρωση του συστήματος, οδηγώντας τις περισσότερες φορές σε αλλαγές.

Οι παραπάνω έλεγχοι μπορούν να χρησιμοποιηθούν σε διάφορες φάσεις της Scrum μεθοδολογίας. Η διοίκηση του οργανισμού και η ομάδα ανάπτυξης μπορεί να χρησιμοποιήσουν τους ελέγχους αυτούς για να διαχειριστούν το ανεκτέλεστο υπόλοιπο του συστήματος, τις αλλαγές και τυχόν προβλήματα που μπορεί να

προκύψουν. Οι έλεγχοι αυτοί ανακεφαλαιώνονται και τροποποιούνται σε κάθε συνάντηση sprint.

Η Scrum έχει χρησιμοποιηθεί για την ανάπτυξη σύνθετων εφαρμογών από τις αρχές τις δεκαετίας του 1990. Δεν πρόκειται για μια διαδικασία ή μια τεχνική για τη δημιουργία εφαρμογών και ούτε ένα πλαίσιο μέσα στο οποίο μπορούμε να χρησιμοποιήσουμε διάφορες μεθόδους και τεχνικές. Ο ρόλος της Scrum είναι η αποτελεσματικότητα των τεχνικών ανάπτυξης ώστε να μπορούμε να βελτιώνουμε τις εφαρμογές και η παροχή ενός πλαισίου εντός του οποίου οι πολύπλοκες εφαρμογές μπορούν να αναπτυχθούν. Βασίζεται στην εμπειρική διαδικασία ελέγχου και στην επανάληψη διαδικασιών ώστε να βελτιστοποιηθούν η προβλεψιμότητα και η διαχείριση επικινδυνότητας. Τρεις πυλώνες υποστηρίζουν της εφαρμογή κάθε εμπειρικής διαδικασίας ελέγχου και είναι οι επόμενοι.

- Η διαφάνεια: εξασφαλίζει ότι οι παράγοντες που επηρεάζουν το αποτέλεσμα μιας διαδικασίας θα πρέπει να είναι ορατοί στα εμπλεκόμενα άτομα που σχετίζονται με τη διαδικασία καθώς επίσης και όσα παρατηρούνται θα πρέπει να γίνονται γνωστά μέσα στα μέλη της ομάδας. Όταν μια ομάδα αναφέρει ότι μια εργασία έχει ολοκληρωθεί, θα πρέπει το παραδοτέο τμήμα του προϊόντος να συμφωνεί με τις απαιτήσεις οι οποίες έχουν καθοριστεί.
- Η επιθεώρηση: όλες οι διαδικασίες θα πρέπει να επιθεωρούνται σε μια μόνιμη και σταθερή βάση έτσι ώστε να μπορούν να ανιχνεύονται μη προβλέψιμες αποκλίσεις. Η συχνότητα των επιθεωρήσεων θα πρέπει να λαμβάνεται σοβαρά υπόψη, προβλήματα μπορεί να προκύψουν αν η συχνότητα των επιθεωρήσεων υπερβαίνει κατά πολύ τις απαιτούμενες συναντήσεις μιας διαδικασίας. Ευτυχώς αυτό δεν παρατηρείται συχνά στην ανάπτυξη πληροφοριακών συστημάτων. Πολύ σημαντικός παράγοντας στην επιθεώρηση θεωρείται η ικανότητα και η επιμέλεια των ατόμων που την επιτελούν.
- Η προσαρμογή: αν ο επιθεωρητής κρίνει κατά τη διάρκεια της επιθεώρησης ότι ένας ή περισσότεροι παράγοντες είναι εκτός των επιτρεπόμενων ορίων

και ότι το τελικό προϊόν δεν είναι το επιθυμητό, τότε προσαρμόζει τη διαδικασία το συντομότερο δυνατό για να αποφύγουμε περαιτέρω παρεκτροπή. Υπάρχουν τρία σημεία στη Scrum στα οποία συντελούνται οι επιθεωρήσεις και οι προσαρμογές. Οι καθημερινές Scrum συναντήσεις που χρησιμοποιούνται για τη σύγκριση της προόδου που έχει συντελεστεί σε σχέση με τον στόχο του Sprint και οι απαραίτητες προσαρμογές που βελτιστοποιούν τις εργασίες της επόμενης ημέρας. Αντίθετα η ανασκόπηση του sprint και οι συναντήσεις σχεδιασμού χρησιμοποιούνται για την αξιολόγηση της προόδου σε σχέση με τον στόχο μιας έκδοσης του προϊόντος και για τον καθορισμό των προσαρμογών που θα κάνουν το επόμενο sprint πιο παραγωγικό.

Η μεθοδολογία Scrum κάνει χρήση των time boxes. Τα στοιχεία της μεθοδολογίας που βασίζονται σε time boxes είναι το sprint, η συνάντηση σχεδιασμού του sprint (Sprint Planning Meeting), το Daily Scrum, ο απολογισμός sprint (Sprint Review) και η ανακεφαλαίωση sprint (Sprint Retrospective). Η καρδιά της Scrum είναι το sprint, πρόκειται για μια επαναληπτική διαδικασία διάρκειας το πολύ έως ένα μήνα. Όλα τα sprints χρησιμοποιούν το ίδιο πλαίσιο και παράγουν ένα μέρος από το συνολικό προϊόν την εφαρμογή και ξεκινάνε αμέσως μετά το τέλος του προηγούμενου sprint.

Η προσέγγιση Scrum χρησιμοποιεί επίσης τέσσερις βασικές τεχνικές:

- Το ανεκτέλεστο υπόλοιπο προϊόντος (Product Backlog) δηλαδή μια λίστα προτεραιοτήτων με οτιδήποτε μπορεί να χρειαστεί για την εκτέλεση των εργασιών του συστήματος.
- Το ανεκτέλεστο υπόλοιπο του sprint (Sprint Backlog) δηλαδή μια λίστα ενεργειών που πρέπει να γίνουν εντός του συγκεκριμένου sprint,
- Το διάγραμμα Release Burndown όπου απεικονίζονται διαγραμματικά οι ενέργειες που ολοκληρώνονται μέσα στο χρόνο σε σχέση με το ανεκτέλεστο υπόλοιπο του προϊόντος.

- Το Sprint Burndown όπου απεικονίζει τις ενέργειες που ολοκληρώνονται μέσα στο χρόνο σε σχέση με το ανεκτέλεστο υπόλοιπο του sprint.

Οι κανόνες (Rules) συνδυάζουν αρμονικά σε ένα Scrum project τα time boxes, τους ρόλους και τις τεχνικές. Για παράδειγμα ένας κανόνας θα μπορούσε να ήταν ότι μόνο τα μέλη των ομάδων που συμμετέχουν στην πρόοδο του ανεκτέλεστου υπόλοιπου προϊόντος μπορούν να παίρνουν μέρος στη συζήτηση μιας Daily Scrum. Όταν οι κανόνες δεν δηλώνονται με σαφή τρόπο, τότε οι χρήστες της μεθοδολογίας είναι εκείνοι που θα πάρουν αποφάσεις σχετικά με την εξέλιξη της εφαρμογής. Γενικότερα δεν είναι σωστή πρακτική να προσπαθούμε να βρούμε μια τέλεια λύση μιας και το πρόβλημα μπορεί να αλλάξει ξαφνικά (προσθήκη νέων απαιτήσεων ή τροποποίηση ήδη υπαρχόντων απαιτήσεων). Είναι προτιμότερο να δοκιμάσουμε κάτι και να δούμε πως δουλεύει το σύστημά μας. Σε αυτό μπορεί να μας φανεί ιδιαίτερα χρήσιμος ο μηχανισμός επιθεώρησης και τροποποίησης της Scrum.

Οι Scrum ομάδες έχουν σχεδιαστεί για να παρέχουν ευελιξία και παραγωγικότητα και αποτελούνται από τρεις ρόλους και είναι οι ακόλουθοι:

- Ιδιοκτήτης
- Scrum Master
- Ομάδα

Ο ιδιοκτήτης του προϊόντος λειτουργεί για να κατευθύνει την ομάδα στο σωστό στόχο. Αυτό γίνεται δημιουργώντας ένα όραμα για το προϊόν και στη συνέχεια μεταφέροντας αυτό το όραμα στην ομάδα μέσω του ανεκτέλεστου υπόλοιπου προϊόντος. Ο ιδιοκτήτης του προϊόντος είναι υπεύθυνος για να εξασφαλίσει ότι το ανεκτέλεστο κομμάτι της εφαρμογής παραμένει σε πρώτη προτεραιότητα για την ομάδα.

Ο Scrum Master θεωρείται ο προπονητής της ομάδας και βοηθά τα μέλη της ομάδας να επιτύχουν το υψηλότερο επίπεδο της απόδοσής τους. Ένας Scrum

Master διαφέρει από ένα Project Manager στο γεγονός ότι δεν παρέχει κατευθυντήριες οδηγίες καθημερινά στην ομάδα και στο ότι δεν αναθέτει καθήκοντα ατομικά σε μέλη της ομάδας. Ένας καλός Scrum Master προστατεύει την ομάδα από εξωτερικούς παράγοντες επιτρέποντας στα μέλη της να επικεντρωθούν κατά τη διάρκεια του sprint στο στόχο που έχουν επιλέξει.

Ο τρίτος και τελευταίος ρόλος σε ένα Scrum Project είναι η ίδια η ομάδα η οποία αποτελείται από άτομα με διαφορετικές δεξιότητες και ικανότητες και διαφορετικούς επαγγελματικούς προσανατολισμούς. Όταν τα μέλη μιας ομάδας Scrum συνεργάζονται, οι προσωπικές φιλοδοξίες δεν έχουν καμία σημασία. Κάθε άτομο συνεισφέρει τα μέγιστα που μπορεί προκειμένου να ολοκληρωθούν οι εργασίες σε κάθε sprint και διευρύνει τον ορίζοντα των γνώσεών του όταν και όπου αυτό απαιτείται για το γενικό καλό της ομάδας στην οποία εργάζεται.

Ένας τρόπος για να κατανοήσουμε τη φυσική σύμπλεξη αυτών των τριών ρόλων είναι να σκεφτούμε το αγωνιστικό αυτοκίνητο. Η ομάδα είναι το αυτοκίνητο που είναι έτοιμο να επιταχύνει προς οποιαδήποτε κατεύθυνση. Ο ιδιοκτήτης του προϊόντος είναι ο οδηγός που εξασφαλίζει ότι το αυτοκίνητο κινείται πάντοτε προς τη σωστή κατεύθυνση ενώ ο Scrum Master είναι ο επικεφαλής μηχανικός διατηρώντας το αυτοκίνητο καλά συντηρημένο έτσι ώστε να αποδώσει το μέγιστο δυνατό.

4.1 Ο ΙΔΙΟΚΤΗΤΗΣ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

Ο ιδιοκτήτης του project είναι συνήθως ο βασικός παράγοντας της εφαρμογής. Είναι σημαντικό ο ιδιοκτήτης να έχει ένα όραμα για το τι ακριβώς πρέπει να υλοποιηθεί και να είναι σε θέση να μπορεί να μεταφέρει το όραμα αυτό στην ομάδα. Αυτό γίνεται κυρίως μέσω της λίστας των προτεραιοτήτων του ανεκτέλεστου υπόλοιπου της εφαρμογής. Ο ιδιοκτήτης του project είναι συνήθως κάποιος κύριος χρήστης του συστήματος, ή κάποιος που έχει αντίληψη των απαιτήσεων των χρηστών ή του είδους του συστήματος που αναπτύσσεται, του

ανταγωνισμού της αγοράς εργασίας και των μελλοντικών τάσεων στον τομέα. Παρά το γεγονός ότι ο ιδιοκτήτης δίνει τη λίστα προτεραιοτήτων με το ανεκτέλεστο υπόλοιπο της εφαρμογής, κατά τη συνάντηση σχεδίασης του sprint, η ομάδα είναι εκείνη που επιλέγει το ποσό της εργασίας που πιστεύει ότι μπορεί να ολοκληρωθεί. Η δουλειά του ιδιοκτήτη του project είναι να παρακινήσει την ομάδα με ένα σαφή και καλά καθορισμένο στόχο. Τα μέλη της ομάδας γνωρίζουν καλύτερα τι είναι ικανά να κάνουν και γι' αυτό επιλέγουν τις ανασκοπήσεις των χρηστών από την κορυφή της λίστας προτεραιοτήτων με όλα εκείνα που μπορούν να δεσμευτούν ότι θα υλοποιήσουν.

Ως αντάλλαγμα της δέσμευσης των μελών της ομάδας, ο ιδιοκτήτης του συστήματος δεσμεύεται επίσης ότι δεν θα ζητήσει νέες απαιτήσεις από την ομάδα κατά τη διάρκεια του συγκεκριμένου sprint. Οι απαιτήσεις επιτρέπεται να αλλάξουν (και συγκεκριμένα ενθαρρύνονται οι αλλαγές των απαιτήσεων) αλλά μόνο εκτός από το sprint που βρίσκεται προς υλοποίηση. Μόλις η ομάδα ξεκινήσει το sprint, παραμένει μανιωδώς επικεντρωμένη στο στόχο του sprint. Με λίγα λόγια, ο ιδιοκτήτης του συστήματος έχει τις παρακάτω δραστηριότητες:

- 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.

4.2 Ο SCRUM MASTER

Ο Scrum Master είναι υπεύθυνος να διασφαλίσει ότι η ομάδα Scrum δρα σύμφωνα με τις αξίες και τις πρακτικές του Scrum. Θεωρείται ο προπονητής της ομάδας που τη βοηθάει να κάνει το καλύτερο δυνατό που μπορεί, καθώς επίσης και ο

ιδιοκτήτης της διαδικασίας που δημιουργεί μια ισορροπία μεταξύ της ομάδας και του ιδιοκτήτη του προγράμματος. Κάνει ότι είναι δυνατό για να επιτύχει η ομάδα το μέγιστο αποτέλεσμα, αφαιρώντας τυχόν εμπόδια που μπορεί να προκύψουν, εγκαθιδρύει συνεδριάσεις και συνεργάζεται με τον ιδιοκτήτη της εφαρμογής για να βεβαιωθεί ότι το ανεκτέλεστο υπόλοιπο του συστήματος είναι στο σωστό σημείο και έτοιμο για το επόμενο sprint. Ο ρόλος του Scrum Master μπορεί να καλυφθεί από έναν πρώην project manager ή έναν τεχνικό υπεύθυνο, χωρίς αυτό να αποκλείει το γεγονός ότι μπορεί να είναι οποιοδήποτε άλλο μέλος της ομάδας όπως ο developer του συστήματος. Ωστόσο αυτό τις περισσότερες φορές οδηγεί σε διαμάχες όταν ο Scrum Master πρέπει να επιλέξει μεταξύ της μετακίνησης πιθανών εμποδίων ή της εκτέλεσης επιπλέον εργασιών. Σε καμία περίπτωση Scrum Master δεν μπορεί να είναι ο ιδιοκτήτης του συστήματος.

[<http://www.mountaingoatsoftware.com/topics/scrum>]

Ο Scrum Master συχνά μπορεί να θεωρείται και προστάτης της ομάδας, παρέχοντας προστασία στα μέλη της ώστε να μη δεσμευτούν με προτάσεις τις οποίες δεν μπορούν να υλοποιήσουν κατά τη διάρκεια ενός sprint. Αυτό μπορεί να έχει ως αποτέλεσμα την υπερβολική πίεση ή και την επιθετική συμπεριφορά από μέρους του ιδιοκτήτη. Πολλοί από εκείνους που είναι νέοι στο ρόλο του Scrum Master, αντιφάσκουν μεταξύ ενός ηγετικού ρόλου στην ομάδα και ενός ρόλου χωρίς καμία ηγετική ικανότητα. Η φαινομενική αντίφαση εξαφανίζεται όταν συνειδητοποιούμε ότι παρόλο που ο Scrum Master δεν έχει καμία εξουσία πάνω στα μέλη της ομάδας, έχει εξουσία επί της διαδικασίας που πρόκειται να υλοποιηθεί. Παρόλο που ο Scrum Master δεν μπορεί να απολύσει κάποιο από τα μέλη της ομάδας, είναι σε θέση να αποφασίσει ότι θα υλοποιηθεί ένα sprint δύο εβδομάδων το επόμενο χρονικό διάστημα. [<http://www.mountaingoatsoftware.com/topics/scrum>]

Ο Scrum Master είναι αυτός που θα βοηθήσει την ομάδα, όπως ένας προσωπικός γυμναστής μας βοηθάει να μην κολλήσουμε με το πρόγραμμα μιας συγκεκριμένης άσκησης και να εκτελέσουμε όλες τις ασκήσεις με τη σωστή σειρά. Όπως ένας καλός εκπαιδευτής, έτσι και ο Scrum Master παρέχει κίνητρα, ενώ παράλληλα εξασφαλίζει ότι δεν θα παραβλέψουμε να εκτελέσουμε μια επίπονη και σκληρή

διαδικασία. Όπως και να έχει, δεν μπορεί να επιβάλει στα μέλη της ομάδας να κάνουν κάτι που δεν θέλουν, μπορεί όμως με τον τρόπο του να τους θυμίσει τους στόχους και τον τρόπο που έχουμε επιλέξει για να τους εκπληρώσουμε. Με λίγα λόγια οι Scrum Masters έχουν εξουσία, αλλά η εξουσία τους πηγάζει και παρέχεται πάνω σε μια συγκεκριμένη διαδικασία. Για παράδειγμα έστω ότι έχουμε να παραδώσουμε ένα μέρος ενός κώδικα μέχρι το τέλος ενός sprint. Αν κάτι πάει στραβά και ο κώδικας δεν τρέξει, τότε ο Scrum Master δεν μπορεί να επιβάλει σε κάποιο μέλος της ομάδας να επανεξετάσει τον κώδικα. Η επανεξέταση του κώδικα είναι κάτι το οποίο πρέπει να γίνει χωρίς όμως ο Scrum Master να αποφασίσει ποιος είναι εκείνος που θα κάνει την επανεξέταση. Γιατί στην περίπτωση αυτή υπερβαίνει την εξουσία που έχει επί της διαδικασίας και παρεμβαίνει στον τρόπο με τον οποίο λειτουργεί και δουλεύει η ομάδα.

Με το ρόλο του να περιορίζεται μόνο στο να εξασφαλίσει ότι η ομάδα ακολουθεί τη διαδικασία, το έργο του Scrum Master μπορεί να αποδειχθεί πιο δύσκολο από εκείνο ενός project manager. Οι project managers έχουν την εναλλακτική επιλογή να πούνε «κάντε κάτι επειδή σας το ζητάω». Οι Scrum Masters μπορούν να πούνε κάτι τέτοιο μόνο στην περίπτωση που δεν διασφαλίζεται μια Scrum διαδικασία. Συνοψίζοντας θα μπορούσαμε να πούμε ότι οι κύριες αρμοδιότητες ενός Scrum Master είναι οι ακόλουθες:

- 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.

4.3 Η ΟΜΑΔΑ SCRUM

Η ομάδα η οποία εργάζεται για τη δημιουργία μιας έκδοσης του προϊόντος περιλαμβάνει προγραμματιστές και εξωτερικές οντότητες οι οποίες επηρεάζονται από την έκδοση του προϊόντος όπως οι πελάτες. Στις περιπτώσεις ανάπτυξης ενός μέρους του συστήματος, η τελευταία ομάδα μένει μακριά από τη διαδικασία υλοποίησης έτσι ώστε να μην κάνει τη διαδικασία δυσκολότερη παρέχοντας άχρηστες πληροφορίες. Η μεθοδολογία Scrum ωστόσο ενθαρρύνει τη συμμετοχή τους σε άλλες φάσεις ανάπτυξης του συστήματος μιας και η συμβολή τους μπορεί να αποδειχθεί σημαντική. Οι ακόλουθες ομάδες σχηματίζονται κατά τη διάρκεια μιας νέας έκδοσης λογισμικού.

- Η διοίκηση του οργανισμού: Με ηγέτη τον product manager καθορίζεται το αρχικό περιεχόμενο και ο χρόνος παράδοσης της νέας έκδοσης του προϊόντος. Στη συνέχεια ο project manager παρακολουθεί την εξέλιξη υλοποίησης του συστήματος, καθώς οι μεταβλητές του περιβάλλοντος αλλάζουν. Η διοίκηση ασχολείται κυρίως με θέματα χειρισμού του ανεκτέλεστου υπόλοιπου προϊόντος, με τη διαχείριση επικινδυνότητας και με το περιεχόμενο της νέας έκδοσης του συστήματος.
- Οι ομάδες ανάπτυξης: Είναι συνήθως μικρές σε μέγεθος και η κάθε μία περιέχει προγραμματιστές, προσωπικό που ασχολείται με την τεκμηρίωση και τον έλεγχο της ποιότητας του παραγόμενου προϊόντος. Συνήθως χρησιμοποιούνται μία ή περισσότερες ομάδες μεταξύ τριών και έξι ατόμων σε κάθε μια από τις οποίες ανατίθενται συγκεκριμένες εργασίες και καθήκοντα. Οι ομάδες καθορίζουν τις απαιτούμενες αλλαγές που πρέπει να γίνουν και διαχειρίζονται όλα τα εμπόδια που μπορεί να προκύψουν από τις αλλαγές αυτές. Τα μέλη της κάθε ομάδας επιλέγονται ανάλογα με τις γνώσεις και την εξειδίκευση που έχουν σε συγκεκριμένους τομείς.

Η ομάδα Scrum δεν περιλαμβάνει κανέναν από τους παραδοσιακούς ρόλους της μηχανικής λογισμικού, όπως του προγραμματιστή, του σχεδιαστή ή του ελεγκτή αν

και μερικές φορές μπορεί να υπάρχουν και μέλη αποκλειστικής απασχόλησης (π.χ. database administrator). Όλοι μαζί συνεργάζονται για να ολοκληρώσουν το σύνολο των εργασιών για το οποίο έχουν δεσμευτεί σε ένα συγκεκριμένο sprint. Η ομάδα διακατέχεται από υψηλό αίσθημα αλληλοβοήθειας και έχει την αίσθηση ότι όλοι είναι μαζί και λειτουργούν ομαδικά προκειμένου να ολοκληρώσουν το καθήκον τους. Άτομα τα οποία αρνούνται να γράψουν κώδικα επειδή θεωρούν πως είναι καλύτεροι designers ή αναλυτές δεν αποτελούν καλές επιλογές ως μέλη μιας ομάδας. Όλοι συνεισφέρουν εξίσου ακόμη και αν αυτό συνεπάγεται την απόκτηση νέων ικανοτήτων ή την υπενθύμιση παλαιότερων.

[<http://www.mountaingoatsoftware.com/topics/scrum>]

Μια τυπική ομάδα Scrum αποτελείται από 5 – 9 άτομα και κλιμακώνεται με το σχηματισμό υποομάδων εντός των ομάδων όταν αυτό απαιτείται. Με τον τρόπο αυτό μπορεί να υπάρχουν και meetings μεταξύ των υποομάδων. Για το λόγο αυτό, θα πρέπει η κάθε υποομάδα να ορίσει ένα άτομο το οποίο θα παρακολουθεί τα επιμέρους meetings ώστε να συντονίζονται σωστά και οι εργασίες εντός των υποομάδων. Όταν σε μια ομάδα υπάρχουν λιγότερα από πέντε άτομα τότε υπάρχει μικρότερη αλληλεπίδραση μεταξύ τους και μπορεί να έχουμε χαμηλότερη παραγωγικότητα. Επίσης η ομάδα μπορεί να συναντήσει περιορισμούς ικανοτήτων και να μην είναι σε θέση να ολοκληρώσει μια εργασία μέσα στο sprint με αποτέλεσμα να μην είναι εφικτό να παραδοθεί κάποιο τμήμα του συστήματος. Στην περίπτωση που υπάρχουν περισσότερα από εννέα άτομα, τότε συμβαίνει ακριβώς το αντίθετο, αυξάνεται αρκετά η πολυπλοκότητα και η διαχείριση των εργασιών γίνεται δυσκολότερη. Ωστόσο υπάρχουν και περιπτώσεις στις οποίες ομάδες έχουν υπερβεί τα ανώτερα και τα κατώτερα όρια του αριθμού των μελών, χωρίς αυτό να επηρεάσει την παραγωγικότητά τους. Ο Scrum Master και ο ιδιοκτήτης του προϊόντος δεν λαμβάνονται υπόψη στον παραπάνω υπολογισμό.

Συνοψίζοντας θα μπορούσαμε να πούμε ότι μια ομάδα έχει τα εξής κύρια χαρακτηριστικά:

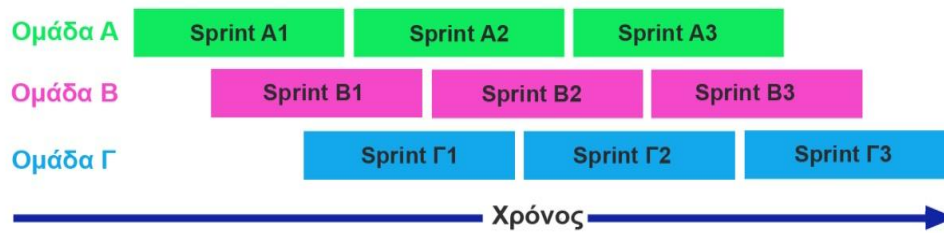
- Τυπικά αποτελείται από 5-9 άτομα.

- Τα μέλη είναι διασταυρωμένης λειτουργίας δηλαδή αλληλοβοηθούνται ανεξάρτητα από την ειδικότητά τους όταν αυτό είναι απαραίτητο.
 - Προγραμματιστές
 - Ελεγκτές
 - Σχεδιαστές εφαρμογών κ.τ.λ.
- Μπορεί να υπάρχουν και μέλη αποκλειστικής απασχόλησης.
 - Ίσως κάποιες εξαιρέσεις (π.χ. database administrator)
- Η ομάδα είναι αυτό-οργανωμένη, ιδανικά χωρίς τίτλους.

4.3.1 ΧΕΙΡΙΣΜΟΣ ΠΟΛΛΑΠΛΩΝ ΟΜΑΔΩΝ ΣΕ ΕΝΑ SCRUM PROJECT

Τα πράγματα γίνονται πιο πολύπλοκα όταν έχουμε πολλές ομάδες Scrum που δουλεύουν στο ίδιο project. Το πρόβλημα αυτό είναι γενικό και δεν έχει να κάνει αποκλειστικά με τη μεθοδολογία Scrum, περισσότεροι προγραμματιστές σημαίνει μεγαλύτερη πολυπλοκότητα και περισσότερες αλληλεπιδράσεις. Όταν υπάρχουν μεγάλες ομάδες (πάνω από 10 μέλη) έχει παρατηρηθεί ότι οι καθημερινές Scrum συναντήσεις διαρκούν περισσότερο από 15 λεπτά και ο Scrum Master συναντά δυσκολίες να κρατήσει όλα τα μέλη της ομάδας αφοσιωμένα στο στόχο ή να αντιμετωπίσει όλα τα εμπόδια που μπορεί να εμφανιστούν. Έτσι ίσως είναι προτιμότερο να έχουμε περισσότερες ομάδες με λιγότερα μέλη στην κάθε μία και σε περιπτώσεις όπου δεν υπάρχει άμεση εξάρτηση και αλληλεπίδραση μεταξύ τους. Ο χωρισμός των ομάδων σε μικρότερες ομάδες είναι ένα από τα δυσκολότερα σημεία της μεθοδολογίας Scrum. Η καλύτερη λύση δεν μπορεί να βρεθεί εξαρχής παρά μόνο ύστερα από συστηματική παρατήρηση για το αν οι ομάδες μπορούν να δουλέψουν καλύτερα από μόνες τους ή σε συνεργασία. [Henrik Kniberg, “Scrum and XP from the Trenches”, 2006, p 69]

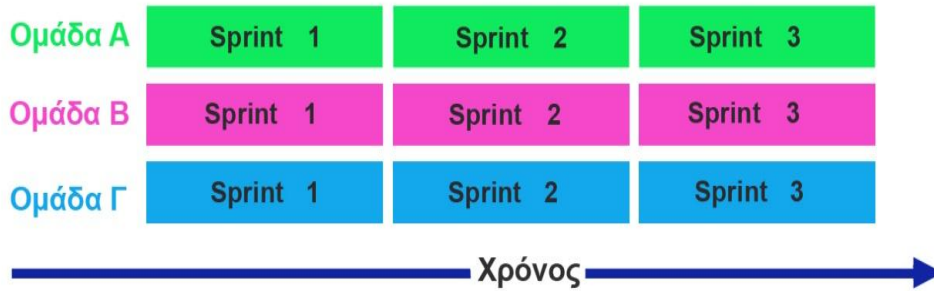
Ένα ερώτημα που μπορεί να προκύψει όταν έχουμε πολλές ομάδες να δουλεύουν πάνω στο ίδιο project είναι αν τα sprints θα πρέπει να είναι συγχρονισμένα (να ξεκινούν να ολοκληρώνονται μαζί) ή αν είναι προτιμότερο να επικαλύπτονται μεταξύ τους. Η δεύτερη περίπτωση αποτυπώνεται γραφικά στην παρακάτω εικόνα.



Εικόνα 8: Πολλαπλές ομάδες με επικαλυπτόμενα sprints [Henrik Kniberg, “Scrum and XP from the Trenches”, 2006, p 71]

Σε αυτή την περίπτωση σε κάθε δεδομένη χρονική στιγμή υπάρχει ένα sprint το οποίο βρίσκεται σχεδόν στο τέλος του και ένα νέο που ξεκινάει. Ο συνολικός φόρτος εργασίας διανέμεται από τον ιδιοκτήτη του συστήματος σε όλο το χρονικό διάστημα και ιδανικά θα πρέπει να υπάρχουν συνεχόμενες εκδόσεις λογισμικού από τις ομάδες. Επειδή είναι αρκετά πειστικό και δύσκολο για τις ομάδες να παράγουν συνεχώς νέες εκδόσεις λογισμικού, προτιμάται συνήθως η πρώτη λύση των συγχρονισμένων sprints, όπως φαίνεται στην εικόνα 9, η οποία παρέχει και τα παρακάτω πλεονεκτήματα. [Henrik Kniberg, “Scrum and XP from the Trenches”, 2006, p 72]

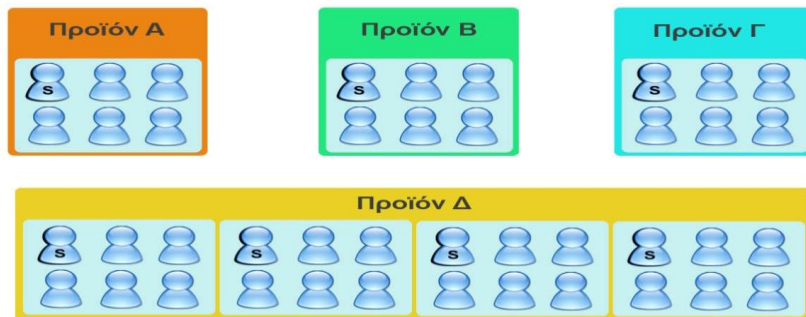
- Υπάρχει ένα φυσικό σημείο στο οποίο μπορούν να αναδιαταχθούν οι ομάδες (μεταξύ των sprints), δυνατότητα η οποία δεν παρέχεται με τα επικαλυπτόμενα sprints χωρίς να διακοπεί τουλάχιστον μια ομάδα στο μέσο ενός sprint.
- Όλες οι ομάδες έχουν ένα κοινό στόχο στη διάρκεια ενός sprint, ο οποίος οδηγεί σε καλύτερη συνεργασία μεταξύ των ομάδων.
- Υπάρχει λιγότερος διαχειριστικός φόρτος μιας και οι συναντήσεις σχεδιασμού sprint και οι δοκιμαστικές εκδόσεις λογισμικού είναι λιγότερες.



Εικόνα 9: Πολλαπλές ομάδες με συγχρονισμένα sprints [Henrik Kniberg, “Scrum and XP from the Trenches”, 2006, p 72]

Σε κάθε περίπτωση όταν έχουμε πολλαπλές ομάδες Scrum που αναπτύσσουν μια εφαρμογή όπως συμβαίνει σε αρκετές περιπτώσεις, τότε πρέπει να επιλέξουμε ένα άτομο το οποίο θα έχει ηγετικό ρόλο μεταξύ των ομάδων. Το άτομο αυτό ονομάζεται team leader, έχει τη γενική επίβλεψη όλων των επιμέρους ομάδων και μπορεί να αποφασίζει για θέματα όπως πόσα μέλη θα έχει η κάθε ομάδα, αν θα δουλεύουν παράλληλα οι ομάδες ή αν ενδεχομένως χρειαστεί να γίνει κάποιος μετασχηματισμός (αλλαγή μελών ή ρόλων μεταξύ των μελών των ομάδων).

Το Scrum of Scrums είναι μία συνάντηση όπως όλες οι υπόλοιπες με τη μόνη διαφορά ότι σε αυτή συμμετέχουν μόνο οι Scrum Masters της κάθε επιμέρους ομάδας. Με βάση το παράδειγμα που ακολουθεί έχουμε τέσσερα διαφορετικά τμήματα της εφαρμογής που πρέπει να υλοποιηθούν. Τα τρία πρώτα τμήματα υλοποιούνται από μια ομάδα, ενώ το τελευταίο τμήμα για να αναπτυχθεί απαιτεί 25 άτομα που πρέπει να χωριστούν σε επιμέρους Scrum ομάδες όπως φαίνεται και στην εικόνα που ακολουθεί.



Εικόνα 10: Πολλαπλές ομάδες σε ένα Scrum project [Henrik Kniberg, “Scrum and XP from the Trenches”, 2006, p 78]

Με βάση την παραπάνω εικόνα και όσα έχουμε αναφέρει μέχρι στιγμής γίνεται αντιληπτό ότι θα έχουμε δύο Scrum of Scrums meetings, το ένα στο επίπεδο του τέταρτου τμήματος του προϊόντος μεταξύ των Scrum Masters των υποομάδων και το άλλο μεταξύ των Scrum Masters όλων των τμημάτων της εφαρμογής. Θέματα που μπορεί να αναλύονται κατά τη διάρκεια των συγκεκριμένων συναντήσεων αφορούν συνήθως εργασίες προγραμματισμού για τη συνάντηση σχεδιασμού του επόμενου sprint, συζητήσεις για το ανεκτέλεστο υπόλοιπο των εργασιών που ολοκληρώθηκε, εργασίες που πρόκειται να ολοκληρωθούν καθώς επίσης και ενημερώσεις ή νέες εκδόσεις της εφαρμογής που πρόκειται να κυκλοφορήσουν.

Το ερώτημα που τίθεται συνήθως όταν έχουμε πολλές Scrum ομάδες είναι πώς θα πρέπει να χωρίσουμε το ανεκτέλεστο υπόλοιπο του προϊόντος. Υπάρχουν τρεις διαφορετικές προσεγγίσεις στη μεθοδολογία Scrum. Στην πρώτη που είναι και η πιο συνηθισμένη έχουμε ένα ιδιοκτήτη συστήματος και ένα ανεκτέλεστο υπόλοιπο με τις ομάδες να αποφασίζουν από μόνες τους για το πώς θα χωρίσουν τις εργασίες ανάλογα με την περιγραφή του ιδιοκτήτη.



Εικόνα 11: Πολλαπλές ομάδες & ανεκτέλεστο υπόλοιπο [Henrik Kniberg, "Scrum and XP from the Trenches", 2006, p 81]

Στη δεύτερη προσέγγιση έχουμε έναν ιδιοκτήτη της εφαρμογής και πολλαπλές λίστες για το ανεκτέλεστο υπόλοιπο του προϊόντος και πιο συγκεκριμένα μία λίστα για την κάθε ομάδα. Μια αδυναμία της συγκεκριμένης στρατηγικής αποτελεί το γεγονός ότι ο ιδιοκτήτης διανέμει τις εργασίες στις ομάδες, διαδικασία που γίνεται

πιο σωστά όταν οι ομάδες επιλέγουν τις εργασίες που μπορούν να ολοκληρώσουν μόνες τους μιας και έχουν καλύτερη γνώση των ικανοτήτων τους.



Εικόνα 12: Πολλαπλές ομάδες & ανεκτέλεστο υπόλοιπο [Henrik Kniberg, "Scrum and XP from the Trenches", 2006, p 82]

Τέλος στην τρίτη προσέγγιση έχουμε περισσότερους από έναν ιδιοκτήτες του συστήματος όπου καθένας τους κατέχει από μια λίστα του ανεκτέλεστου υπόλοιπου της εφαρμογής. Σε αυτή την περίπτωση υπάρχει πιθανότητα οι διαφορετικές λίστες να περιέχουν την ίδια βάση του κώδικα, κάτι που μπορεί να προκαλέσει αντιπαραθέσεις μεταξύ των δύο ιδιοκτητών. Αν οι διαφορετικές λίστες του ανεκτέλεστου υπόλοιπου έχουν σαν βάση τον ίδιο κώδικα που εκτελείται ξεχωριστά από διαφορετικές ομάδες, τότε έχουμε το ίδιο αποτέλεσμα με το να εκτελούσε το sprint η κάθε ομάδα ξεχωριστά από μια φορά.



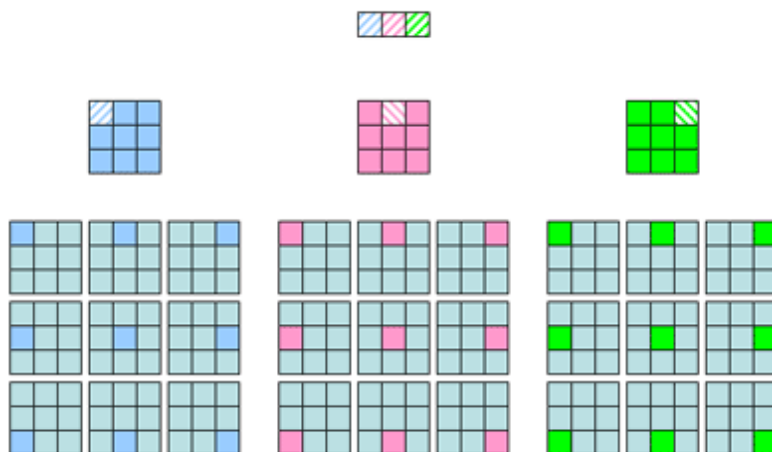
Εικόνα 13: Πολλαπλές ομάδες & ανεκτέλεστο υπόλοιπο [Henrik Kniberg, "Scrum and XP from the Trenches", 2006, p 83]

Σε κάθε περίπτωση θα πρέπει να ακολουθούνται κάποιες γενικές αρχές όταν συνεργάζονται πολλές ομάδες πάνω στο ίδιο σύστημα και είναι οι ακόλουθες:

- Στη χειρότερη περίπτωση θα πρέπει όλα τα τμήματα της εφαρμογής να μεταγλωττίζονται και να περνούν όλους τους απαραίτητους ελέγχους. Σε οποιαδήποτε χρονική στιγμή πρέπει να είμαστε σε θέση να δημιουργήσουμε μια νέα έκδοση του προϊόντος.
- Ιδανικά τα επιμέρους τμήματα της εφαρμογής που παραδίδονται από τις διαφορετικές ομάδες πρέπει να συνδέονται μεταξύ τους και να δημιουργούν ένα ολοκληρωμένο τμήμα της συνολικής εφαρμογής.
- Οποτεδήποτε δημιουργούμε μια νέα έκδοση θα πρέπει να καθορίζουμε τον ακριβή αριθμό της έκδοσης, την ημερομηνία και να περιγράψουμε με συντομία τα κύρια χαρακτηριστικά της έτσι ώστε όταν χρειαστούμε οποιαδήποτε πληροφορία στο μέλλον να είναι άμεσα διαθέσιμη.
- Δημιουργούμε παρακλάδια, δηλαδή επιμέρους προσθήκες στην κύρια ραχοκοκαλιά του κώδικα μόνο όταν είναι απαραίτητο και δεν μπορούμε πλέον να χρησιμοποιήσουμε την ήδη υπάρχουσα βάση του κώδικα.
- Παρακλάδια χρησιμοποιούμε συνήθως για να ξεχωρίσουμε τους διαφορετικούς κύκλους ζωής του συστήματος.
- Συγχρονίζουμε τα παρακλάδια του κώδικα με το κύριο σώμα του κάθε φορά που χτίζουμε κάτι καινούργιο. Συγχρονισμός θα πρέπει επίσης να γίνεται σε συστηματική βάση για να κρατάμε ενημερωμένο το παρακλάδι του κώδικα σε σχέση με τα τμήματα που ολοκλήρωσαν και οι υπόλοιπες ομάδες. [Henrik Kniberg, "Scrum and XP from the Trenches", 2006, p 84]

Η εικόνα που ακολουθεί απεικονίζει τον τρόπο με τον οποίο γίνεται η κλιμάκωση μιας Scrum ομάδας, η οποία στη συγκεκριμένη περίπτωση αποτελείται από 243 άτομα, κάθε κελί απεικονίζει ένα άτομο μέσα στην ομάδα. Στην προκειμένη περίπτωση έχουμε 3 Scrum ομάδες οι οποίες στη συνέχεια κλιμακώνονται ώστε η κάθε μία από αυτές να περιλαμβάνει 9 υποομάδες. Από το κάτω μέρος της εικόνας φαίνεται ότι για κάθε υποομάδα επιλέγεται ένα άτομο (αυτό με το διαφορετικό

χρωματισμό κελιού) όπου θα πάρει μέρος στα meetings των υποομάδων προκειμένου να έχουμε συντονισμό των εργασιών εντός των υποομάδων. Στη συνέχεια επιλέγεται ένας από αυτούς (εκείνος με το διαφορετικό διαγώνιο χρωματισμό) ώστε να συμμετάσχει στα Scrum meetings των ομάδων.



Εικόνα 14: Κλιμάκωση Scrum ομάδας [<http://www.mountaingoatsoftware.com/topics/scrum>]

4.3.2 ΠΙΝΑΚΕΣ ΚΑΘΗΚΟΝΤΩΝ (TASK BOARDS)

Μια καλή πρακτική είναι να οπτικοποιήσουμε το ανεκτέλεστο υπόλοιπο του sprint σε πίνακες καθηκόντων. Τα μέλη της ομάδας ενημερώνουν συνεχώς τους πίνακες καθηκόντων σε όλη τη διάρκεια του sprint, αν κάποιος σκεφτεί μια νέα εργασία, γράφει μια νέα κάρτα και την αναρτά στον πίνακα καθηκόντων. Είτε κατά τη διάρκεια, είτε πριν από την καθημερινή Scrum συνάντηση οι εκτιμήσεις μπορεί να αλλάξουν και οι κάρτες των καθηκόντων μετακινούνται συνεχώς γύρω από τον πίνακα. Η γενική μορφή ενός πίνακα καθηκόντων φαίνεται στην παρακάτω εικόνα.

Περιγραφή	Υπολειπόμενες Εργασίες		Σε εξέλιξη	Επιβεβαίωση	Ολοκληρώθηκαν
	Κωδικοποίησε...	Δοκίμασε ...			
Ο χρήστης θα πρέπει να ...	Ενημέρωσε ...	Έλεγε ...	Υλοποίησε τον πηγαίο κώδικα	Έλεγε τον πηγαίο κώδικα	Κωδικοποίησε ..., Δοκίμασε ..., Έλεγε ...

	Υλοποίη σε ...	Εκτέλεσε...			
Ο διαχειριστή ς θα έχει έλεγχο ...	Τσέκαρε ...	Προγραμμ άτισε ...	Τσέκαρε τη λειτουργικότη τα της νέας έκδοσης	Διόρθωσε τυχόν σφάλματα	Προγραμμάτισε ..., Υλοποίησε ...
	Τρέξε το ...	Ολοκλήρω σε ...			

Πίνακας 6: Πίνακας καθηκόντων

Κάθε γραμμή στον πίνακα είναι μια περιγραφή του χρήστη και μπορεί να αναφέρεται και στο ανεκτέλεστο υπόλοιπο της εφαρμογής. Κατά τη διάρκεια της σχεδίασης συνάντησης του sprint η ομάδα επιλέγει το ανεκτέλεστο υπόλοιπο που μπορεί να ολοκληρώσει μέχρι το επόμενο sprint. Το ανεκτέλεστο υπόλοιπο της εφαρμογής μετατρέπεται σε ανεκτέλεστο υπόλοιπο του sprint. Κάθε ένα από αυτά αντιπροσωπεύεται από ένα καθήκον το οποίο τοποθετείται στον πίνακα καθηκόντων. Κάθε εργασία ξεκινάει από τη στήλη «Υπολειπόμενες Εργασίες», οι στήλες που συνήθως χρησιμοποιούνται σε ένα πίνακα καθηκόντων είναι οι εξής:

- Περιγραφή, περιέχει την περιγραφή του χρήστη.
- Υπολειπόμενες Εργασίες, εργασίες οι οποίες δεν έχουν ξεκινήσει.
- Σε εξέλιξη, εργασίες οι οποίες βρίσκονται σε εξέλιξη.
- Επιβεβαίωση, έλεγχος ότι μια εργασία έχει ολοκληρωθεί.
- Ολοκληρώθηκαν, καθήκοντα τα οποία έχουν τελειώσει.

5.1 ΑΝΕΚΤΕΛΕΣΤΟ ΥΠΟΛΟΙΠΟ ΤΟΥ ΠΡΟΪΟΝΤΟΣ (PRODUCT BACKLOG)

Οι απαιτήσεις για το προϊόν που αναπτύσσεται ταξινομούνται στο ανεκτέλεστο υπόλοιπο της εφαρμογής. Ο ιδιοκτήτης είναι υπεύθυνος για το περιεχόμενο, τη διαθεσιμότητα και την προτεραιότητα του ανεκτέλεστου υπόλοιπου, το οποίο εξελίσσεται και αλλάζει συνεχώς καθώς υλοποιείται το project έτσι ώστε να είναι ανταγωνιστικό, χρήσιμο και σύμφωνο με τις αλλαγές των μεταβλητών του περιβάλλοντος. Τα στοιχεία του ανεκτέλεστου υπόλοιπου δηλώνονται συνήθως ως ιστορίες χρηστών (user stories). Οι περιπτώσεις χρήσης (use cases) θα μπορούσε να είναι παρόμοιος χαρακτηρισμός αλλά έχουν μεγαλύτερη χρησιμότητα στο στάδιο υλοποίησης του λογισμικού.

Το ανεκτέλεστο υπόλοιπο του project είναι ένας κατάλογος προτεραιότητας που περιέχει σύντομες περιγραφές όλων των επιθυμητών λειτουργιών της εφαρμογής. Όταν χρησιμοποιούμε τη Scrum δεν είναι απαραίτητο να ξεκινάμε ένα project με μακροσκελή προσπάθεια καταγραφής όλων των απαιτήσεων. Συνήθως μια ομάδα Scrum και ο ιδιοκτήτης του project ξεκινάνε καταγράφοντας πρώτα όλα όσα μπορούν εύκολα να σκεφτούνε (brainstorming). Αυτό σαν πρώτο βήμα είναι περισσότερο από αρκετό για το πρώτο sprint. Το ανεκτέλεστο υπόλοιπο αυξάνεται και αλλάζει συνεχώς καθώς μαθαίνουμε ολοένα και περισσότερα στοιχεία για το project και τους τελικούς χρήστες του. Μια τυπική περίπτωση ανεκτέλεστου υπόλοιπου εφαρμογής μπορεί να περιλαμβάνει τα ακόλουθα στοιχεία:

- Αποσφαλμάτωση (bugs)
- Τεχνικά χαρακτηριστικά / λειτουργίες
- Τεχνικές εργασίες
- Απόκτηση γνώσης

Μέχρι στιγμής, ο κυρίαρχος τρόπος για να εκφράσει μια ομάδα Scrum τα τεχνικά χαρακτηριστικά του ανεκτέλεστο υπόλοιπου είναι οι περιγραφές των χρηστών, οι οποίες είναι σύντομες, απλές και με την επιθυμητή λειτουργικότητα από την πλευρά του χρήστη. Τα σφάλματα σχετίζονται συνήθως με το ανεκτέλεστο υπόλοιπο της εφαρμογής και περιλαμβάνουν διορθώσεις που πρέπει να γίνουν στον κώδικα της εφαρμογής για να μπορέσει να ολοκληρωθεί σωστά μια λειτουργία. Οι τεχνικές εργασίες μπορεί να σχετίζονται για παράδειγμα με την αναβάθμιση της πλατφόρμας του λειτουργικού συστήματος ή του λογισμικού στην οποία αναπτύσσεται η εφαρμογή ενώ ένα παράδειγμα απόκτησης γνώσεων θα μπορούσε να είναι η αναζήτηση διάφορων βιβλιοθηκών javascript και η επιλογή της καταλληλότερης.

Το ανεκτέλεστο υπόλοιπο αναπαριστά οτιδήποτε είναι απαραίτητο για να ξεκινήσει για να ολοκληρωθεί ένα πληροφοριακό σύστημα, είναι μια λίστα με όλα τα χαρακτηριστικά, τις λειτουργίες και τις βελτιώσεις του προϊόντος που θα κυκλοφορήσει. Τα στοιχεία του product backlog χαρακτηρίζονται από την περιγραφή, την προτεραιότητα και το χρονικό προσδιορισμό. Η προτεραιότητα είναι αυτή που καθοδηγεί άμεσα τις ενέργειες υλοποίησης του συστήματος και συγκεκριμένα υψηλότερη προτεραιότητα σημαίνει πιο επείγουσες και αναλυτικότερες σε περιγραφή ενέργειες που πρέπει να ολοκληρωθούν. Ο χρονικός προσδιορισμός ολοκλήρωσης των εργασιών είναι πιο εύκολο να υπολογισθεί όσο πιο σαφείς και αναλυτικές είναι οι περιγραφές.

Ο ιδιοκτήτης του project εμφανίζεται στη συνάντηση σχεδίασης του sprint με το ανεκτέλεστο υπόλοιπο της εφαρμογής ταξινομημένο κατά προτεραιότητα και περιγράφει τα σημαντικότερα στοιχεία στην ομάδα. Η ομάδα καθορίζει στη συνέχεια ποια από τα στοιχεία αυτά μπορούν να ολοκληρωθούν στο επόμενο sprint και μετακινεί τα στοιχεία από το ανεκτέλεστο υπόλοιπο της εφαρμογής, στο ανεκτέλεστο υπόλοιπο του sprint. Στο σημείο αυτό σε κάθε ανεκτέλεστο υπόλοιπο project αντιστοιχίζονται μια ή περισσότερες εργασίες ανεκτέλεστο υπόλοιπου του sprint έτσι ώστε να μπορούμε με πιο αποτελεσματικό τρόπο να κάνουμε τον καταμερισμό των εργασιών κατά τη διάρκεια ενός sprint. Η ομάδα ξεκινά από την

κορυφή των προτεραιοτήτων του ανεκτέλεστου υπόλοιπου της εφαρμογής και σχεδιάζει μια γραμμή μέχρι το σημείο των προτεραιοτήτων που μπορεί να ολοκληρωθεί. Τα υπόλοιπα στοιχεία από τη λίστα προτεραιότητας που δεν συμπεριλήφθηκαν σε αυτό το sprint, θα ολοκληρωθούν σε κάποιο από τα επόμενα.

Καθώς το προϊόν χρησιμοποιείται, τότε αρχίζει να παρέχει ανατροφοδότηση και η λίστα του ανεκτέλεστου υπόλοιπου γίνεται ολοένα και μεγαλύτερη, οι απαιτήσεις δεν σταματούν ποτέ να αλλάζουν καθώς τροποποιούνται και οι μεταβλητές του περιβάλλοντος. Τα στοιχεία του ανεκτέλεστου υπόλοιπου που πρέπει να ολοκληρωθούν από τις ομάδες στα επόμενα sprints είναι αρκετά λεπτομερή, ώστε να μπορούν να υλοποιούνται εντός της διάρκειας του sprint για το οποίο έχουν αρχικά σχεδιαστεί. Πολλές ομάδες Scrum μπορούν να δουλεύουν ταυτόχρονα για την ανάπτυξη ενός πληροφοριακού συστήματος και το product backlog είναι αυτό το οποίο θα οργανώσει τις εργασίες τις κάθε ομάδας, ανάλογα με την τεχνολογία και την αρχιτεκτονική που θα χρησιμοποιηθεί.

Οι υπολογισμοί για το χρόνο ολοκλήρωσης του product backlog, όπως φαίνεται στον πίνακα 7, γίνονται κατά τη διάρκεια του release planning, ωστόσο μπορούν να ενημερώνονται οποιαδήποτε χρονική στιγμή. Η ομάδα είναι υπεύθυνη για όλους τους υπολογισμούς, αλλά και ο ιδιοκτήτης του προϊόντος μπορεί να βοηθήσει την ομάδα να επιλέξει συμβιβαστικές λύσεις. Ο ιδιοκτήτης του συστήματος διατηρεί επίσης ενημερωμένη ανά πάσα χρονική στιγμή τη λίστα του ανεκτέλεστου υπόλοιπου προϊόντος καθώς επίσης και του burndown γραφήματος.

Στοιχείο Ανεκτέλεστου Υπόλοιπου	Εκτίμηση
Δικαίωμα κράτησης	5
Ακύρωση κράτησης	3
Αλλαγή ημερομηνίας	8
Εκτέλεση αναφοράς	3
Βελτίωση χειρισμού εξαιρέσεων	8
...	50
...	30

Πίνακας 7: Παράδειγμα ανεκτέλεστου υπόλοιπου προϊόντος

Η λίστα του ανεκτέλεστου υπόλοιπου του συστήματος περιέχει τα ακόλουθα πεδία:

[Henrik Kniberg, “Scrum and XP from the Trenches”, 2006, p 7]

- **ID:** Πρόκειται για ένα μοναδικό αναγνωριστικό που κάθε φορά αυξάνεται κατά μια μονάδα. Αυτό γίνεται συνήθως για να αποφύγουμε το ενδεχόμενο να χάσουμε εργασίες που πρέπει να εκτελεστούν όταν χρειαστεί να τροποποιήσουμε τη λίστα προτεραιοτήτων.
- **Όνομα:** Είναι μια σύντομη και περιεκτική ονομασία των ανεκτέλεστων εργασιών που διευκολύνει τους developers και τον ιδιοκτήτη του συστήματος να κατανοήσουν ανά πάσα στιγμή σε ποια ακριβώς εργασία αναφερόμαστε.
- **Σημασία:** Πρόκειται για τη βαθμολογία από πλευράς του ιδιοκτήτη για μια συγκεκριμένη εργασία. Συγκεκριμένα όσο πιο υψηλός είναι ο βαθμός σημασίας, τόσο πιο σημαντική είναι η εργασία που θα εκτελεστεί.
- **Αρχικός υπολογισμός:** Η ομάδα αρχικά εκτιμά το χρονικό διάστημα που θα χρειαστεί για την υλοποίηση μιας εργασίας σε ανθρωποημέρες αν δουλέψει μια τυπική Scrum ομάδα. Θα μπορούμε να αναφερόμαστε στον αριθμό των ανθρωποημερών και ως story points.
- **Κλιμάκωση εργασιών:** Στο σημείο αυτό περιγράφουμε τις ενέργειες που πρέπει να κάνουμε ώστε να υλοποιηθεί σωστά μια εργασία και να ολοκληρωθεί ένα τμήμα της εφαρμογής. Θα μπορούσαμε να πούμε ότι είναι ο ψευδοκώδικας του συστήματός μας.
- **Σημειώσεις:** Οποιοσδήποτε άλλες πληροφορίες και διευκρινίσεις δηλώνονται ως σημειώσεις.

Πρόκειται για τα έξι σημαντικότερα πεδία που χρησιμοποιούνται σε κάθε sprint. Πιο συγκεκριμένα τα πεδία «ID», «όνομα» και «σημασία» πρέπει να δηλώνονται πριν τη συνάντηση σχεδιασμού του sprint (sprint planning meeting), μπορεί όμως να αλλάξουν κατά τη διάρκεια του meeting. Ο αρχικός υπολογισμός και το πεδίο

κλιμάκωση πρέπει επίσης να συμπληρώνονται και να δηλώνονται με σαφήνεια πριν μια εργασία ξεκινήσει να υλοποιείται σε κάποιο sprint. Οι παραπάνω ενέργειες θα πρέπει να γίνονται σε ένα έγγραφο Excel στο οποίο θα πρέπει να έχουν πρόσβαση τόσο ο ιδιοκτήτης όσο και η ομάδα ώστε να μπορούν να το τροποποιήσουν σε περίπτωση που χρειαστεί. Ο παρακάτω πίνακας παρουσιάζει τα πεδία που θα πρέπει να δηλώνονται στη λίστα εργασιών του ανεκτέλεστου υπόλοιπου του συστήματος.

ΑΝΕΚΤΕΛΕΣΤΟ ΥΠΟΛΟΙΠΟ ΠΡΟΪΟΝΤΟΣ					
ID	Όνομα	Σημασία	Εκτίμηση	Κλιμάκωση	Σημειώσεις
1	Κατάθεση	30	5	Ταυτοποιούμε στο σύστημα, κάνουμε μια δοκιμαστική κατάθεση και βλέπουμε αν το υπόλοιπο έχει αυξηθεί	Χρειαζόμαστε ένα UML διάγραμμα ακολουθίας. Δεν χρειάζεται να ανησυχούμε για την κωδικοποίηση προς το παρόν
2	Ιστορικό συναλλαγών	10	9	Ταυτοποιούμε, κάνουμε μια συναλλαγή και τσεκάρουμε αν η τελευταία συναλλαγή εμφανίζεται στο ιστορικό	Προσπαθούμε να αποφύγουμε τα μεγάλα Database Queries

Πίνακας 8: Πεδία του ανεκτέλεστου υπόλοιπου

Μερικές φορές χρησιμοποιούμε επιπρόσθετα πεδία κυρίως για να διευκολύνουμε τον ιδιοκτήτη του συστήματος να ταξινομήσει τις προτεραιότητές του. Τα πεδία αυτά είναι τα ακόλουθα: [Henrik Kniberg, “Scrum and XP from the Trenches”, 2006, p 8]

- Παρακολούθηση: Ο ιδιοκτήτης της εφαρμογής φιλτράρει τις εργασίες και θέτει προτεραιότητες (χαμηλή, μεσαία, υψηλή).
- Συνιστώσες: Στο σημείο αυτό η ομάδα και ο ιδιοκτήτης καθορίζουν τα τεχνικά χαρακτηριστικά που θα χρησιμοποιηθούν για την υλοποίηση των εργασιών. Αυτή η τεχνική είναι ιδιαίτερα χρήσιμη όταν στο project υπάρχουν πολλές ομάδες.
- Αιτών: Στο πεδίο αυτό αποθηκεύουμε τον πελάτη ο οποίος αρχικά αιτήθηκε την υλοποίηση μιας συγκεκριμένης εργασίας ώστε να έχουμε τη δυνατότητα να του παρέχουμε ανατροφοδότηση σχετικά με την πρόοδο της εργασίας.
- Bug ID: Αν υπάρχει ξεχωριστό σύστημα διαχείρισης σφαλμάτων (π.χ. Jira), μπορούμε να σημειώνουμε την άμεση σχέση μεταξύ του ανεκτέλεστου υπόλοιπου και ενός ή περισσότερων δηλούμενων σφαλμάτων.

5.2 ΣΥΝΑΝΤΗΣΗ ΣΧΕΔΙΑΣΗΣ SPRINT

Για να είμαστε απόλυτα προετοιμασμένοι για τη συνάντηση σχεδιασμού θα πρέπει να έχει δημιουργηθεί ένα ανεκτέλεστο υπόλοιπο και να υπάρχει ένας ιδιοκτήτης για το σύστημα. Ο ιδιοκτήτης θα πρέπει να έχει κατανοήσει κάθε εργασία της λίστας του ανεκτέλεστου υπόλοιπου. Σε ορισμένες περιπτώσεις άλλα μέλη της ομάδας μπορούν να προσθέτουν εργασίες, όμως σε κάθε περίπτωση ο ιδιοκτήτης είναι εκείνος που θέτει τις προτεραιότητες. Ο ιδιοκτήτης δεν χρειάζεται να γνωρίζει το λόγο για τον οποίο κάθε εργασία πρέπει να υλοποιηθεί, είναι υποχρεωμένος να γνωρίζει όμως το λόγο για τον οποίο η κάθε εργασία βρίσκεται στη συγκεκριμένη θέση.

Το sprint planning meeting παρακολουθείται από τον ιδιοκτήτη του συστήματος, τον Scrum Master και από όλη την ομάδα. Εκτός από τα ενδιαφερόμενα μέρη, μπορεί να παρευρίσκονται μετά από πρόσκληση της ομάδας και άλλα άτομα. Κατά τη διάρκεια της συνάντησης, ο ιδιοκτήτης της εφαρμογής περιγράφει τις απαιτήσεις με σειρά προτεραιότητας. Η ομάδα Scrum υποβάλλει αρκετά ερωτήματα ώστε να δημιουργηθεί ένα επαρκές ιστορικό περιγραφών για τις ανεκτέλεστες εργασίες του συστήματος και να δημιουργηθούν τα λεπτομερή καθήκοντα των sprints.

Η συνάντηση σχεδιασμού του sprint γίνεται όταν μια επανάληψη έχει σχεδιαστεί και μπορεί να διαρκέσει μέχρι και οχτώ ώρες για ένα sprint διάρκειας ενός μήνα. Για sprints μικρότερης διάρκειας αφιερώνουμε αναλογικά λιγότερο χρόνο, για παράδειγμα για ένα sprint δύο εβδομάδων ένα sprint planning meeting θα μπορούσε να διαρκέσει μέχρι και τέσσερις ώρες. Η συνάντηση σχεδιασμού αποτελείται από δύο μέρη, στο πρώτο μέρος αποφασίζεται τι θα υλοποιηθεί κατά τη διάρκεια του sprint και στο δεύτερο μέρος η ομάδα αποφασίζει πως θα φτάσει στο μέγιστο δυνατό επίπεδο της λειτουργικότητας στο τέλος του sprint.

Στο πρώτο μέρος, ο ιδιοκτήτης του συστήματος παρουσιάζει στην ομάδα τις σημαντικότερες προτεραιότητες από τη λίστα του ανεκτέλεστου υπόλοιπου προϊόντος και συνεργάζεται μαζί της για τον καθορισμό του προβλεπόμενου επιπέδου λειτουργικότητας που πρέπει να επιτευχθεί. Ο ιδιοκτήτης δεν χρειάζεται να περιγράψει κάθε λεπτομέρεια του ανεκτέλεστου υπόλοιπου του συστήματος αλλά να είναι προετοιμασμένος στη συνάντηση σχεδίασης ώστε να αναλύσει όλα όσα είναι απαραίτητα να εκτελεστούν για παράδειγμα στα δύο επόμενα sprints. Αν υποθέσουμε ότι η ομάδα ολοκληρώνει πέντε καθήκοντα από το ανεκτέλεστο υπόλοιπο, τότε ο ιδιοκτήτης θα πρέπει να είναι σε θέση στη συνάντηση σχεδίασης να αναλύσει τις δέκα σημαντικότερες εργασίες. Τα inputs για τη συνάντηση σχεδιασμού είναι το ανεκτέλεστο υπόλοιπο του συστήματος, η πρόοδος που έχει μέχρι τώρα ολοκληρωθεί, η ικανότητα της ομάδας και η εκπλήρωση των εργασιών κατά τη διάρκεια του προηγούμενου sprint. Το σύνολο των εργασιών που πρόκειται να ολοκληρωθούν αποφασίζεται αποκλειστικά από την ομάδα. Στη συνέχεια καθορίζεται ο σκοπός του sprint ο οποίος παρέχει καθοδήγηση στην ομάδα σχετικά

με τη λειτουργικότητα που πρέπει να παρέχει το συγκεκριμένο τμήμα της εφαρμογής.

Στο δεύτερο μέρος η ομάδα θα πρέπει να επικεντρωθεί ώστε να μετατρέψει το ανεκτέλεστο υπόλοιπο σε ολοκληρωμένες εργασίες. Για να γίνει αυτό η ομάδα σχεδιάζει τις εργασίες που πρέπει να γίνουν και στη συνέχεια καθορίζει τα καθήκοντα που θα εκτελεστούν. Τα καθήκοντα είναι λεπτομερή τμήματα εργασιών που πρέπει να γίνουν ώστε να έχουμε ένα ολοκληρωμένο τμήμα λογισμικού της εφαρμογής. Τα καθήκοντα και οι εργασίες που πρέπει να ολοκληρωθούν μέσα στο τρέχον sprint, ονομάζεται ανεκτέλεστο υπόλοιπο του sprint (sprint backlog). Ο ιδιοκτήτης του προϊόντος παρευρίσκεται και στο δεύτερο μέρος του Sprint Planning Meeting ώστε να αποσαφηνίσει το ανεκτέλεστο υπόλοιπο της εφαρμογής και για να κάνει τους απαραίτητους σχεδιασμούς. Σε περίπτωση που η ομάδα έχει περισσότερο ή λιγότερο φόρτο εργασίας, τότε μπορεί να διαπραγματευτεί με τον ιδιοκτήτη του συστήματος για το ανεκτέλεστο υπόλοιπο. Συνοψίζοντας συμπεραίνουμε ότι υπάρχουν δύο αντικείμενα τα οποία προκύπτουν από τη συνάντηση σχεδίασης του sprint.

- Ο στόχος του sprint.
- Το ανεκτέλεστο υπόλοιπο του sprint.

Ένας στόχος του sprint είναι μια μικρή περιγραφή μιας ή δύο προτάσεων για το τι ακριβώς σχεδιάζει να επιτύχει η ομάδα κατά τη διάρκεια του sprint. Ο στόχος αυτός πηγάζει με συνεργασία της ομάδας και του ιδιοκτήτη του προϊόντος. Για μια εφαρμογή ηλεκτρονικού εμπορίου κάποιοι στόχοι θα μπορούσαν να είναι οι παρακάτω:

- Υλοποίηση του καλαθιού αγορών που περιέχει τις επιλογές προσθήκη, αφαίρεση, ενημέρωση.
- Η διαδικασία της αποσύνδεσης, παραλαβής του εμπορεύματος και το περιτύλιγμα σε συσκευασία δώρου.

Ο στόχος του sprint μπορεί να χρησιμοποιηθεί για μια γρήγορη αναφορά σε όσους βρίσκονται εκτός του sprint. Υπάρχουν πάντοτε εκείνοι που θέλουν να μάθουν για το τι ακριβώς υλοποιεί η ομάδα αυτή τη στιγμή, αλλά και κάποιοι άλλοι που δεν χρειάζεται να ακούσουν για το ανεκτέλεστο υπόλοιπο με λεπτομέρεια. Η επιτυχία του sprint στη συνέχεια θα αξιολογηθεί κατά τη διάρκεια απολογισμού της συνεδρίασης του sprint σε σχέση με το στόχο του sprint και όχι σε σχέση με κάθε συγκεκριμένο καθήκον που επιλέγεται από το ανεκτέλεστο υπόλοιπο της εφαρμογής.

Το ανεκτέλεστο υπόλοιπο του sprint είναι μια λίστα του ανεκτέλεστου υπόλοιπου του συστήματος για την οποία η ομάδα δεσμεύεται ότι θα τα φέρει εις πέρας. Ένα σημείο που πρέπει να τονιστεί ιδιαίτερα είναι το γεγονός ότι η ομάδα είναι εκείνη που επιλέγει πόση εργασία μπορεί να προσφέρει μέσα στο επόμενο sprint, ο ιδιοκτήτης του συστήματος δεν ασχολείται καθόλου με αυτό. Αυτό στο οποίο ελπίζει είναι ότι η ομάδα θα καταφέρει να ολοκληρώσει όσα υποσχέθηκε (ή και περισσότερα), αλλά εξαρτάται από την ομάδα ο καθορισμός του τι μπορεί να προσφέρει μέσα στο sprint.

Επομένως γίνεται εύκολα αντιληπτό ότι η συνάντηση σχεδίασης του sprint μπορεί να χωριστεί σε δύο διαφορετικές διαδικασίες, την ιεράρχηση του sprint και τη σχεδίαση του sprint. Κάθε μία από τις παραπάνω διαδικασίες περιγράφει τις ενέργειες οι οποίες αναφέρονται παρακάτω:

- Ιεράρχηση sprint
 - Ανάλυση και αξιολόγηση του ανεκτέλεστου υπόλοιπου του συστήματος.
 - Επιλογή στόχου του sprint.
- Σχεδίαση sprint
 - Καθορισμός για το πώς θα επιτευχθεί ο στόχος του sprint.
 - Δημιουργία του ανεκτέλεστου υπόλοιπου του sprint με βάση τα στοιχεία από το ανεκτέλεστο υπόλοιπο του συστήματος.
 - Εκτίμηση του χρόνου διάρκειας για το ανεκτέλεστο υπόλοιπο του sprint (κατά προτίμηση σε ώρες).

Η συνάντηση σχεδιασμού του sprint είναι πιθανόν το πιο σημαντικό γεγονός στη Scrum. Μια άσχημα εκτελεσμένη συνάντηση σχεδιασμού, μπορεί να έχει αρνητικές επιπτώσεις σε όλο το πληροφοριακό σύστημα. Ο σκοπός του sprint planning meeting είναι να δώσει στην ομάδα αρκετές πληροφορίες για μερικές εβδομάδες εργασίας. Τα αναμενόμενα αποτελέσματα από μια συνάντηση σχεδιασμού μπορεί να είναι τα ακόλουθα:

- Ο καθορισμός του στόχου του sprint.
- Μια ολοκληρωμένη λίστα των μελών της ομάδας.
- Το ανεκτέλεστο υπόλοιπο του sprint.
- Μια εκτίμηση σχετικά με την ημερομηνία ολοκλήρωσης του sprint.
- Ο καθορισμός του τύπου και του χρόνου του Daily Scrum.

Ένα συνηθισμένο πρόβλημα που αντιμετωπίζουν συνήθως οι νέες ομάδες στη συνάντηση σχεδιασμού είναι ότι ενώ πλησιάζουν στο τέλος της συνάντησης, δεν έχουν ακόμη καθορίσει ούτε το στόχο του sprint ούτε το ανεκτέλεστο υπόλοιπο του sprint. Το πρόβλημα αυτό συμβαίνει γιατί ο χρόνος διεξαγωγής του sprint planning meeting δεν ήταν ο σωστός (είτε πολύ μικρός, είτε πολύ μεγάλος). Είναι πολύ σημαντικό να αφιερώνεται ο σωστός χρόνος στη συνάντηση σχεδιασμού. Το να παρατείνεται η χρονική διάρκεια ενός planning meeting όταν η συνάντηση τελειώνει και δεν έχουν οριστεί ο στόχος και η λίστα ανεκτέλεστου υπόλοιπου δεν είναι σωστή πρακτική. Η χρονική παράταση μπορεί να οδηγήσει σε βεβαιασμένες εκτιμήσεις γεγονός που θα έχει αρνητικές επιπτώσεις τόσο στο sprint όσο και στο σύνολο του συστήματος. Ο σωστός χρονικός προσδιορισμός εξαρτάται από σύνολο των εργασιών που θα υλοποιηθούν κατά τη διάρκεια του επόμενου sprint σε συνδυασμό με το βαθμό της πολυπλοκότητάς τους.

5.2.1 VELOCITY: ΜΙΑ ΧΡΗΣΙΜΗ ΠΑΡΑΜΕΤΡΟΣ ΓΙΑ ΑΚΡΙΒΕΙΣ ΥΠΟΛΟΓΙΣΜΟΥΣ

Μια πολύ χρήσιμη τεχνική που εφαρμόζεται κατά τη διάρκεια της συνάντησης σχεδίασης είναι η ακόλουθη. Όταν η ομάδα και ο ιδιοκτήτης του προϊόντος διαφωνούν για τους χρονικούς υπολογισμούς και τις προτεραιότητες,

χρησιμοποιείται μια παράμετρος για να μετρηθεί ο βαθμός ολοκλήρωσης των εργασιών. Η παράμετρος αυτή είναι η ταχύτητα (velocity) και είναι εύκολη στον υπολογισμό. Ανατρέχουμε στις ανεκτέλεστες εργασίες που σχεδιάστηκαν να υλοποιηθούν κατά τη διάρκεια του προηγούμενου sprint και αθροίζουμε τους αρχικούς χρονικούς υπολογισμούς (story points). Αυτός ο αριθμός αντιπροσωπεύει την υπολογιζόμενη ταχύτητα (estimated velocity) του τελευταίου sprint με μονάδες μέτρησης τα story points. [Henrik Kniberg, “Scrum and XP from the Trenches”, 2006, p 12,13]

Στη συνέχεια πρέπει να δούμε ποιες εργασίες ολοκληρώθηκαν στην πράξη και λειτούργησαν επιτυχώς στο τμήμα της εφαρμογής που παρουσιάστηκε στο τέλος του sprint (actual velocity), μη λαμβάνοντας υπόψη τα καθήκοντα τα οποία υλοποιήθηκαν εν μέρη. Ο αριθμός αυτός αντιπροσωπεύει την πραγματική ταχύτητα. Επομένως η ταχύτητα είναι μια μέτρηση σχετικά με το ποσό της εργασίας που ολοκληρώθηκε σε σχέση με τους αρχικούς χρονικούς υπολογισμούς.

ΤΕΛΕΥΤΑΙΟ SPRINT

ΕΚΤΙΜΩΜΕΝΗ ΤΑΧΥΤΗΤΑ: 65

ΠΡΑΓΜΑΤΙΚΗ ΤΑΧΥΤΗΤΑ: 44

Εκτιμώμενη και πραγματική ταχύτητα [Henrik Kniberg, “Scrum and XP from the Trenches”, 2006, p 13]

Η πολύ υψηλή ή πολύ χαμηλή ταχύτητα εξαρτάται από πολλούς παράγοντες όπως λανθασμένες αρχικές εκτιμήσεις, άπειρους προγραμματιστές και μη προβλεπόμενες δυσμενείς εξελίξεις κατά τη διάρκεια του sprint. Πρόκειται για έναν αριθμό ο οποίος στην πραγματικότητα μας δείχνει τη διαφορά μεταξύ του έργου που είχε σχεδιαστεί να παραχθεί και εκείνου που τελικά παράχθηκε στο τέλος του sprint. Φυσικά δεν υπολογίζονται εργασίες οι οποίες είναι μισοτελειωμένες, αυτό που μας ενδιαφέρει είναι το τμήμα της εφαρμογής που δοκιμάστηκε και λειτούργησε σωστά κατά τη διάρκεια του sprint. Εφόσον έχουμε καταγράψει την εκτιμώμενη και την πραγματική ταχύτητα προχωράμε με τα υπόλοιπα στοιχεία.

Για να υπολογίσουμε τις ανθρωποημέρες σε ένα project, πολλαπλασιάζουμε το μέγεθος της ομάδας με τις συνολικές ημέρες του sprint. Αν για παράδειγμα μια ομάδα έχει 7 άτομα και το sprint έχει διάρκεια 15 ημέρες (3 εργάσιμες εβδομάδες), τότε οι συνολικές ανθρωποημέρες που θα χρειαστούμε είναι 105. Ο σωστός υπολογισμός των ανθρωποημερών είναι πολύ σημαντικός. Εξίσου σημαντικός υπολογισμός είναι και ο βαθμός συγκέντρωσης (focus) της ομάδας. Ωστόσο δεν υπάρχει κάποιος μαθηματικός τύπος που να υπολογίζει το βαθμό συγκέντρωσης, υπολογίζεται κάθε φορά εμπειρικά συγκρίνοντας την εκτιμώμενη με την πραγματική ταχύτητα του προηγούμενου sprint. Χαμηλός βαθμός συγκέντρωσης σημαίνει ότι η ομάδα αναμένει να έχει αρκετές διαταραχές από εξωτερικούς παράγοντες ή κάποια από τα μέλη της να χρειαστεί να απουσιάσουν κάποιες ημέρες κατά τη διάρκεια του sprint. Επομένως μπορούμε να υπολογίσουμε και την εκτιμώμενη ταχύτητα και ως εξής:

TREXON SPRINT

ΣΥΝΟΛΙΚΕΣ ΑΝΘΡΩΠΟΗΜΕΡΕΣ: 80

ΒΑΘΜΟΣ ΣΥΓΚΕΝΤΡΩΣΗΣ: 75%

ΕΚΤΙΜΩΜΕΝΗ ΤΑΧΥΤΗΤΑ: 65

Υπολογισμός εκτιμώμενης ταχύτητας Henrik Kniberg, "Scrum and XP from the Trenches", 2006, p 13]

Η εκτιμώμενη ταχύτητα του sprint είναι πολύ σημαντική για δύο λόγους:

1. Δρα σαν ένα φυσικό όριο για την εργασία που η ομάδα θα πρέπει να δεσμευτεί ότι θα ολοκληρώσει. Εφόσον η ομάδα έχει αποφασίσει ποιες εργασίες από το ανεκτέλεστο υπόλοιπο θα υλοποιήσει, τότε θα πρέπει να αθροίσει τους αρχικούς χρονικούς υπολογισμούς και να βεβαιωθεί ότι δεν ξεπερνάνε την εκτιμώμενη ταχύτητα. Σε διαφορετική περίπτωση θα πρέπει να επαναυπολογίσει το βαθμό συγκέντρωσης, τους χρονικούς υπολογισμούς ή να αφαιρέσει κάποιες από τις εργασίες τις οποίες δεσμεύτηκε ότι θα ολοκληρώσει κατά τη διάρκεια του συγκεκριμένου sprint.
2. Μπορεί να συγκριθεί με την πραγματική ταχύτητα και σε περίπτωση που υπάρξουν μεγάλες αποκλίσεις τότε η ομάδα στα επόμενα sprints θα μπορεί

να υπολογίσει καλύτερα τους χρονικούς υπολογισμούς και τις εργασίες που πρόκειται να ολοκληρωθούν. [Henrik Kniberg, “Scrum and XP from the Trenches”, 2006, p 13]

5.2.2 ΛΙΣΤΕΣ ΕΡΓΑΣΙΩΝ

Ο περισσότερος χρόνος στη συνάντηση σχεδιασμού του sprint έχει να κάνει με το χειρισμό του ανεκτέλεστου υπόλοιπου, προσδιορίζοντας το χρόνο εκτέλεσης των εργασιών, την ταξινόμησή τους με σειρά προτεραιότητας και τη διάσπασή τους σε μικρότερες εργασίες. Μια καλή πρακτική είναι να χωρίσουμε τις εργασίες εκείνες που έχουν την υψηλότερη και εκείνες που έχουν τη χαμηλότερη προτεραιότητα όπως φαίνεται και στην παρακάτω εικόνα.



Εικόνα 15: Λίστα εργασιών ανά προτεραιότητα [Henrik Kniberg, “Scrum and XP from the Trenches”, 2006, p 16]

5.2.3 ΠΡΟΤΕΡΑΙΟΤΗΤΕΣ ΣΤΗ ΣΥΝΑΝΤΗΣΗ ΣΧΕΔΙΑΣΜΟΥ

Είναι αλήθεια ότι η συνάντηση σχεδιασμού περιλαμβάνει πολλές διαδικασίες που πρέπει να ολοκληρωθούν σε συγκεκριμένο χρονικό διάστημα. Όταν ο χρόνος πιέζει και το meeting πρέπει να ολοκληρωθεί πρέπει να έχουμε μια ιεραρχημένη λίστα ενεργειών που πρέπει να γίνουν και είναι οι παρακάτω:

1. Καθορισμός του στόχου και του χρονικού προσδιορισμού ολοκλήρωσης ενός τμήματος της εφαρμογής. Αυτό είναι το ελάχιστο που πρέπει να γίνει για να ξεκινήσει το sprint. Η ομάδα έχει ένα στόχο και ένα χρονοδιάγραμμα που

πρέπει να ακολουθήσει για να ολοκληρώσει τις εργασίες από το ανεκτέλεστο υπόλοιπο του συστήματος.

2. Λίστα των ανεκτέλεστων εργασιών που η ομάδα έχει δεσμευτεί να ολοκληρώσει κατά τη διάρκεια ενός συγκεκριμένου sprint.
3. Υπολογισμός του χρόνου για κάθε ανεκτέλεστη εργασία του sprint.
4. Υπολογισμός της ταχύτητας ως μια τεχνική πραγματικού ελέγχου για το πλάνο του sprint (περιέχει λίστα των μελών και των δεσμεύσεων τους, διαφορετικά δεν μπορούμε να υπολογίσουμε την ταχύτητα).
5. Καθορισμός του τύπου και του χρόνου για τις καθημερινές Scrum συναντήσεις.
6. Διάσπαση των ανεκτέλεστων εργασιών σε επιμέρους καθήκοντα όπου αυτό κρίνεται απαραίτητο.

5.2.4 ΚΕΝΑ ΔΙΑΣΤΗΜΑΤΑ ΜΕΤΑΞΥ ΤΩΝ SPRINTS

Δεν είναι πάντοτε εφικτό να τρέχουμε συνέχεια να προλάβουμε και να ολοκληρώνουμε συνεχώς εργασίες. Μέσα σε ένα Scrum project πρέπει να βρούμε ένα μικρό χρονικό διάστημα να χαλαρώνουμε πριν ξεκινήσουμε την εκτέλεση του επόμενου sprint. Το καταλληλότερο χρονικό διάστημα είναι μετά το sprint demo και την ανακεφαλαίωση του sprint όπου τόσο η ομάδα όσο και ο ιδιοκτήτης του συστήματος θα πρέπει να αφομοιώσουν πλήρως ότι ολοκληρώθηκε και ότι απομένει να υλοποιηθεί. Αν ξεκινήσουμε απευθείας το σχεδιασμό του επόμενου sprint, ο ιδιοκτήτης δεν θα έχει προλάβει να προσαρμόσει τις προτεραιότητες των εργασιών και η ομάδα δεν θα μπορεί να λειτουργήσει αποτελεσματικά. Θα μπορούσαμε να πούμε ότι είναι σωστή πρακτική η ανακεφαλαίωση του sprint και ο σχεδιασμός του επόμενου να μην ξεκινάει την ίδια ημέρα, όπως φαίνεται και στους πίνακες 9 και 10 που ακολουθούν. [Henrik Kniberg, “Scrum and XP from the Trenches”, 2006, p 49]

Δευτέρα		Τρίτη
9:00 – 10:00	Sprint Demo	9:00 – 13:00 Σχεδιασμός επόμενου Sprint
10:00 – 11:00	Ανακεφαλαίωση Sprint	

Πίνακας 9: Σωστή πρακτική ανακεφαλαίωσης και σχεδιασμού sprint

Παρασκευή		Σάββατο	Κυριακή	Δευτέρα
9:00 – 10:00	Sprint Demo			9:00 – 13:00 Σχεδιασμός επόμενου
10:00 – 11:00	Ανακεφαλαίωση			

Πίνακας 10: Ίδανική πρακτική ανακεφαλαίωσης και σχεδιασμού sprint

5.3 ΤΟ ΑΝΕΚΤΕΛΕΣΤΟ ΥΠΟΛΟΙΠΟ ΤΟΥ SPRINT

Το ανεκτέλεστο υπόλοιπο του sprint αποτελείται από καθήκοντα που πρέπει να μετατραπούν σε ολοκληρωμένες (done) εργασίες κατά τη διάρκεια του sprint. Τα καθήκοντα αυτά αναπτύχθηκαν κατά τη διάρκεια συνάντησης σχεδιασμού του sprint (sprint planning meeting) και πρέπει να ολοκληρωθούν έτσι ώστε να επιτευχθεί και ο στόχος του sprint. Η ομάδα τροποποιεί το ανεκτέλεστο υπόλοιπο κατά τη διάρκεια του sprint καθώς μπορεί να διαπιστώσει ότι περισσότερες ή λιγότερες ενέργειες είναι απαραίτητες σε σχέση με αυτές που είχαν αρχικά υπολογιστεί. Όταν κάποια καθήκοντα δεν κριθούν απαραίτητα, τότε διαγράφονται από το ανεκτέλεστο υπόλοιπο του sprint. Καθώς οι εργασίες εξελίσσονται η ομάδα υπολογίζει ταυτόχρονα το υπολειπόμενο χρονικό διάστημα ολοκλήρωσης και ενημερώνει των πίνακα εργασιών. Μόνο η ομάδα μπορεί να τροποποιήσει το περιεχόμενο, τους χρονικούς υπολογισμούς και το sprint backlog κατά τη διάρκεια του sprint. Το ανεκτέλεστο υπόλοιπο του sprint αποτελεί την πραγματική εικόνα της ομάδας, αποτυπώνει τα πλάνα των εργασιών που πρόκειται να ολοκληρωθούν και για το λόγο αυτό ανήκει αποκλειστικά στην ομάδα.

Το ανεκτέλεστο υπόλοιπο sprint είναι μια λίστα εργασιών που προσδιορίζεται από την ομάδα Scrum κατά τη διάρκεια σχεδίασης του sprint. Η ομάδα επιλέγει ένα τυχαίο αριθμό εργασιών, συνήθως από τις περιγραφές των χρηστών και προσδιορίζει τις εργασίες που απαιτούνται για να ολοκληρωθεί η κάθε περιγραφή.

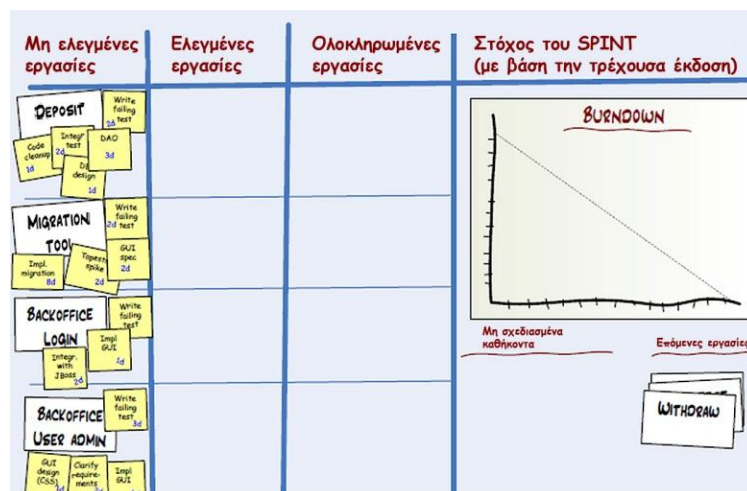
Οι περισσότερες ομάδες υπολογίζουν τις ώρες που απαιτούνται προκειμένου να ολοκληρωθεί η κάθε εργασία. Είναι σημαντικό η ομάδα να επιλέγει τις εργασίες και το μέγεθος του ανεκτέλεστου υπόλοιπου του sprint που θα υλοποιήσει μιας και είναι η μόνη υπεύθυνη για να ολοκληρώσει τα καθήκοντα για τα οποία έχει δεσμευτεί. Το ανεκτέλεστο υπόλοιπο πολύ συχνά διατηρείται σε ένα υπολογιστικό φύλλο και έχει τη μορφή του πίνακα που ακολουθεί.

Περιγραφή χρήστη	Καθήκοντα	1 ^η Ημέρα	2 ^η Ημέρα	3 ^η Ημέρα	4 ^η Ημέρα	...
Ανάγνωση profile	Κωδικοποίησε...	8	4	8	0	
	Σχεδίασε...	16	12	10	4	
	Συνάντηση με...	8	16	16	11	
	Σχεδίασε UI...	12	6	0	0	
	Αυτοματοποίησε έλεγχους...	4	4	1	0	
	Κωδικοποίησε το επόμενο ...	8	8	8	8	
Ενημέρωση πληροφοριών λογαριασμού	Ενημέρωσε ...	6	6	4	0	
	Βρες τη λύση ...	12	6	0	0	
	Γράψε το πλάνο ελέγχου ...	8	8	4	0	
	Υλοποίησε ...	12	12	10	6	
	Ανάπτυξε ...	8	8	8	4	

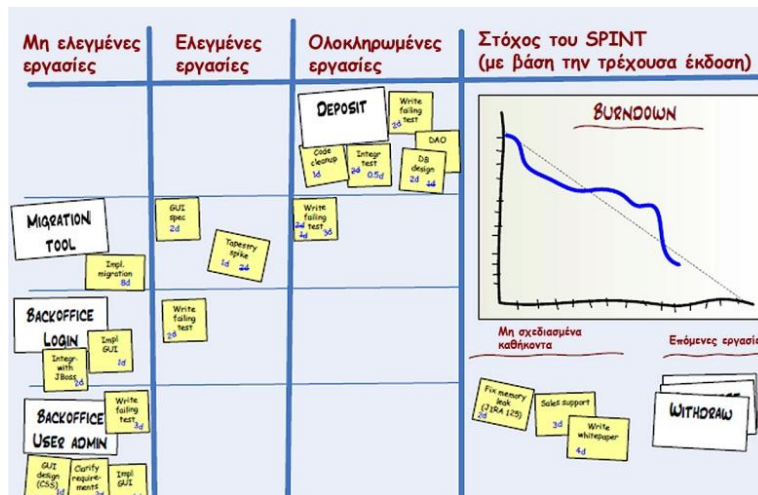
Πίνακας 11: Παράδειγμα ανεκτέλεστου υπόλοιπου sprint

Κάθε μέλος της ομάδας μπορεί να προσθέσει, να διαγράψει ή να αλλάξει το ανεκτέλεστο υπόλοιπο sprint. Εάν η εργασία δεν είναι σαφής, καθορίζουμε ένα στοιχείο στο sprint backlog, με μεγαλύτερο χρονικό διάστημα και ενημερώνουμε την υπολειπόμενη ανεκτέλεστη εργασία προκειμένου να έχουμε καλύτερη πληροφόρηση.

Αφού έχει ολοκληρωθεί η συνάντηση σχεδιασμού του Sprint και είναι όλοι ενήμεροι για το τι θα υλοποιηθεί στο επόμενο sprint, ο Scrum Master θα πρέπει να δημιουργήσει το ανεκτέλεστο υπόλοιπο του sprint, διαδικασία η οποία πρέπει να γίνει μετά τη συνάντηση σχεδιασμού του sprint και πριν από την πρώτη καθημερινή συνάντηση Scrum. Οι πιο διαδεδομένες πρακτικές για τη δημιουργία του ανεκτέλεστου υπόλοιπου του sprint είναι τα λογιστικά φύλλα (Excel) και οι πίνακες καθηκόντων (taskboards). Υπάρχουν αρκετά πρότυπα λογιστικών φύλλων όπως τα burndown charts που μπορούν να χρησιμοποιηθούν για τη δημιουργία του ανεκτέλεστου υπόλοιπου του sprint. Ωστόσο έχει αποδειχθεί ότι είναι πολύ πιο πρακτική η τεχνική του πίνακα καθηκόντων. Το μόνο που χρειάζεται είναι ένας ανεκμετάλλευτος τοίχος στον χώρο εργασίας μας στον οποίο μπορούμε να αναρτήσουμε ένα μεγάλο κομμάτι χαρτιού στο οποίο μπορούμε να σημειώνουμε την εξέλιξη των εργασιών. Στην αρχή ενός sprint ο πίνακας καθηκόντων μπορεί να έχει την παρακάτω μορφή, η οποία σαφώς αλλάζει καθώς το sprint εξελίσσεται.



Εικόνα 16: Παράδειγμα πίνακα καθηκόντων 1 [Henrik Kniberg, "Scrum and XP from the Trenches", 2006, p 30,31]



Εικόνα 17: Παράδειγμα πίνακα καθηκόντων 2[Henrik Kniberg, “Scrum and XP from the Trenches”, 2006, p 30,31]

5.4 ΟΙ ΚΑΘΗΜΕΡΙΝΕΣ SCRUM ΣΥΝΑΝΤΗΣΕΙΣ

Κάθε ημέρα ενός sprint η ομάδα έχει καθημερινές συναντήσεις (daily scrum). Οι συναντήσεις αυτές συνήθως πραγματοποιούνται στην ίδια τοποθεσία, την ίδια ώρα κάθε ημέρα. Ιδανικά σε κάθε περίπτωση οι Scrum συναντήσεις πραγματοποιούνται τα πρωινά, καθώς βοηθούν στον καθορισμό του περιεχομένου των εργασιών της ημέρας που ξεκινάει. Οι συναντήσεις αυτές έχουν διάρκεια έως δεκαπέντε λεπτά, κάνοντας τη συζήτηση σύντομη και περιεκτική.

Η Scrum παρέχει ένα ειδικό καθεστώς και επιβάλλει ένα κανόνα σύμφωνα με τον οποίο επιτρέπεται να συμμετέχουν στη συζήτηση μιας καθημερινής Scrum συνάντησης μόνο όσοι έχουν δεσμευτεί να προσφέρουν για την υλοποίηση της εφαρμογής. Όλα τα μέλη της ομάδας υποχρεούνται να παρακολουθήσουν την καθημερινή συνάντηση, δεδομένου ότι και ο Scrum Master και ο ιδιοκτήτης του συστήματος δεσμεύονται ως μέλη της ομάδας. Οποιοσδήποτε άλλος (π.χ. χρήστης του συστήματος, προγραμματιστής ενός άλλου project) μπορεί να παρακολουθήσει τη συνάντηση, χωρίς όμως να παρέμβει στη συζήτηση. Αυτό κάνει τις καθημερινές Scrum συναντήσεις ένα εξαιρετικό τρόπο διάδοσης πληροφοριών για την ομάδα. Αν μας ενδιαφέρει να ενημερωθούμε για το πού βρίσκονται οι εξελίξεις τότε θα πρέπει να παρακολουθήσουμε τις καθημερινές συναντήσεις.

Οι καθημερινές Scrum συναντήσεις δεν χρησιμοποιούνται ως συναντήσεις επίλυσης προβλημάτων ή ζητημάτων που προκύπτουν, αυτά τα ζητήματα αντιμετωπίζονται από την αρμόδια υποομάδα μετά την καθημερινή συνάντηση. Κατά τη διάρκεια της καθημερινής συνάντησης κάθε μέλος της ομάδας απαντά στα εξής τρία ερωτήματα:

- Τι έκανε χθες.
- Τι θα κάνει σήμερα.
- Τι εμπόδια μπορεί να προκύψουν.

Με το να εστιάσουμε στο τι κάναμε και στο τι απομένει να κάνουμε, η ομάδα έχει μια σωστή κατανόηση για το έργο που της απομένει. Οι καθημερινές συναντήσεις δεν είναι συναντήσεις ενημέρωσης της υπάρχουσας κατάστασης στην οποία ο ιδιοκτήτης συλλέγει πληροφορίες για το ποιος βρίσκεται πίσω από το χρονοδιάγραμμα. Πρόκειται για συναντήσεις στις οποίες τα μέλη της ομάδας δεσμεύονται ο ένας στον άλλο, αν για παράδειγμα ο developer μιας εφαρμογής πει ότι θα τελειώσει ένα συγκεκριμένο τμήμα της εφαρμογής, όλοι θα ξέρουν ότι στην αυριανή συνάντηση θα επιβεβαιώσει ότι ολοκλήρωσε αυτό που ακριβώς υποσχέθηκε.

Οποιαδήποτε εμπόδια προκύψουν είναι υπεύθυνος να τα επιλύσει ο Scrum Master όσο πιο γρήγορα μπορεί. Σε περίπτωση που δεν μπορεί να κάνει κάτι τέτοιο άμεσα (π.χ. συνήθως τα πιο τεχνικά θέματα), έχει την ευθύνη να σιγουρευτεί ότι θα το πραγματοποιήσει κάποιος από την ομάδα. Η συντριπτική πλειοψηφία των ομάδων πραγματοποιούν την καθημερινή συνάντηση έχοντας κάποιο άτομο που μπορεί να απαντήσει στα τρία ερωτήματα με τη σειρά.

5.4.1 ΚΑΘΟΡΙΣΜΟΣ ΤΟΥ ΤΟΠΟΥ ΚΑΙ ΤΟΥ ΧΡΟΝΟΥ ΤΩΝ SCRUM ΣΥΝΑΝΤΗΣΕΩΝ

Η πρώτη καθημερινή Scrum είναι απαραίτητη μιας και καθορίζει το τι θα πρέπει να κάνει ο καθένας μέσα στην ομάδα. Χωρίς την καθημερινή συνάντηση Scrum το sprint μπορεί να αποτύχει ή μην έχει το ανάλογο ξεκίνημα ή να μην αποδώσει τα αναμενόμενα. Οι ομάδες θα πρέπει να καθορίσουν όμως το χρόνο και τον τόπο

διεξαγωγής των καθημερινών Scrum συναντήσεων. Όταν οι καθημερινές Scrums πραγματοποιούνται τα απογεύματα, τα μέλη της ομάδας ίσως να μην έχουν την ανάλογη διάθεση ή τη συγκέντρωση μετά από μια κοπιαστική ημέρα εργασίας.

Από την άλλη πλευρά, όταν οι καθημερινές Scrum συναντήσεις γίνονται το πρωί, η ομάδα θα πρέπει να θυμηθεί το έργο που υλοποίησε την προηγούμενη ημέρα και να το αναφέρει. Από τις μέχρι τώρα έρευνες έχει αποδειχθεί ότι είναι πιο εύκολο για τις ομάδες να αναφέρουν τις εργασίες που ολοκληρώθηκαν σε σχέση με τα καθήκοντα που έχουν αποφασιστεί από την προηγούμενη ημέρα στη Scrum συνάντηση ότι πρέπει να υλοποιηθούν. Αυτός είναι και ο σημαντικότερος λόγος για τον οποίο οι Scrum Masters αποφασίζουν να πραγματοποιούν τις Daily Scrums κάθε πρωί κάποια στιγμή από τις 9:00 μέχρι τις 10:00 όπου όλα τα μέλη της ομάδας μπορούν να παρευρεθούν.

5.5 ΣΥΝΑΝΤΗΣΗ ΣΧΕΔΙΑΣΜΟΥ ΕΚΔΟΣΗΣ ΛΟΓΙΣΜΙΚΟΥ

Ο λόγος της συνάντησης σχεδιασμού (Release Planning Meeting) έκδοσης είναι να εγκαθιδρύσει ένα πλάνο και τους στόχους τους οποίους οι ομάδες και οι υπόλοιποι οργανισμοί μπορούν να κατανοήσουν. Ο λόγος ύπαρξής του είναι να βρει τον τρόπο ώστε να κάνουμε το σύστημα να λειτουργήσει με τον καλύτερο δυνατό τρόπο, να ξεπεράσουμε τις προσδοκίες και την ικανοποίηση των πελατών καθώς επίσης και να έχουμε τα μέγιστα δυνατά οφέλη για τον οργανισμό (return on investment). Το release πλάνο αποδεικνύει την υψηλότερη προτεραιότητα του ανεκτέλεστου υπόλοιπου του προϊόντος, τους κυριότερους κινδύνους, τη λειτουργικότητα και τα βασικά χαρακτηριστικά της εφαρμογής. Καθιερώνει επίσης την πιθανή ημερομηνία παράδοσης και το κόστος σε περίπτωση που τίποτε δεν αλλάξει.

Τα συστήματα υλοποιούνται στη Scrum μετά από μια σειρά επαναλήψεων, σε κάθε μία από τις οποίες πραγματοποιείται και ένα κομμάτι της συνολικής εφαρμογής. Όταν όλα τα κομμάτια του πληροφοριακού συστήματος έχουν ολοκληρωθεί, τότε το προϊόν μπορεί να βγει σε κυκλοφορία. Οι περισσότεροι οργανισμοί έχουν ένα

πλάνο κυκλοφορίας και στις περισσότερες διαδικασίες ο σχεδιασμός γίνεται στην αρχή της κυκλοφορίας χωρίς να αλλάξει καθώς προχωράμε. Το release planning απαιτεί τον υπολογισμό του χρόνου ολοκλήρωσης της εφαρμογής και τον καθορισμό προτεραιοτήτων στο ανεκτέλεστο υπόλοιπο του προϊόντος. Επίσης καθορίζεται ένας γενικός στόχος και τα πιθανά αποτελέσματα από τη λειτουργία του συστήματος.

Σε ορισμένες περιπτώσεις πρέπει να σχεδιάσουμε περισσότερα από ένα sprint κάθε φορά. Αυτό γίνεται κυρίως για να προσδιορίσουμε το χρονικό διάστημα ολοκλήρωσης του project και για να καθορίσουμε το ύψος του προϋπολογισμού. Με άλλα λόγια ο σχεδιασμός της έκδοσης του λογισμικού (release planning) είναι μια προσπάθεια χρονικού προσδιορισμού όπου το σύστημά μας θα μπορέσει να λειτουργήσει στην πρώτη του έκδοση. Στο σημείο αυτό θα πρέπει επίσης να καθορίσουμε τι θα είναι παραδοτέο μαζί με την πρώτη έκδοση του λογισμικού.

- Όλα τα στοιχεία που έχουν επισημανθεί από τον ιδιοκτήτη με σπουδαιότητα ≥ 100 πρέπει οπωσδήποτε να περιέχονται στην πρώτη έκδοση.
- Τα στοιχεία που έχουν επισημανθεί από τον ιδιοκτήτη με σπουδαιότητα μεταξύ 50 - 99 θα πρέπει να περιέχονται στην πρώτη έκδοση ή στην αμέσως επόμενη.
- Όλα τα στοιχεία που έχουν επισημανθεί από τον ιδιοκτήτη με σπουδαιότητα μεταξύ 25 - 49 απαιτούνται, αλλά μπορούν να υλοποιηθούν στην έκδοση 1.1.
- Όλα τα στοιχεία που έχουν επισημανθεί από τον ιδιοκτήτη με σπουδαιότητα < 25 είναι θεωρητικά και μπορεί να μην χρειαστεί να υλοποιηθούν ποτέ.

Έτσι καταλαβαίνουμε ότι είναι ασφαλές για την ομάδα ανάπτυξης να παραδώσει στην πρώτη έκδοση λογισμικού όλες τις προτεραιότητες με αριθμό μεγαλύτερο του 50. Ότι καταφέρουμε να ολοκληρώσουμε με μικρότερη σπουδαιότητα στην πρώτη

έκδοση λογισμικού δεν είναι απαιτούμενο και αποτελεί έξτρα μόνους για την ομάδα. Ο χρονικός προσδιορισμός της έκδοσης λογισμικού είναι χρήσιμος όταν συμπίπτει με την ημερομηνία παράδοσης του συστήματος, λιγότερο χρήσιμος όταν υπάρχει μια μικρή απόκλιση και εντελώς άχρηστος αν δεν υπάρχει καμία συσχέτιση μεταξύ τους.

Για να είναι ακριβείς οι υπολογισμοί του χρόνου παράδοσης θα πρέπει να λάβουμε υπόψη μας τον παράγοντα ταχύτητα (velocity). Επομένως θα πρέπει να καθορίσουμε τον παράγοντα συγκέντρωσης (focus factor), οποίος προσδιορίζει το πόσο αφοσιωμένη είναι η ομάδα στην ολοκλήρωση του ανεκτέλεστου υπόλοιπου του προϊόντος. Ο παράγοντας συγκέντρωσης δεν είναι ποτέ 100% μιας και η ομάδα μπορεί να εκτελεί μη σχεδιασμένες ενέργειες που απαιτούνται ή να αφιερώνει μέρος του χρόνου βοηθώντας κάποια άλλη ομάδα που αντιμετωπίζει δυσκολίες στην υλοποίηση.

Συνήθως επιλέγεται ως παράγοντας συγκέντρωσης 70%. Αν για παράδειγμα επιλέξουμε ως παράγοντα συγκέντρωσης 50% και έχουμε ένα sprint 3 εβδομάδων (15 ημερών) το οποίο θα υλοποιήσει ομάδα έξι ατόμων, τότε χρειαζόμαστε 90 ανθρωποημέρες (storypoints). Επειδή ο παράγοντας συγκέντρωσης είναι 50%, καταλαβαίνουμε ότι η εκτιμώμενη ταχύτητα είναι 45 ανθρωποημέρες. Έτσι αν για κάθε εργασία του ανεκτέλεστου υπόλοιπου χρειαζόμαστε 5 ημέρες τότε η ομάδα μπορεί στο συγκεκριμένο sprint να ολοκληρώσει με επιτυχία κατά προσέγγιση 9 εργασίες από το ανεκτέλεστο υπόλοιπο.

Για να είναι πιο ακριβής ο χρονικός υπολογισμός πρέπει να συγκρίνουμε την πραγματική με την εκτιμώμενη ταχύτητα στο τέλος του sprint. Αν απαιτηθεί ίσως χρειαστεί να αναπροσαρμόσουμε την εκτιμώμενη ταχύτητα για τα επόμενα sprints σε περίπτωση που παρουσιάζει απόκλιση σε σχέση με την πραγματική ταχύτητα.

Οι δοκιμαστικές εκδόσεις λογισμικού στο τέλος κάθε sprint αποτελούν σημαντικό μέρος της μεθοδολογίας Scrum, όμως αρκετοί έχουν την τάση να τις υποτιμούν.

Μια καλά εκτελεσμένη δοκιμαστική έκδοση λογισμικού μπορεί να έχει σημαντικά πλεονεκτήματα, τα κυριότερα από τα οποία παρουσιάζονται παρακάτω.

- Η ομάδα πιστώνεται την ολοκλήρωση του sprint και αυξάνεται η αυτοπεποίθησή της με αποτέλεσμα τα μέλη της να αισθάνονται πιο όμορφα.
- Τα υπόλοιπα μέλη ενημερώνονται για το έργο της ομάδας.
- Το demo προσφέρει ζωτικής σημασίας πληροφορίες σε όλα τα ενδιαφερόμενα μέρη.
- Οι δοκιμαστικές εκδόσεις είναι (ή θα πρέπει να είναι) ένα συμβάν στο οποίο οι ομάδες μπορούν να αλληλεπιδρούν μεταξύ τους και να συζητούν για την εργασία τους.
- Τα demos είναι ένας τρόπος ολοκλήρωσης των εργασιών στο έπακρο, μπορεί να ολοκληρώνουμε λιγότερες εργασίες αλλά είμαστε βέβαιοι ότι έχουν πραγματικά ολοκληρωθεί.

Αν μια ομάδα είναι περισσότερο ή λιγότερο αναγκασμένη να υλοποιήσει μια δοκιμαστική έκδοση του sprint ακόμα και όταν στο συγκεκριμένο sprint δεν υπάρχει αρκετή λειτουργικότητα, τότε το demo μπορεί να μην έχει τα αναμενόμενα αποτελέσματα και τα σχόλια που θα ακουστούν από τα υπόλοιπα μέλη του project μπορεί να είναι μέτρια. Τα ενδιαφερόμενα μέρη μπορεί να χάσουν την εκτίμησή τους προς την ομάδα και κάποιοι από αυτούς μπορεί να αισθανθούν ότι χάθηκε σημαντικός χρόνος υλοποιώντας ένα μέτριο demo. Αυτό μπορεί να είναι ένα άσχημο συναίσθημα για την ομάδα όπου μπορεί να λειτουργήσει θετικά για την ομάδα κατά τη διάρκεια του επόμενου sprint, κάνοντας τα μέλη της να συνειδητοποιήσουν ότι πρέπει να παρουσιάσουν ένα ολοκληρωμένο demo στο επόμενο sprint.

6.1 ΑΠΟΛΟΓΙΣΜΟΣ SPRINT

Στο τέλος κάθε sprint ένας απολογισμός (review) για το sprint πραγματοποιείται. Κατά τη συνάντηση αυτή η ομάδα Scrum αποδεικνύει ότι κατάφερε να ολοκληρώσει ότι είχε υποσχεθεί στη διάρκεια του sprint. Συνήθως αυτό μπορεί να γίνεται με την παρουσίαση μιας demo εφαρμογής για το κομμάτι του συστήματος που ολοκληρώθηκε. Η συνεδρίαση για τον απολογισμό του sprint είναι συνήθως λιτή και η παρουσίαση μπορεί να γίνει με την παράθεση Power Point slides. Η προετοιμασία του απολογισμού δεν θα πρέπει να ξεπερνά σε χρόνο τις δύο ώρες και δεν θα πρέπει να αποσπά την προσοχή και τη φυσική ροή εργασίας της ομάδας.

Οι συμμετέχοντες στον απολογισμό του sprint είναι ο ιδιοκτήτης του συστήματος, ο Scrum Master, η ομάδα Scrum, η διοίκηση του οργανισμού, οι πελάτες και οι developers από άλλα παρεμφερή projects. Κατά τη διάρκεια του απολογισμού το έργο αξιολογείται σε σχέση με το στόχο που καθορίστηκε κατά το στάδιο της σχεδίασης του sprint. Ιδανικά η ομάδα πρέπει να έχει ολοκληρώσει κάθε ανεκτέλεστο υπόλοιπο που υποσχέθηκε κατά τη διάρκεια του sprint, αλλά αυτό που έχει μεγαλύτερη σημασία είναι να επιτευχθεί ο γενικός στόχος του.

Κατά τη διάρκεια του απολογισμού η ομάδα ανακεφαλαιώνει τις εργασίες που έχουν ολοκληρωθεί και συζητάει σχετικά με τις επόμενες ενέργειες που θα μπορούσαν να γίνουν. Πρόκειται για μια ανεπίσημη συνάντηση κατά τη διάρκεια της οποίας σχεδιάζεται το τι θα υλοποιηθεί στη συνέχεια. Ο ιδιοκτήτης του συστήματος καθορίζει τις εργασίες που ολοκληρώθηκαν και εκείνες που δεν ολοκληρώθηκαν, ενώ η ομάδα συζητά για το τι πήγε καλά και τι προβλήματα υπήρξαν κατά τη διάρκεια του συγκεκριμένου sprint. Στη συνέχεια η ομάδα παραθέτει το παραχθέν έργο και απαντά σε οποιεσδήποτε απορίες έχουν δημιουργηθεί. Ο ιδιοκτήτης στο τέλος αναφέρει το ανεκτέλεστο υπόλοιπο του προϊόντος όπως έχει μέχρι τώρα τροποποιηθεί καθώς επίσης και την πιθανή

ημερομηνία παράδοσης του πληροφοριακού συστήματος. Ο απολογισμός του sprint παρέχει σημαντικές πληροφορίες για τον ακόλουθο σχεδιασμό του sprint.

6.2 ΑΝΑΚΕΦΑΛΑΙΩΣΗ SPRINT (RETROSPECTIVE)

Μετά τον απολογισμό του sprint και πριν από την επόμενη συνάντηση σχεδιασμού, η ομάδα Scrum πραγματοποιεί την ανακεφαλαίωση του sprint (Sprint Retrospective). Στη διάρκεια αυτής της συνάντησης ο Scrum Master ενθαρρύνει την ομάδα να ανακεφαλαιώσει εντός του γενικού πλαισίου της Scrum και των πρακτικών της, τη διαδικασία της υλοποίησης ώστε να γίνει πιο αποτελεσματική και πιο ευχάριστη στο επόμενο sprint. Ο σκοπός της ανακεφαλαίωσης είναι να επιθεωρήσει πώς το προηγούμενο sprint εξελίχθηκε σε σχέση με τις διαδικασίες, τα εργαλεία που χρησιμοποιήθηκαν και τα άτομα που απασχολήθηκαν σε αυτό. Η ανακεφαλαίωση θα πρέπει να καθορίζει τις κύριες εργασίες που ολοκληρώθηκαν, καθώς επίσης και εκείνες που αν γινότανε διαφορετικά θα μπορούσαν να είχαν κάνει τα πράγματα ακόμη πιο εύκολα. Στο τέλος της ανακεφαλαίωσης η ομάδα Scrum θα πρέπει να έχει καθορίσει δραστικά μέτρα βελτίωσης που θα υλοποιηθούν στο αμέσως επόμενο sprint.

Όσο καλή και αν είναι μια ομάδα Scrum, πάντοτε υπάρχει δυνατότητα βελτίωσης. Μια καλή ομάδα θα πρέπει συνεχώς να αναζητά ευκαιρίες βελτίωσης, γι' αυτό αφιερώνει ένα σύντομο χρονικό διάστημα στο τέλος κάθε sprint για να βρίσκει τρόπους και μέσα βελτίωσης. Αυτό συμβαίνει κατά τη διάρκεια ανακεφαλαίωσης του sprint, και είναι το τελευταίο πράγμα που γίνεται στη διάρκειά του. Πολλές ομάδες κάνουν κάτι τέτοιο αμέσως μετά τον απολογισμό, όλη η ομάδα συμπεριλαμβανομένου και του Scrum Master και του ιδιοκτήτη του συστήματος θα πρέπει να συμμετάσχουν. Ο συνηθισμένος χρόνος της ανακεφαλαίωσης είναι περίπου μία ώρα, ωστόσο μπορεί να διαρκέσει και πολύ περισσότερο σε περίπτωση που προκύψει κάποιο «καυτό» θέμα στη συζήτηση ή κάποια διαφωνία μεταξύ των συμμετεχόντων.

Αν και υπάρχουν πολλοί τρόποι για να διεξαχθεί μια ανακεφαλαίωση sprint το καλύτερο θα ήταν να γίνει ως μια start – stop – continue συνάντηση, αυτός είναι συχνά ο πιο απλός αλλά και ο πιο αποτελεσματικός τρόπος. Με αυτή την προσέγγιση κάθε μέλος της ομάδας καλείται να προσδιορίσει συγκεκριμένα πράγματα που η ομάδα θα πρέπει να:

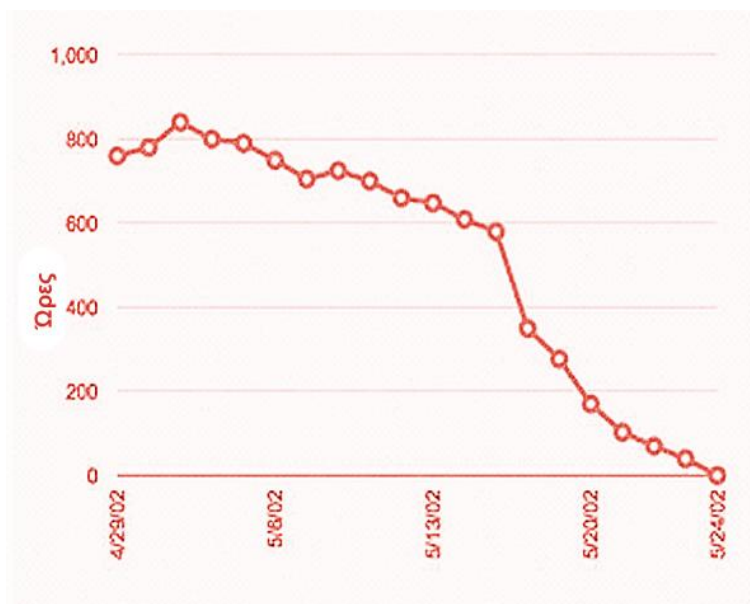
- Αρχίσει να κάνει.
- Σταματήσει να κάνει.
- Συνεχίσει να κάνει.

Υπάρχουν πολλές παραλλαγές σε αυτή την απλή προσέγγιση. Ο Scrum Master μπορεί να διευκολύνει αυτή τη συνάντηση, ζητώντας από όλους να προτείνουν ιδέες και προτρέποντας τον καθένα να καθορίσει ένα πράγμα που πρέπει να ξεκινήσει, να σταματήσει ή να συνεχίσει να γίνεται. Για παράδειγμα ο Scrum Master μπορεί να ζητήσει από όλους να σταματήσουν να εκτελούν μια εργασία γιατί τη συγκεκριμένη χρονική περίοδο δεν έχει δοθεί ιδιαίτερη προσοχή στην εργασία αυτή. Μετά από μια αρχική λίστα ιδεών οι ομάδες θα αποφασίσουν για συγκεκριμένα θέματα στα οποία θα επικεντρωθούν κατά το επόμενο sprint. Στο τέλος του επόμενου sprint, η ανακεφαλαίωση ξεκινάει με τον απολογισμό της λίστας των ενεργειών που επιλέχθηκαν ότι θα δοθεί ιδιαίτερη προσοχή κατά την προηγούμενη ανακεφαλαίωση.

Πολλές φορές υπάρχουν ομάδες που παραλείπουν να κάνουν ανακεφαλαίωση όταν τελειώνουν ένα sprint και προχωράνε στην εκτέλεση του επόμενου. Μπορούμε να πούμε ότι η ανακεφαλαίωση του sprint είναι το δεύτερο σημαντικότερο πράγμα στη μεθοδολογία Scrum (μετά τη συνάντηση σχεδιασμού) μιας και είναι ένας αποτελεσματικός τρόπος για να βελτιώσουμε το σύστημά μας. Χωρίς την ανακεφαλαίωση του sprint έχει παρατηρηθεί ότι οι ομάδες υποπίπτουν συνεχώς στα ίδια λάθη.

6.3 ΔΙΑΓΡΑΜΜΑ SPRINT BURNDOWN

Κατά τη διάρκεια του sprint, τα μέλη της ομάδας αναμένεται να ενημερώσουν το ανεκτέλεστο υπόλοιπο καθώς είναι διαθέσιμες νέες πληροφορίες. Οι περισσότερες ομάδες κάνουν κάτι τέτοιο κατά τη διάρκεια του καθημερινού Scrum. Μία φορά κάθε ημέρα το εκτιμώμενο έργο που απομένει στο sprint υπολογίζεται και απεικονίζεται σε γράφημα από τον Scrum Master και έχει τη μορφή του burndown chart που ακολουθεί. Ο άξονας γ του διαγράμματος αποτυπώνει το φόρτο εργασίας που απομένει να ολοκληρωθεί στην αρχή της κάθε επανάληψης, ενώ ο άξονας x αποτυπώνει τις ημερομηνίες από την έναρξη μέχρι και την ολοκλήρωση του sprint. Η καμπύλη του διαγράμματος θα πρέπει να ελαττώνεται συνεχώς καθώς οι ημέρες προχωρούν. Ωστόσο μικρές αυξομειώσεις στην καμπύλη του διαγράμματος είναι αποδεκτές, καθώς απρόσμενοι παράγοντες μπορεί να προκύψουν κατά τη διάρκεια της επανάληψης.

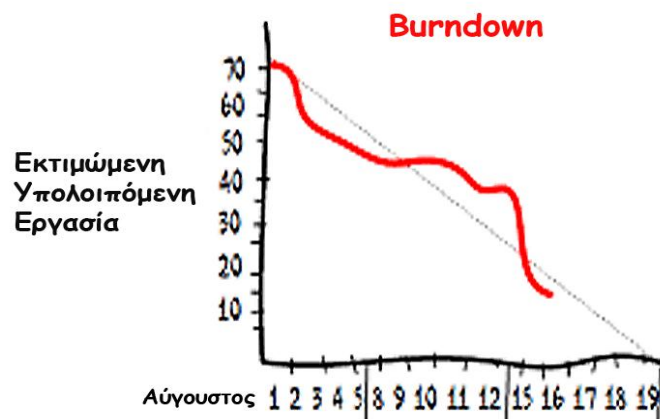


Εικόνα 18: Burndown Chart 1

Η ομάδα κάνει κάθε δυνατή προσπάθεια να εφαρμόσει το σωστό φόρτο εργασίας σε κάθε sprint, αλλά μερικές φορές περισσότερη ή λιγότερη εργασία μπορεί να προκύψει κατά τη διάρκεια του meeting για το σχεδιασμό του Sprint. Σε αυτή την περίπτωση η ομάδα θα πρέπει να προσθέσει ή να αφαιρέσει καθήκοντα. Από το

παραπάνω διάγραμμα βλέπουμε ότι η ομάδα είχε μεγάλο φόρτο εργασίας στις 06/05, στην περίπτωση αυτή ζητήθηκε από τον ιδιοκτήτη του συστήματος να αφαιρέσει κάποια εργασία και αυτό εξηγεί την πτώση της καμπύλης του διαγράμματος για τις ημέρες που ακολουθούν. Από το σημείο αυτό και ύστερα η ομάδα ακολούθησε σταθερή πρόοδο και ολοκλήρωσε επιτυχώς το sprint. Το sprint backlog burndown γράφημα αποτυπώνει το μέγεθος των εργασιών που απομένει να ολοκληρωθούν κατά τη διάρκεια του sprint. Οι εργασίες που απομένουν να ολοκληρωθούν είναι το άθροισμα όλων των ανεκτέλεστων καθηκόντων του sprint που δεν έχουν ακόμη υλοποιηθεί. Το άθροισμα αυτό καταγράφεται κάθε ημέρα από την ομάδα και στη συνέχεια χρησιμοποιείται για τη δημιουργία του γραφήματος, το οποίο αποτυπώνει το υπόλοιπο των εργασιών σε σχέση με το χρόνο για τη διάρκεια του συγκεκριμένου sprint.

Εστιάζοντας στο παρακάτω γράφημα καταλαβαίνουμε ότι την πρώτη ημέρα του sprint, την 1^η Αυγούστου η ομάδα υπολογίζει ότι της απομένουν κατά προσέγγιση 70 story points ή ιδανικά ανθρωποημέρες. Την 16^η Αυγούστου η ομάδα εκτιμά ότι απομένουν άλλα 15 story points τα οποία πρέπει να ολοκληρωθούν μέχρι το τέλος του sprint. Στους υπολογισμούς στον άξονα x δεν περιλαμβάνονται σκόπιμα τα Σαββατοκύριακα μιας και ελάχιστες ή και καθόλου εργασίες ολοκληρώνονται κατά τη διάρκειά τους.

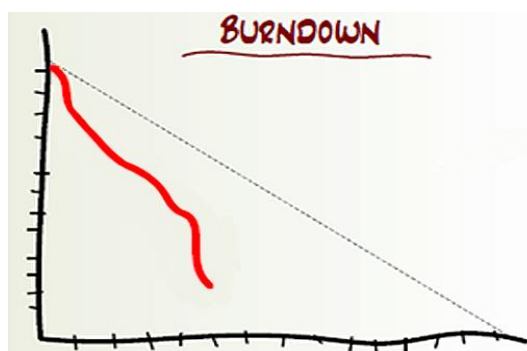


Εικόνα 19: Burndown Chart 2 [Henrik Kniberg, "Scrum and XP from the Trenches", 2006, p 33]

Το διάγραμμα burndown chart μπορεί επίσης να μας καθοδηγήσει για το ποσό του ανεκτέλεστου υπόλοιπου των εργασιών που θα πρέπει να προστεθεί ή να αφαιρεθεί. Πιο συγκεκριμένα αν η καμπύλη του διαγράμματος είναι πολύ πιο πάνω από τη νοητή ευθεία που συνδέει την κορυφή των story points και την ημέρα ολοκλήρωσης του sprint αυτό σημαίνει ότι θα πρέπει να αφαιρέσουμε κάποιες εργασίες από το ανεκτέλεστο υπόλοιπο του sprint. Στην αντίθετη περίπτωση θα πρέπει να προστεθούν επιπλέον εργασίες στο ανεκτέλεστο υπόλοιπο μιας και η ομάδα μπορεί να ολοκληρώσει περισσότερες εργασίες σε σχέση με αυτές που είχε αρχικά υπολογίσει. Αυτό αποτυπώνεται γραφικά και στα ακόλουθα burndown charts.



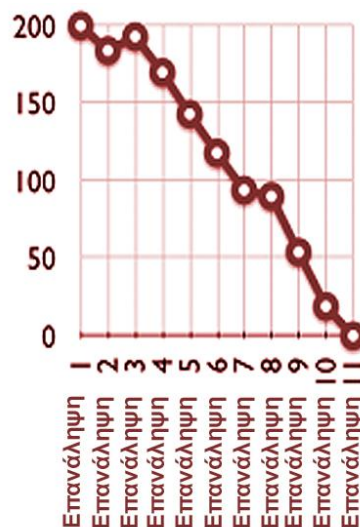
Εικόνα 20: Αφαίρεση μέρους εργασιών στο Burndown Chart 2 [Henrik Kniberg, "Scrum and XP from the Trenches", 2006, p 34]



Εικόνα 21: Προσθήκη μέρους εργασιών στο Burndown Chart [Henrik Kniberg, "Scrum and XP from the Trenches", 2006, p 34]

6.4 ΔΙΑΓΡΑΜΜΑ RELEASE BURNDOWN

Το release burndown γράφημα καταγράφει το σύνολο του υπολειπόμενου ανεκτέλεστου υπόλοιπου του συστήματος σε σχέση με το χρόνο. Οι υπολειπόμενες εργασίες μπορούν να υπολογιστούν σε οποιαδήποτε μονάδα χρόνου επιλέξει η ομάδα Scrum και ο οργανισμός για τον οποίο κατασκευάζεται το σύστημα. Σε ένα Scrum project, η ομάδα παρακολουθεί την πρόοδο που έχει επιτευχθεί, ενημερώνοντας το burndown διάγραμμα στο τέλος κάθε sprint. Στον οριζόντιο άξονα του διαγράμματος απεικονίζονται τα sprints, ενώ στον κάθετο άξονα η ποσότητα της εργασίας που απομένει στην αρχή του κάθε sprint. Η υπολειπόμενη εργασία μπορεί να μετράται σε οποιαδήποτε μονάδα επιλέξουμε (π.χ. ημέρες, ώρες, εβδομάδες) αλλά συνήθως επιλέγεται ως μονάδα μέτρησης τα story points.



Εικόνα 22: Release Burndown Chart

Στο παραπάνω διάγραμμα η ομάδα ξεκίνησε ένα έργο το οποίο σχεδιάστηκε να έχει συνολικά έντεκα sprints διάρκειας δύο εβδομάδων το καθένα. Όπως φαίνεται και από το παραπάνω διάγραμμα το πρώτο sprint πήγε αρκετά καλά αφού στη διάρκειά του παρήχθησαν 20 story points. Από το δεύτερο μέχρι το τρίτο sprint βλέπουμε ότι το εκτιμώμενο υπολειπόμενο έργο αυξήθηκε, αυτό μπορεί να συνέβηκε επειδή προστέθηκε κάποια επιπλέον εργασία ή γιατί η ομάδα άλλαξε κάποιες εκτιμήσεις σχετικά με το υπολειπόμενο έργο. Από το τρίτο sprint και μετά η πρόοδος

συνεχίστηκε επιτυχώς, με μια μικρή επιβράδυνση βέβαια στη διάρκεια του έβδομου sprint αλλά στη συνέχεια η ομάδα ανέκαμψε γρήγορα και έφερε εις πέρας το τμήμα της εφαρμογής για το οποίο είχε αρχικά δεσμευτεί.

Αυτός ο τύπος γραφήματος λειτουργεί καλά σε πολλές περιπτώσεις και για πολλές ομάδες. Ωστόσο σε projects που αλλάζουν οι απαιτήσεις, ίσως θα πρέπει να σκεφτούμε ένα εναλλακτικό τύπο διαγράμματος.

6.5 ΟΛΟΚΛΗΡΩΣΗ ΕΡΓΑΣΙΩΝ

Η μεθοδολογία Scrum αναμένει από τις ομάδες να εξελίσσουν τη λειτουργικότητα του συστήματος μετά το τέλος του κάθε sprint και συγκεκριμένα η νέα λειτουργικότητα θα πρέπει να μπορεί να ενσωματωθεί στην ήδη υπάρχουσα και να λειτουργήσει το σύστημα με επιτυχία. Για να γίνει κάτι τέτοιο η νέα λειτουργικότητα πρέπει να είναι ένα ολοκληρωμένο (done) τμήμα του συνολικού προϊόντος. Κάθε νέο τμήμα είναι πρόσθετο του προηγούμενου και δοκιμασμένο σχολαστικά εξασφαλίζοντας ότι τα επιμέρους τμήματα δουλεύουν σωστά.

Όταν λέμε ότι ένα τμήμα της εφαρμογής έχει ολοκληρωθεί εννοούμε ότι όλες οι εργασίες που σχετίζονται με την ανάλυση, το σχεδιασμό, τον προγραμματισμό, την αποσφαλμάτωση, την τεκμηρίωση και τη δοκιμή έχουν τελειώσει. Κάποιες ομάδες δεν μπορούν να συμπεριλάβουν όλες τις παραπάνω διαδικασίες και έτσι ο προσδιορισμός done δεν είναι σωστά καθορισμένος, κάτι που θα πρέπει να γίνεται σαφές στον ιδιοκτήτη του προϊόντος. Οι μη ολοκληρωμένες (undone) εργασίες συχνά συσσωρεύονται στο ανεκτέλεστο υπόλοιπο του προϊόντος και έτσι το product burndown backlog γράφημα παραμένει πιο ακριβές και πιο ενημερωμένο.

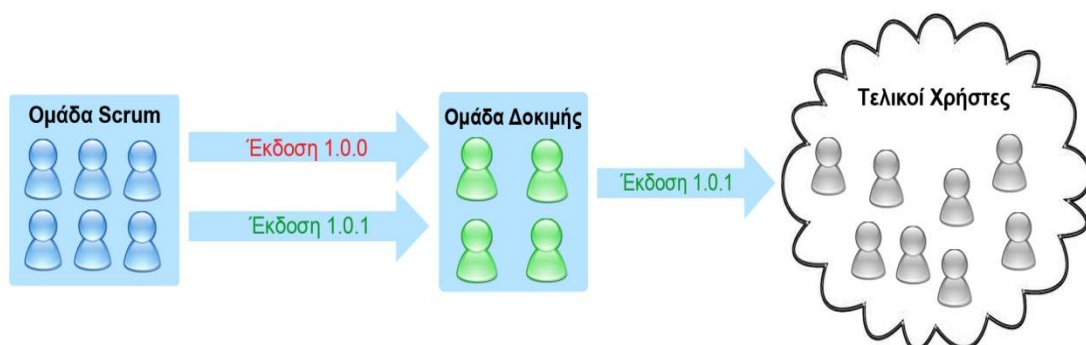
Μερικοί οργανισμοί δεν μπορούν να δημιουργήσουν ένα ολοκληρωμένο τμήμα της εφαρμογής κατά τη διάρκεια ενός sprint. Ίσως γιατί δεν έχουν ακόμη την ανάλογη υποδομή για να ολοκληρώσουν ένα σύνολο δοκιμών. Σε αυτή την περίπτωση έχουμε δύο κατηγορίες, τις ολοκληρωμένες εργασίες και τις μη ολοκληρωμένες

εργασίες. Οι μη ολοκληρωμένες εργασίες είναι ένα σύνολο εργασιών που πρέπει να ολοκληρωθεί σε μετέπειτα χρονική στιγμή. Ο ιδιοκτήτης του προϊόντος γνωρίζει ακριβώς τι πρέπει να επιθεωρήσει στο τέλος κάθε sprint από τον προσδιορισμό των ολοκληρωμένων εργασιών. Οι μη ολοκληρωμένες εργασίες προστίθενται στο ανεκτέλεστο υπόλοιπο προϊόντος και αναπαριστώνται γραφικά στο release burndown chart. Αυτή η τεχνική προσθέτει διαφάνεια και σαφήνεια στη διαδικασία της προόδου του συστήματος.

Κατά τη διάρκεια των sprints οι μη ολοκληρωμένες εργασίες συσσωρεύονται και πρέπει να έχουν ολοκληρωθεί πριν από την τελική κυκλοφορία του προϊόντος. Ο φόρτος των μη ολοκληρωμένων εργασιών αποτυπώνεται στο γράφημα γραμμικά αν και κάποιες φορές μπορεί να παρουσιάζει εκθετική μορφή. Ο αριθμός των sprints μπορεί να είναι απρόβλεπτος στην περίπτωση που οι μη ολοκληρωμένες εργασίες έχουν στο γράφημα εκθετική μορφή.

Το δοκιμαστικό στάδιο είναι το δυσκολότερο κομμάτι, όχι μόνο όσο αφορά τη μεθοδολογία Scrum αλλά γενικότερα την ανάπτυξη λογισμικού. Εξαρτάται από διάφορους παράγοντες όπως από τον αριθμό των ατόμων που συμμετέχουν στη δοκιμή, τον τύπο του συστήματος που αναπτύσσεται, το μέγεθος του κύκλου ζωής του πληροφοριακού συστήματος και το βαθμό της κρισιμότητάς του.

Όπως έχουμε αναφέρει στο τέλος κάθε sprint αναμένουμε ένα πλήρως λειτουργικό τμήμα της συνολικής εφαρμογής. Όμως η μέχρι τώρα πορεία έχει δείξει ότι οι πρώτες εκδόσεις λογισμικού εμφανίζουν bugs τα οποία είναι αρκετά ενοχλητικά για τους τελικούς χρήστες. Για το λόγο αυτό απαιτείται να υπάρχει μια δοκιμαστική φάση αποδοχής κατά την οποία μια ανεξάρτητη ομάδα δοκιμής θα κάνει το testing της εφαρμογής, θα ενεργεί ως οι τελικοί χρήστες του συστήματος και θα εντοπίζει σφάλματα τα οποία η Scrum ομάδα δεν αντιλήφθηκε κατά τη διάρκεια της υλοποίησης. Στη συνέχεια η ομάδα Scrum θα διορθώνει τα σφάλματα που εντοπίστηκαν κυκλοφορώντας μια καινούργια έκδοση λογισμικού. Η δοκιμαστική φάση αποδοχής αναφέρεται στη συνολική χρονική περίοδο της δοκιμής, της αποσφαλμάτωσης και της κυκλοφορίας της νέας έκδοσης λογισμικού. Η διαδικασία που αναλύθηκε μέχρι στιγμής αποτυπώνεται γραφικά στην παρακάτω εικόνα. [Henrik Kniberg, Scrum and XP from the Trenches (2006), p 61]



Εικόνα 23: Δοκιμαστική φάση αποδοχής [Henrik Kniberg, "Scrum and XP from the Trenches", 2006, p 61]

Όπως καταλαβαίνουμε, η διαδικασία εντοπισμού σφαλμάτων και η διόρθωσή της είναι αρκετά χρονοβόρα και επίπονη ενώ δεν παρέχει ιδιαίτερη ευελιξία στη συνολική διαδικασία. Είναι όμως απαραίτητη και για το λόγο αυτό δεν μπορούμε να την παραβλέψουμε, μπορούμε όμως να ελαχιστοποιήσουμε τη χρονική διάρκεια που απαιτείται για την ολοκλήρωσή της. Αυτό γίνεται βελτιστοποιώντας την ποιότητα του κώδικα που υλοποιείται από την ομάδα ενσωματώνοντας στα μέλη της άτομα που έχουν σημαντική εμπειρία από τη φάση της δοκιμής και ολοκληρώνοντας όλες τις εργασίες που έχουμε δεσμευτεί στο έπακρο κατά τη διάρκεια του sprint.



Εικόνα 24: Προσθήκη Tester στην ομάδα Scrum [Henrik Kniberg, “Scrum and XP from the Trenches”, 2006, p 62]

Όπως φαίνεται στην εικόνα 24, ο tester είναι ένα μέλος της ομάδας όπου έχει ιδιαίτερες ικανότητες στη δοκιμή της εφαρμογής και όχι ένα άτομο του οποίου πρωταρχικός ρόλος είναι μόνο το testing του συστήματος. Οι προγραμματιστές δεν είναι πάντοτε η καλύτερη επιλογή για τη δοκιμή της εφαρμογής ειδικά όταν πρέπει να διορθώσουν το δικό τους κώδικα και πολλές φορές θεωρούνε κάτι τελειωμένο όταν αυτό δεν έχει πλήρως ολοκληρωθεί. Ο tester είναι αυτός που θα έχει τον τελικό λόγο για το πότε τα πάντα έχουν ολοκληρωθεί μέσα στο sprint και μπορούμε να συνεχίσουμε παρακάτω. Για να γίνει αυτό ο tester είναι υποχρεωμένος να υποβάλλει σε δοκιμή το τμήμα της εφαρμογής το οποίο θα πρέπει να μπορεί να λειτουργήσει χωρίς πρόβλημα στον server που έχει επιλέξει.

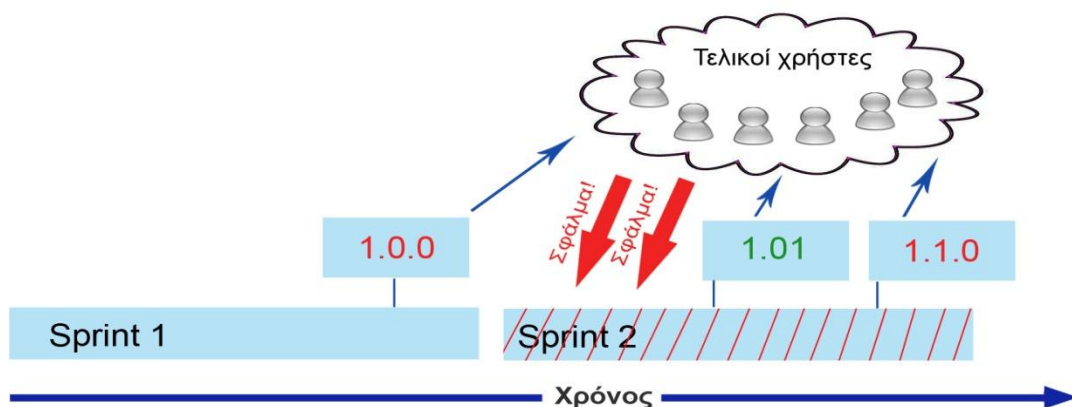
Ίσως χρειαστεί μια εβδομάδα μέχρι η ομάδα να ολοκληρώσει την πρώτη εργασία από το πρώτο ανεκτέλεστο υπόλοιπο του sprint. Επειδή τα χρονικά περιθώρια σε

ένα sprint είναι συγκεκριμένα, ο tester θα πρέπει με κάτι να ασχοληθεί το χρονικό διάστημα πριν υποβάλλει την εφαρμογή σε δοκιμή. Έτσι ο Scrum Master μπορεί να αποφασίσει ότι πρέπει να είναι έτοιμες οι προδιαγραφές και το περιβάλλον της δοκιμής της εφαρμογής πριν την ολοκλήρωση της πρώτης εργασίας από το ανεκτέλεστο υπόλοιπο, οπότε είναι δουλειά του tester να κάνει κάτι τέτοιο έτσι ώστε όταν οι προγραμματιστές έχουν ένα τμήμα της εφαρμογής έτοιμο να μη χρειαστεί να περιμένουμε. Αν ο tester της εφαρμογής έχει ικανότητες προγραμματισμού, τότε μπορεί να βοηθήσει την ομάδα ανάπτυξης, αν όχι, κάνει οτιδήποτε μπορεί ώστε να επιτευχθεί ο στόχος του sprint και αναλαμβάνει όλες τις μη προγραμματιστικές διαδικασίες οι οποίες μπορεί να είναι οι ακόλουθες. [Henrik Kniberg, "Scrum and XP from the Trenches", 2006, p 63]

- Ετοιμάζει το περιβάλλον δοκιμής.
- Διευκρινίζει τις απαιτήσεις.
- Επικοινωνεί με εξωτερικούς παράγοντες (π.χ. GUI designers).
- Βελτιώνει τα σενάρια υλοποίησης.
- Χωρίζει το ανεκτέλεστο υπόλοιπο των εργασιών σε επιμέρους υπό-εργασίες.
- Αναγνωρίζει και απαντά σε τυχόν ερωτήσεις ή απορίες που μπορεί να προκύψουν κατά τη διάρκεια του sprint.

Η φάση της δοκιμής της εφαρμογής δεν πρέπει να αποτελεί μέρος του sprint, το sprint βασίζεται σε χρονοκύβους. Είναι ιδιαίτερα δύσκολο να προσδιορίσουμε χρονικά τη διαδικασία της αποσφαλμάτωσης και της κυκλοφορίας μιας νέας έκδοσης, ενέργειες που εμπεριέχονται στη δοκιμή της εφαρμογής. Όταν ο χρόνος του sprint εξαντλείται και υπάρχει κάποιο σφάλμα, τότε δεν μπορούμε να κυκλοφορήσουμε μια έκδοση λογισμικού που περιέχει κάποιο bug αλλά ούτε έχουμε χρόνο να τη διορθώσουμε. Επίσης όταν έχουμε πολλές Scrum ομάδες που δουλεύουν σε ένα project, τότε η δοκιμή της εφαρμογής γίνεται συνδυάζοντας τα αποτελέσματα της εργασίας των ομάδων. Για τους παραπάνω λόγους καταλαβαίνουμε ότι η δοκιμή της εφαρμογής θα πρέπει να αποτελεί ξεχωριστή φάση του κύκλου ανάπτυξης του πληροφοριακού συστήματος.

Ιδανικά θα μπορούσαμε να αναμένουμε μια πλήρως λειτουργική έκδοση λογισμικού στο τέλος κάθε sprint, όμως γνωρίζουμε πως αυτό δεν είναι εφικτό. Μετά από ένα sprint παράγεται μια έκδοση λογισμικού που περιέχει κάποια σφάλματα. Καθώς προχωράμε στην υλοποίηση των επόμενων sprints, νέα σφάλματα προστίθενται στα ήδη υπάρχοντα και κάνουν την εφαρμογή μας περισσότερο προβληματική όπως απεικονίζεται και στην ακόλουθη εικόνα.



Εικόνα 25: Συσσωρευμένα σφάλματα κατά την υλοποίηση των sprints [Henrik Kniberg, "Scrum and XP from the Trenches", 2006, p 65]

Ακόμη και όταν στην ομάδα υπάρχει κάποιος tester, σφάλματα και πάλι θα προκύψουν με τη μόνη διαφορά ότι τα παράπονα δεν θα προέρχονται από τους εξαγριωμένους τελικούς χρήστες αλλά από την ομάδα που εκτελεί τη δοκιμή. Από την πλευρά των προγραμματιστών οι ενέργειες που πρέπει να γίνουν είναι οι ίδιες, από την επιχειρηματική σκοπιά όμως υπάρχει τεράστια διαφορά μιας και οι testers είναι πάντοτε πιο φιλικόι από τους τελικούς χρήστες και σαφώς δεν μπορούν να προκαλέσουν δυσφήμιση για την υπόληψη του οργανισμού. Από όποια πλευρά και να το δούμε, τα σφάλματα θα πρέπει να διορθώνονται και να μη διαιωνίζονται κατά τη διαδοχική εκτέλεση των sprints.

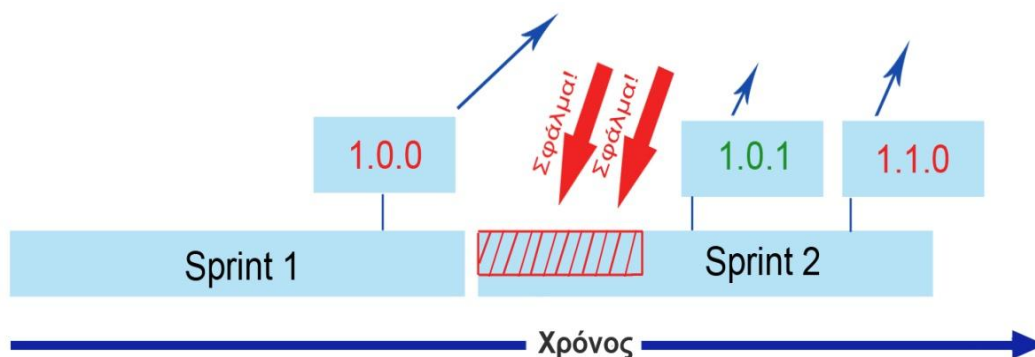
Για το λόγο αυτό εφαρμόζονται δύο πρακτικές, στην πρώτη δεν ξεκινάει η εκτέλεση του επόμενου sprint αν δεν υπάρξει μια πλήρως λειτουργική έκδοση του προηγούμενου, όπως φαίνεται και στην παρακάτω εικόνα. Αυτό όμως προϋποθέτει ότι θα πρέπει να προσαρμόσουμε μια μη οριοθετημένη σε χρονοκύβους περίοδο κυκλοφορίας μεταξύ των διαδοχικών sprints στην οποία περιέχονται οι διαδικασίες

της αποσφαλμάτωσης και της δημιουργίας νέας έκδοσης χωρίς σφάλματα. Η διαδικασία αυτή δεν είναι πάντοτε τόσο βολική μιας και διακόπτει την κανονική ροή των sprints και δεν μπορούμε να υπολογίσουμε το χρονικό διάστημα κατά το οποίο θα ξεκινήσουμε το επόμενο sprint.



Εικόνα 26: Release εκδόσεις μεταξύ των sprints [Henrik Kniberg, Scrum and XP from the Trenches (2006), p 66]

Η δεύτερη προσέγγιση είναι πιο συνηθισμένη και προτιμάται περισσότερο στη Scrum μεθοδολογία γιατί προσφέρει περισσότερη ευελιξία και πιο ακριβείς υπολογισμούς για τη χρονική διάρκεια εκτέλεσης των επόμενων sprints. Με βάση την πρακτική αυτή, όταν τελειώνουμε ένα sprint μπορούμε να προχωράμε στο επόμενο. Θα πρέπει όμως να αναμένουμε ότι θα αφιερώσουμε κάποιο χρονικό διάστημα στο επόμενο sprint για να διορθώσουμε σφάλματα από το προηγούμενο, όπως φαίνεται στην εικόνα 27. Το μόνο που χρειάζεται είναι να εκτιμήσουμε σωστά το χρόνο που απαιτείται για τη διόρθωση σφαλμάτων των προηγούμενων sprints, όπου καθώς προχωράμε στην υλοποίηση των επόμενων sprints οι εκτιμήσεις μας για το χρόνο που απαιτείται για τις διορθώσεις των σφαλμάτων γίνονται πιο ακριβείς. [Henrik Kniberg, “Scrum and XP from the Trenches”, 2006, p 67]



Εικόνα 27: Διορθώσεις σφαλμάτων προηγούμενων sprints [Henrik Kniberg, “Scrum and XP from the Trenches”, 2006, p 67]

Οι παραδοσιακές μεθοδολογίες σχεδιάστηκαν για να ανταποκρίνονται στη μη προβλεψιμότητα του εξωτερικού περιβάλλοντος στο ξεκίνημα κάθε κύκλου εξέλιξης. Προσεγγίσεις όπως η ελικοειδής μεθοδολογία και οι παρόμοιές της περιορίζουν την ικανότητά τους να ανταποκρίνονται στις αλλαγές των απαιτήσεων όταν το πληροφοριακό σύστημα έχει ξεκινήσει να υλοποιείται.

Αντιθέτως, η μεθοδολογία Scrum έχει σχεδιαστεί έτσι ώστε να παρέχει ευελιξία σε όλη τη διάρκεια του κύκλου ανάπτυξης ενός πληροφοριακού συστήματος. Παρέχει μηχανισμούς ελέγχου για το σχεδιασμό μιας νέας έκδοσης του προϊόντος και στη συνέχεια διαχειρίζεται τις μεταβλητές καθώς η εφαρμογή εξελίσσεται. Αυτό το γεγονός επιτρέπει στους οργανισμούς να κάνουν αλλαγές στο project και στα παραδοτέα του σε οποιαδήποτε φάση ανάπτυξης, οποιαδήποτε χρονική στιγμή. Η Scrum προσέγγιση επιτρέπει στους developers να υλοποιήσουν τις πιο έξυπνες λύσεις για το πληροφοριακό σύστημα καθώς η εφαρμογή αναπτύσσεται.

Η αντικειμενοστρεφής τεχνολογία είναι η βάση για τη Scrum μεθοδολογία, ο διαδικασιακός κώδικας και οι σχετικές με αυτόν διεπαφές δεν εφαρμόζονται σε αυτή την περίπτωση. Η Scrum μπορεί επιλεκτικά να εφαρμοστεί σε διαδικασιακά συστήματα με συγκεκριμένες διεπαφές και ισχυρό προσανατολισμό δεδομένων. Ο πίνακας που ακολουθεί παρουσιάζει τη σύγκριση μεταξύ της Scrum προσέγγισης και των βασικότερων μη ευέλικτων μεθοδολογιών.

	Καταρράκτης	Ελικοειδής	Επαναληπτική	Scrum
Καθορισμός διαδικασιών	Απαιτείται	Απαιτείται	Απαιτείται	Στις φάσεις σχεδιασμού & ολοκλήρωσης
Τελικό προϊόν	Καθορίζεται στη φάση	Καθορίζεται στη φάση	Καθορίζεται κατά τη	Καθορίζεται κατά τη

	σχεδιασμού	σχεδιασμού	διάρκεια του project	διάρκεια του project
Κόστος έργου	Καθορίζεται στη φάση σχεδιασμού	Μεταβλητό εν μέρη	Καθορίζεται κατά τη διάρκεια του project	Καθορίζεται κατά τη διάρκεια του project
Ημερομηνία ολοκλήρωσης	Καθορίζεται στη φάση σχεδιασμού	Μεταβλητή εν μέρη	Καθορίζεται κατά τη διάρκεια του project	Καθορίζεται κατά τη διάρκεια του project
Ανταπόκριση στο περιβάλλον	Κατά τη φάση σχεδιασμού	Κατά τη φάση σχεδιασμού	Στο τέλος κάθε επανάληψης	Σε όλη τη διάρκεια του project
Ευελιξία	Περιορισμένη	Περιορισμένη	Περιορισμένη	Απεριόριστη
Πιθανότητα επιτυχίας	Χαμηλή	Μέτρια χαμηλή	Μέτρια	Υψηλή

Πίνακας 12: Σύγκριση Scrum και μη ευέλικτων μεθοδολογιών [Henrik Kniberg, “Scrum and XP from the Trenches”, 2006, p 34]

Από τον παραπάνω πίνακα γίνεται εύκολα κατανοητό ότι η Scrum μεθοδολογία παρέχει τα εξής πλεονεκτήματα έναντι των παραδοσιακών μεθόδων ανάπτυξης λογισμικού:

- **Ευκολία αλλαγών:** Η μεθοδολογία Scrum έχει σχεδιαστεί έτσι ώστε να παρέχει ευελιξία σε όλη τη διάρκεια του κύκλου ανάπτυξης ενός πληροφοριακού συστήματος. Παρέχει πρακτικές για το σχεδιασμό μιας νέας έκδοσης του προϊόντος και στη συνέχεια διαχειρίζεται τις πρακτικές αυτές καθώς η εφαρμογή εξελίσσεται. Αντίθετα στις παραδοσιακές προσεγγίσεις ακολουθείται το πλάνο που έχει ορισθεί εξ' αρχής και οι αλλαγές που μπορεί να προκύψουν, ίσως αποδειχθούν καταστροφικές.
- **Ανθρωποκεντρικός προσανατολισμός:** Η συνεργασία των Scrum ομάδων έχει ως αποτέλεσμα την εύκολη ολοκλήρωση των εργασιών μιας και όλοι

είναι ενήμεροι για το τι υλοποιείται στην ομάδα. Αντίθετα οι παραδοσιακές προσεγγίσεις υποστηρίζουν προσανατολισμό στις διαδικασίες και έτσι είναι δύσκολη και πολύπλοκη η διασύνδεση όλων των εμπλεκομένων μερών γιατί καθένα τους δουλεύει ανεξάρτητα.

- **Τεκμηρίωση επαναλήψεων:** Κύρια εργασία της μεθοδολογίας Scrum είναι οι επαναλήψεις. Το αποτέλεσμα των επαναλήψεων είναι το τμήμα του λογισμικού που παράγεται κάθε φορά. Έτσι μπορούμε να πούμε ότι η τεκμηρίωση των επαναλήψεων είναι ο κώδικας που παράγεται σε κάθε τμήμα του λογισμικού. Στις παραδοσιακές προσεγγίσεις πρέπει πρώτα να υπάρχει το έγγραφο των απαιτήσεων (τεκμηρίωση) και στη συνέχεια παράγεται ο κώδικας που ολοκληρώνει την εφαρμογή.
- **Γρήγορη ανατροφοδότηση:** Με την ολοκλήρωση των sprints ο πελάτης μπορεί να βλέπει συνεχώς καινούργιες εκδόσεις λογισμικού σε μικρά χρονικά διαστήματα (2 – 3 εβδομάδες). Με τον τρόπο αυτό μπορεί να έχει πλήρη επίγνωση σχετικά με την εξέλιξη του έργου και σαφώς καλύτερη και γρηγορότερη ανατροφοδότηση.
- **Ποιοτικότερα συστήματα:** Οι επαναλήψεις οδηγούν στην παραγωγή ποιοτικότερων συστημάτων, μιας και πριν ξεκινήσει να υλοποιείται το επόμενο sprint, έχουμε ήδη εξασφαλίσει ένα πλήρως λειτουργικό τμήμα του συνολικού συστήματος.
- **Ικανοποίηση πελατών:** Όπως αναφέραμε μέχρι τώρα, ο πελάτης έχει ενεργή συμμετοχή στην όλη διαδικασία, μιας και είναι εκείνος ο οποίος ορίζει τη λίστα προτεραιοτήτων στο ανεκτέλεστο υπόλοιπο του προϊόντος. Επίσης ενημερώνεται συνεχώς για τη διαδικασία ολοκλήρωσης του συστήματος μέσα από τα τμήματα λογισμικού που παρουσιάζονται μετά από κάθε επανάληψη. Επίσης όταν η Scrum εφαρμόζεται σωστά και ακολουθούνται οι πρακτικές της το τελικό προϊόν παραδίδεται on time. Επομένως καταλαβαίνουμε ότι η ικανοποίηση των πελατών είναι μεγαλύτερη στη μεθοδολογία Scrum σε σχέση με τις υπόλοιπες παραδοσιακές προσεγγίσεις.

Στο σημείο αυτό θα επικεντρωθούμε στη σύγκριση δύο ευέλικτων μεθοδολογιών και πιο συγκεκριμένα της Scrum και του eXtreme Programming χρησιμοποιώντας το εργαλείο 4 – DAT (Dimensional Analytical Tool). Το 4 – DAT αναπτύχθηκε από τους ερευνητές (Asif Qumer, Brian Henderson – Sellers) με σκοπό να παρέχει ανάλυση για τη σύγκριση των ευέλικτων μεθοδολογιών. Πιο συγκεκριμένα το τετραδιάστατο αναλυτικό εργαλείο χρησιμοποιείται στην ανάλυση της ανάπτυξης λογισμικού και παρέχει ένα μηχανισμό, ο οποίος μετράει το βαθμό της ευελιξίας οποιασδήποτε ποιοτικής ή ποσοτικής μεθόδου σε ένα συγκεκριμένο σημείο μιας διαδικασίας, χρησιμοποιώντας συγκεκριμένες πρακτικές.

Το εργαλείο 4 – DAT έχει τέσσερις διαφορετικές διαστάσεις οι οποίες παρέχουν κριτήρια αξιολόγησης για τη λεπτομερή εκτίμηση των ευέλικτων μεθοδολογιών της ανάπτυξης λογισμικού από πολλές διαφορετικές σκοπιές. Αυτές οι διαστάσεις αναπτύχθηκαν αξιοποιώντας προηγούμενες σκέψεις και αντιλήψεις που σχετίζονταν με την ευελιξία των μεθοδολογιών ανάπτυξης, τις παραδοσιακές μεθοδολογίες ανάπτυξης λογισμικού καθώς επίσης και τις ευέλικτες αρχές και αξίες. Οι ακόλουθες υποενότητες περιγράφουν με λεπτομέρεια τις τέσσερις διαστάσεις του εργαλείου 4 – DAT. [Asif Qumer, Brian Henderson – Sellers “Comparative Evaluation of XP and Scrum Using The 4D Analytical Tool (4 – DAT)”, 2006, p 1-2]

9.1 ΠΡΩΤΗ ΔΙΑΣΤΑΣΗ: ΧΑΡΑΚΤΗΡΙΣΜΟΣ ΤΟΥ ΠΕΔΙΟΥ ΔΡΑΣΗΣ ΤΗΣ ΜΕΘΟΔΟΥ

Ο χαρακτηρισμός του πεδίου δράσης της μεθόδου είναι ένα σύνολο χαρακτηριστικών που χρησιμοποιούνται στις συγκεκριμένες ευέλικτες μεθοδολογίες και μας βοηθάνε στη σύγκριση των διαφορετικών μεθοδολογιών σε υψηλό επίπεδο. Ο παρακάτω πίνακας περιγράφει την πρώτη διάσταση του εργαλείου 4 – DAT. [Asif Qumer, Brian Henderson – Sellers “Comparative Evaluation of XP and Scrum Using The 4D Analytical Tool (4 – DAT)”, 2006, p 2]

Πεδίο δράσης	Περιγραφή
Μέγεθος Έργου	Παρέχει η μέθοδος υποστήριξη για μικρά, μεσαία ή μεγάλα έργα;
Μέγεθος Ομάδας	Υποστήριξη μικρών ή μεγάλων ομάδων (απλές ή πολλαπλές ομάδες);
Στυλ ανάπτυξης	Επαναληπτικό (iterative) ή ταχύ (rapid) στυλ;
Στυλ κώδικα	Καθορίζεται στη μέθοδο το στυλ του κώδικα (απλό ή σύνθετο);
Τεχνολογικό περιβάλλον	Ποια εργαλεία ή μεταγλωττιστές καθορίζει η μεθοδολογία;
Φυσικό περιβάλλον	Υποστηρίζεται κατανεμημένο ή κεντροποιημένο περιβάλλον;
Επιχειρησιακή κουλτούρα	Τι είδος επιχειρησιακή κουλτούρα υποστηρίζει η μέθοδος (συνεργασιακή ή μη συνεργασιακή);
Αφαιρετικός μηχανισμός	Καθορίζει η μέθοδος κάποιον αφαιρετικό μηχανισμό (object oriented, agent oriented)

Πίνακας 13: Η πρώτη διάσταση του εργαλείου 4 – DAT

9.2 ΔΕΥΤΕΡΗ ΔΙΑΣΤΑΣΗ: ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΗΣ ΕΥΕΛΙΞΙΑΣ

Τα χαρακτηριστικά της ευελιξίας είναι η δεύτερη διάσταση του εργαλείου 4 – DAT. Η δεύτερη διάσταση ελέγχει την παρουσία της ευελιξίας στις μεθοδολογίες τόσο σε επίπεδο διαδικασιών όσο και σε επίπεδο πρακτικών. Είναι η μοναδική από τις τέσσερις διαστάσεις η οποία είναι ποσοτική. [Asif Qumer, Brian Henderson – Sellers “Comparative Evaluation of XP and Scrum Using The 4D Analytical Tool (4 – DAT)”, 2006, p 2]

Χαρακτηριστικό	Περιγραφή
Ευελιξία	Διευκολύνει η μέθοδος προβλεπόμενες ή μη προβλεπόμενες αλλαγές;
Ταχύτητα	Παρέχει η μέθοδος γρήγορα αποτελέσματα;

Ισχύνητα	Χρησιμοποιούνται οικονομικά, απλά και ποιοτικά μέσα για την παραγωγή του συστήματος;
Μάθηση	Ενθαρρύνει η μέθοδος νέα γνώση και νέα μάθηση;
Ανταπόκριση	Παρουσιάζει ανταπόκριση η μέθοδος;

Πίνακας 14: Η δεύτερη διάσταση του εργαλείου 4 – DAT

9.3 ΤΡΙΤΗ ΔΙΑΣΤΑΣΗ: ΧΑΡΑΚΤΗΡΙΣΜΟΣ ΤΩΝ ΕΥΕΛΙΚΤΩΝ ΤΙΜΩΝ

Ο χαρακτηρισμός των ευέλικτων τιμών αποτελείται από έξι στοιχεία. Η τρίτη διάσταση εξετάζει την υποστήριξη που παρέχεται στις διάφορες μεθοδολογίες όπως φαίνεται και στον παρακάτω πίνακα. [Asif Qumer, Brian Henderson – Sellers “Comparative Evaluation of XP and Scrum Using The 4D Analytical Tool (4 – DAT)”, 2006, p 3]

Ευέλικτες τιμές	Περιγραφή
Αλληλεπιδράσεις πάνω σε διαδικασίες και εργαλεία	Ποιες πρακτικές ευνοούν την αλληλεπίδραση διαδικασιών και εργαλείων;
Λειτουργικό λογισμικό και κατανοητή τεκμηρίωση	Ποιες πρακτικές ευνοούν την κατανοητή τεκμηρίωση και το καλό λογισμικό;
Συνεργασία πελατών	Ποιες πρακτικές ευνοούν τη συνεργασία των πελατών;
Ανταπόκριση σε αλλαγές	Ποιες πρακτικές ευνοούν τις αλλαγές με βάση τον αρχικό μας σχεδιασμό;
Διατήρηση της ευελιξίας	Ποιες πρακτικές ευνοούν τη διατήρηση της συνολικής ευελιξίας;
Διατήρηση του κόστους της διαδικασίας	Ποιες πρακτικές ευνοούν τη διατήρηση του κόστους;

Πίνακας 15: Η τρίτη διάσταση του εργαλείου 4 – DAT

9.4 ΤΕΤΑΡΤΗ ΔΙΑΣΤΑΣΗ: ΧΑΡΑΚΤΗΡΙΣΜΟΣ ΤΗΣ ΔΙΑΔΙΚΑΣΙΑΣ ΛΟΓΙΣΜΙΚΟΥ

Η τέταρτη διάσταση αποτελείται από δύο βασικά συστατικά της διαδικασίας του λογισμικού, τη διαδικασία της μηχανικής του λογισμικού (αποτελείται επίσης από τη φάση της ανάπτυξης, της διαχείρισης του έργου και της υποστήριξης) και τη διαδικασία της διαχείρισης. Η τέταρτη διάσταση του εργαλείου 4 – DAT εξετάζει τις πρακτικές που υποστηρίζουν τις παραπάνω διαδικασίες και αναλύονται στον παρακάτω πίνακα. [Asif Qumer, Brian Henderson – Sellers “Comparative Evaluation of XP and Scrum Using The 4D Analytical Tool (4 – DAT)”, 2006, p 3]

Διαδικασία	Περιγραφή
Ανάπτυξης	Ποιες πρακτικές καλύπτουν τις κύριες διαδικασίες του κύκλου ζωής και της δοκιμής της εφαρμογής;
Διαχ/σης έργου	Ποιες πρακτικές καλύπτουν τη συνολική διαχείριση του έργου;
Τροποποίησης / Υποστήριξης	Ποιες πρακτικές διευκολύνουν την τροποποίηση;
Διαχ/σης διαδικασίας	Ποιες πρακτικές καλύπτουν τη διαχείριση μιας συγκεκριμένης διαδικασίας του έργου;

Πίνακας 16: Η τέταρτη διάσταση του εργαλείου 4 – DAT

9.5 ΣΥΓΚΡΙΣΗ XP & SCRUM ΜΕ ΒΑΣΗ ΤΙΣ ΔΙΑΣΤΑΣΕΙΣ ΤΟΥ ΕΡΓΑΛΕΙΟΥ 4 - DAT

Το πεδίο δράσης του eXtreme Programming και της Scrum αναλύεται ποσοτικά χρησιμοποιώντας την πρώτη διάσταση του εργαλείου 4 – DAT. Ο παρακάτω πίνακας παρουσιάζει το πεδίο δράσης και τη χρησιμότητα των δύο μεθοδολογιών. Με βάση τον πίνακα φαίνεται ότι η XP είναι κατάλληλη για μικρού και μεσαίου μεγέθους projects, ενώ η Scrum μπορεί να λειτουργήσει αποδοτικά σε έργα οποιουδήποτε βεληνεκούς. Επίσης και οι δύο μεθοδολογίες χρησιμοποιούν επαναληπτική και αντικειμενοστρεφή προσέγγιση προκειμένου να αναπτύξουν συστήματα τα οποία θα υλοποιήσουν ομάδες μικρότερες των δέκα ατόμων. Ακόμη η XP αναλύει ζητήματα όπως το στυλ του κώδικα (απλός και καλογραμμένος), τεχνολογίας του

περιβάλλοντος (απαιτεί άμεση ανατροφοδότηση), επιχειρησιακής κουλτούρας (συνεργατικής ή μη), ενώ η Scrum δεν καταπιάνεται με λεπτομέρεια στα παραπάνω ζητήματα. [Asif Qumer, Brian Henderson – Sellers “Comparative Evaluation of XP and Scrum Using The 4D Analytical Tool (4 – DAT)”, 2006, p 4]

Κριτήρια	XP	Scrum
Πεδίο Δράσης		
Μέγεθος Έργου	Μικρό, Μεσαίο	Μικρό, Μεσαίο, Μεγάλο
Μέγεθος Ομάδας	<10	<10 και πολλαπλές ομάδες
Στυλ ανάπτυξης	Επαναληπτικό, γρήγορο	Επαναληπτικό, γρήγορο
Στυλ κώδικα	Καθαρός και απλός	Δεν προσδιορίζεται
Τεχνολογικό περιβάλλον	Απαιτείται άμεση ανατροφοδότηση	Δεν προσδιορίζεται
Φυσικό περιβάλλον	Συστεγαζόμενες και κατανεμημένες ομάδες (περιορισμένη αλληλεπίδραση)	Δεν προσδιορίζεται
Επιχειρησιακή κουλτούρα	Συνεργατική	Δεν προσδιορίζεται
Αφαιρετικός μηχανισμός	Object - oriented	Object - oriented

Πίνακας 17: Αξιολόγηση του πεδίου εφαρμογής του XP και της Scrum

Σε αυτό το σημείο η σύγκριση επικεντρώνεται στο βαθμό της ευελιξίας των δύο μεθοδολογιών ο οποίος προκύπτει λαμβάνοντας υπόψη τα χαρακτηριστικά του πίνακα της δεύτερης διάστασης του εργαλείου 4 – DAT, δηλαδή της ευελιξίας (FY), της ταχύτητας (SD), της ισχύτητας (LS), της μάθησης (LG) και της ανταπόκρισης (RS) που μπορούν να προκύψουν σε κάποιο στάδιο του κύκλου ζωής του συστήματος ή σαν αποτέλεσμα των πρακτικών που χρησιμοποιούνται σε κάθε μεθοδολογία. Αν κάποια φάση ή μεθοδολογία υποστηρίζει ένα συγκεκριμένο χαρακτηριστικό ευελιξίας, τότε βάζουμε την τιμή ένα δίπλα από αυτό, ενώ σε διαφορετική περίπτωση παίρνει την τιμή μηδέν. [Asif Qumer, Brian Henderson – Sellers

“Comparative Evaluation of XP and Scrum Using The 4D Analytical Tool (4 – DAT)”,
2006, p 5]

XP	Χαρακτηριστικά Ευελιξίας					Σύνολο
	FY	SD	LS	LG	RS	
Φάση						
Έρευνας	1	1	0	1	1	4
Σχεδιασμού	1	1	0	1	1	4
Επανάληψης	1	1	0	1	1	4
Παραγωγής	1	1	1	1	1	5
Συντήρησης	1	0	0	1	1	3
Θανάτου	0	1	0	0	0	1
Σύνολο	5	5	1	5	5	21
Βαθμός Ευελιξίας	5/6	5/6	1/6	5/6	5/6	21/(6*5)
Πρακτικές						
Σχεδιασμού	1	1	0	1	1	4
Δοκιμαστικής Έκδοσης	1	1	0	1	1	4
Μεταφοράς	0	1	1	0	0	2
Υλοποίησης	1	1	1	1	1	5
Δοκιμής	1	1	0	1	1	4
Ανακατασκευής	1	1	1	1	1	5
Pair Programming	1	0	0	1	1	3
Collective Ownership	1	0	0	1	1	3
Continuous Integration	1	1	1	1	1	5
40ωρη εβδομάδα	0	0	0	1	0	1
On – site Customer	1	0	0	1	1	3
Πρότυπα Κωδικοποίησης	1	1	1	1	1	5
Σύνολο	10	8	5	11	10	44
Βαθμός Ευελιξίας	10/12	8/12	5/12	11/12	10/12	44/(12*5)

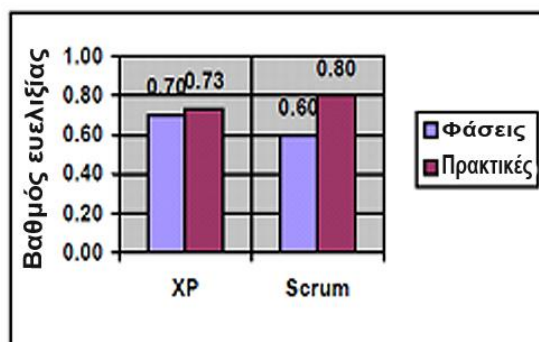
Πίνακας 18: Βαθμός ευελιξίας στο XP

Scrum	Χαρακτηριστικά Ευελιξίας					Σύνολο
	FY	SD	LS	LG	RS	
Φάση						
Πριν την υλοποίηση	1	1	0	1	1	4
Κατά την υλοποίηση	1	1	0	1	1	4
Μετά την υλοποίηση	0	1	0	0	0	1
Σύνολο	2	3	0	2	2	9
Βαθμός Ευελιξίας	2/3	3/3	0/3	2/3	2/3	9/(3*5)
Πρακτικές						
Scrum Master	1	1	0	1	1	4
Ομάδα Scrum	1	1	0	1	1	4
Ανεκτέλεστο υπόλοιπο προϊόντος	1	1	0	1	1	4
Sprint	1	1	0	1	1	4
Συνάντηση Σχεδιασμού Sprint	1	1	0	1	1	4
Καθημερινή Συνάντηση Scrum	1	1	0	1	1	4
Ανασκόπηση Sprint	1	1	0	1	1	4
Σύνολο	7	7	0	7	7	28
Βαθμός Ευελιξίας	7/7	7/7	0/7	7/7	7/7	28/(7*5)

Πίνακας 19: Βαθμός ευελιξίας στη Scrum

Διαδικασίες & Πρακτικές	XP	Scrum
Φάσεις	21/30=0,70	9/15=0,60
Κατάταξη	1	2
Πρακτικές	44/60=0,73	28/35=0,80
Κατάταξη	2	1

Πίνακας 20: Βαθμός ευελιξίας στο XP και στη Scrum συγκεντρικώς



Εικόνα 28: Βαθμός ευελιξίας στο XP και στη Scrum συγκεντρωτικά

Στο σημείο αυτό της αξιολόγησης εξετάζουμε τα χαρακτηριστικά της τρίτης διάστασης του εργαλείου 4 – DAT σε σχέση με την ευελιξία που παρέχουν στις δύο διαφορετικές μεθοδολογίες. Αυτό το οποίο παρατηρούμε από τον πίνακα που ακολουθεί είναι ότι για τα πρώτα τέσσερα χαρακτηριστικά τόσο η Scrum όσο και το XP παρέχουν επαρκή υποστήριξη, σε αντίθεση με τα δύο τελευταία χαρακτηριστικά. Πιο συγκεκριμένα για τη διατήρηση της ευελιξίας και του κόστους της διαδικασίας η πρακτική του eXtreme Programming δεν παρέχει καμία υποστήριξη. [Asif Qumer, Brian Henderson – Sellers “Comparative Evaluation of XP and Scrum Using The 4D Analytical Tool (4 – DAT)”, 2006, p 6]

Ευέλικτα Χαρακτηριστικά	XP	Scrum
Αλληλεπιδράσεις πάνω σε διαδικασίες και εργαλεία	<ol style="list-style-type: none"> 1. Σχεδιασμός 2. Pair Programming 3. Collective Ownership 4. On – site Customer 	<ol style="list-style-type: none"> 1. Ομάδες Scrum 2. Σχεδιασμός Συνάντησης Sprint 3. Καθημερινή συνάντηση Scrum
Λειτουργικό λογισμικό και κατανοητή τεκμηρίωση	<ol style="list-style-type: none"> 1. Δοκιμαστικής Έκδοσης 2. Δοκιμής 3. Continuous Integration 	<ol style="list-style-type: none"> 1. Sprint 2. Ανασκόπηση Sprint
Συνεργασία πελατών	<ol style="list-style-type: none"> 1. Σχεδιασμού 2. On – site Customer 	<ol style="list-style-type: none"> 1. Ανεκτέλεστο υπόλοιπο προϊόντος 2. Συνάντηση Σχεδιασμού Sprint

Ανταπόκριση σε αλλαγές	<ol style="list-style-type: none"> 1. Μεταφορά 2. Υλοποίηση 3. Ανακατασκευή 4. Πρότυπα Κωδικοποίησης 	<ol style="list-style-type: none"> 1. Ανασκόπηση Sprint 2. Συνάντηση Σχεδιασμού Sprint
Διατήρηση της ευελιξίας	-	<ol style="list-style-type: none"> 1. Ανασκόπηση Sprint 2. Καθημερινή συνάντηση Scrum
Διατήρηση του κόστους της διαδικασίας	-	-

Πίνακας 21: Βαθμός ευελιξίας στο XP και στη Scrum για την 3^η διάσταση

Στο σημείο αυτό της αξιολόγησης εξετάζουμε τα χαρακτηριστικά της τέταρτης διάστασης του εργαλείου 4 – DAT σε σχέση με την ευελιξία που παρέχουν στις δύο διαφορετικές μεθοδολογίες. [Asif Qumer, Brian Henderson – Sellers “Comparative Evaluation of XP and Scrum Using The 4D Analytical Tool (4 – DAT)”, 2006, p 7]

Διαδικασίες Λογισμικού	XP	Scrum
Ανάπτυξη	<ol style="list-style-type: none"> 1. Δοκιμαστικής Έκδοσης 2. Μεταφορά 3. Σχεδιασμός 4. Δοκιμή 5. Ανακατασκευή 6. Pair Programming 7. Continuous Integration 8. Πρότυπα Κωδικοποίησης 	<ol style="list-style-type: none"> 1. Ομάδες Scrum 2. Ανεκτέλεστο υπόλοιπο προϊόντος 3. Sprint 4. Ανασκόπηση Sprint
Διαχ/ση έργου	<ol style="list-style-type: none"> 1. Σχεδιασμού 	<ol style="list-style-type: none"> 1. Scrum Master 2. Συνάντηση σχεδιασμού Sprint 3. Καθημερινή συνάντηση Scrum

Τροποποίησης / Υποστήριξης	-	-
Διαχ/ση διαδικασίας	-	-

Πίνακας 22: Βαθμός ευελιξίας στο XP και στη Scrum για την 4^η διάσταση

Από όσα έχουμε αναφέρει μέχρι τώρα γίνεται εύκολα αντιληπτό ότι η μεθοδολογία του eXtreme Programming προσφέρει μεγαλύτερη ευελιξία κατά τις φάσεις του κύκλου ζωής ενός πληροφοριακού συστήματος, ενώ η μεθοδολογία Scrum παρέχει πιο ευέλικτες πρακτικές.

Στο σημείο αυτό θα επικεντρωθούμε σε μια μελέτη περίπτωσης (Case Study) για ένα Πληροφοριακό Σύστημα το οποίο υλοποιήθηκε κατά τη διάρκεια της φοίτησης στο Πρόγραμμα Μεταπτυχιακών Σπουδών «Διδακτική της Τεχνολογίας και Ψηφιακά Συστήματα» του τμήματος Ψηφιακών Συστημάτων του Πανεπιστημίου Πειραιώς. Κατά την ανάπτυξη του συγκεκριμένου πληροφοριακού συστήματος χρησιμοποιήθηκε ως μεθοδολογία ανάπτυξης αυτή του eXtreme Programming, ωστόσο εμείς στη δική μας περίπτωση θα τη συγκρίνουμε με τη μεθοδολογία Scrum έτσι ώστε να δούμε τι θα μπορούσε να είχε γίνει διαφορετικό και ποια προβλήματα θα μπορούσαν να είχαν αποφευχθεί αν χρησιμοποιούσαμε στη θέση της τη μεθοδολογία Scrum. Σε πρώτο στάδιο θα παρουσιάσουμε τα στάδια ανάπτυξης των πληροφοριακών συστημάτων που ισχύουν γενικά σε όλες τις εφαρμογές και στη συνέχεια θα εμβαθύνουμε στη σύγκριση των παραπάνω μεθοδολογιών.

Η μεθοδολογία ανάπτυξης μιας εμπορικής εφαρμογής δίνει τη δυνατότητα στην ομάδα εργασίας να έχει τον πλήρη έλεγχο όλων των φάσεων ανάπτυξης της εφαρμογής και να διασφαλίσει τον σωστό συντονισμό των ενεργειών που θα οδηγήσουν στην επιτυχία του έργου. Για να ολοκληρωθεί ένα έργο, το οποίο περιλαμβάνει την ανάπτυξη μιας εμπορικής εφαρμογής, είναι απαραίτητο να ακολουθηθούν κάποιες συγκεκριμένες διαδικασίες, που τοποθετούνται μέσα σε σαφώς καθορισμένα στάδια. Τα βασικά στάδια - φάσεις για την ολοκλήρωση μιας εμπορικής εφαρμογής παρουσιάζονται στην συνέχεια.

10.1 ΣΤΑΔΙΑ ΑΝΑΠΤΥΞΗΣ ΕΦΑΡΜΟΓΩΝ ΛΟΓΙΣΜΙΚΟΥ

Στάδιο 1: Ανάλυση αναγκών και απαιτήσεων

Πρόκειται για τον ορισμό του προβλήματος, καταγράφεται ένα σύνολο απαιτήσεων που συμφωνούνται μεταξύ του πελάτη και της ομάδας που παράγει το λογισμικό. Συνήθως η καταγραφή γίνεται σε φυσική γλώσσα και το αποτέλεσμα είναι ένα

κείμενο που περιγράφει σε γενικές γραμμές τι θα είναι το προϊόν αλλά και πώς θα δουλεύει ακριβώς ή πώς θα είναι δομημένο εσωτερικά.

Στάδιο 2: Προδιαγραφές

Συμβόλαιο Πελάτη - Ομάδας Προγραμματιστών, δηλαδή του τμήματος που αναπτύσσει τις τυπικές προδιαγραφές.

- Λεπτομέρειες τρόπου λειτουργίας.
- Περιγραφή συνδεδεμένων τμημάτων.
- Απαιτήσεις σε ζητήματα λειτουργίας.

Στάδιο 3: Πλάνο εργασιών – Χρονοπρογραμματισμός

Σε αυτό το στάδιο το έργο χωρίζεται σε πακέτα εργασίας, καθένα από τα οποία μπορεί να χωρισθεί σε μικρότερα κομμάτια, τις εργασίες. Οι εργασίες αυτές κατανομούνται χρονικά σε όλο το έργο και μπορεί κάποιος να τις διαχειρισθεί ανεξάρτητα. Οι εργασίες αποτελούν τα δομικά στοιχεία της διαχείρισης του έργου. Ο υπολογισμός και η κατανομή του χρόνου που θα απαιτήσει κάθε εργασία μπορεί να αναπαρασταθεί οπτικά πάνω σε ένα ημερολόγιο. Αυτό θα αποτελέσει και το πλάνο εργασιών του έργου (project plan). Στον χρονοπρογραμματισμό θα πρέπει πάντα να υπολογίζεται και ο χρόνος έγκρισης του αποτελέσματος των διάφορων φάσεων του έργου από τον πελάτη. Ο χρόνος αυτός είναι συνήθως αρκετά μεγάλος και οδηγεί σε επανασχεδιασμό της εργασίας.

Στάδιο 4: Σχεδιασμός

Επιλογή δομών δεδομένων, σύνδεση μεταξύ των διαφόρων τμημάτων του συστήματος και ροή ελέγχου μέσα στα τμήματα αυτά.

- Δίνει απάντηση στο ερώτημα του πώς θα δουλεύει το πρόγραμμα.
- Από το σωστό σχεδιασμό εξαρτάται η ευκολία υλοποίησης, ελέγχου και συντήρησης της εφαρμογής.
- Μερικές φορές το στάδιο αυτό περιλαμβάνει και την επιλογή γλώσσας προγραμματισμού ή εργαλείων.

Στάδιο 5: Υλοποίηση

Περιλαμβάνει την ανάπτυξη του κώδικα

- Δεν πρέπει να γράφεται γραμμή κώδικα εάν δεν έχουν λυθεί όλα τα σχεδιαστικά προβλήματα.
- Η στιγμή ανάπτυξης του κώδικα είναι και η στιγμή για να γραφτεί η αντίστοιχη τεκμηρίωση.

Στάδιο 6: Έλεγχος και εγκατάσταση εφαρμογής

Απαραίτητο στάδιο πριν την παράδοση του προϊόντος στον πελάτη

- Προτεινόμενη ανάλυση ελέγχων.
- Προσεκτική ανάγνωση του κώδικα (από κάποιο μέλος της ομάδας ανάπτυξης).
- Έλεγχος υπό-διαδικασιών (κάθε πρόγραμμα ελέγχεται ανεξάρτητα).
- Έλεγχος ενοτήτων (modules). Κάθε ομάδα υποπρογραμμάτων που ανήκουν σε ένα μεγαλύτερο ανεξάρτητο τμήμα ελέγχεται για σωστή επικοινωνία μεταξύ των υπό-διαδικασιών.
- Έλεγχος υποσυστήματος (Πολλά modules ελέγχονται μαζί).
- Έλεγχος συστήματος (Όλο το προϊόν ελέγχεται στο σύνολό του για να είναι βέβαιο ότι πληροί τις προδιαγραφές λειτουργίας του).
- Έλεγχος αποδοχής (Έλεγχος παρουσία του πελάτη με χρήση πραγματικών πλέον δεδομένων).
- Έλεγχος εγκατάστασης (Επιβεβαιώνεται η λειτουργία του συστήματος στο περιβάλλον που θα δουλεύει).

Στάδιο 7: Συντήρηση

Ακολουθεί αναπόφευκτα την παράδοση του συστήματος διακρίνεται σε δύο κατηγορίες:

- Συντήρηση για διόρθωση (λάθη ή παραλείψεις που έχουν διαφύγει από προηγούμενους ελέγχους ή δεν είχαν προβλεφθεί).
- Συντήρηση για προσαρμογή (για να ανταποκριθεί σε αλλαγές του επιχειρηματικού περιβάλλοντος).

10.2 ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΠΟΥ ΥΛΟΠΟΙΗΘΗΚΕ

Το πληροφοριακό σύστημα το οποίο έπρεπε να υλοποιηθεί ήταν μια εφαρμογή ηλεκτρονικής συνταγογράφησης, για την ολοκλήρωση της οποίας απασχολήθηκε μια ομάδα φοιτητών έξι ατόμων και για χρονικό διάστημα ενενήντα συνολικά ημερών. Αρχικά προβήκαμε σε καταμερισμό του συνόλου των εργασιών του προς υλοποίηση συστήματος και στη συνέχεια υπολογίσαμε εμπειρικά τον καλύτερο, το χειρότερο και τον εκτιμώμενο χρόνο για καθεμία εργασία ξεχωριστά. Βέλτιστος είναι ο χρόνος νωρίτερα από τον οποίο δεν μπορεί να ολοκληρωθεί κάποια δραστηριότητα, ενώ χειρότερος είναι ο χρόνος μετά από τον οποίο η προγραμματισμένη εργασία θα πρέπει να έχει τελειώσει έτσι ώστε να μη βγούμε εκτός χρονικού προγραμματισμού.

1. Μελέτη Σκοπιμότητας
2. Καταγραφή Απαιτήσεων
3. Ανάλυση Απαιτήσεων
4. Μοντελοποίηση Συστήματος
5. Επιλογή Τεχνολογιών
6. Σχεδίαση Β.Δ
7. Αρχιτεκτονική Λογισμικού
8. Κατασκευή Λογισμικού
9. Έλεγχος Συστήματος

Δραστηριότητα	Βέλτιστος Χρόνος	Εκτιμώμενος Χρόνος	Χειρότερος Χρόνος
Μελέτη Σκοπιμότητας	2	3	4
Καταγραφή Απαιτήσεων	1	2	3
Ανάλυση Απαιτήσεων	2	3	5
Μοντελοποίηση Συστήματος	3	5	5
Επιλογή Τεχνολογιών	1	1	2
Σχεδίαση Β.Δ.	3	5	7

Αρχιτεκτονική Λογισμικού	5	5	7
Κατασκευή Λογισμικού	35	40	50
Έλεγχος Συστήματος	5	5	7

Πίνακας 23: Πίνακας Εργασιών (Βέλτιστος, εκτιμώμενος, και χειρότερος χρόνος)

Η μεθοδολογία η οποία αρχικά επιλέχθηκε για την υλοποίηση του συστήματος ήταν αυτή του eXtreme Programming. Έτσι έγινε μια γρήγορη καταγραφή και ανάλυση των απαιτήσεων του συστήματος, όπως αποτυπώνεται στον παραπάνω πίνακα. Η εκτίμηση των μελών της ομάδας ήταν ότι το μεγαλύτερο βάρος θα πρέπει να δοθεί στη σωστή σχεδίαση της Βάσης Δεδομένων με την οποία θα επικοινωνεί το σύστημα καθώς επίσης και στην κατασκευή του συνολικού κώδικα της εφαρμογής. Με άλλα λόγια δεν δόθηκε η αρμόζουσα προσοχή στα τρία πρώτα στάδια του κύκλου ζωής του πληροφοριακού συστήματος (ανάλυση απαιτήσεων, καταγραφή προδιαγραφών και χρονοπρογραμματισμού). Η βιαστική καταγραφή και η ανάλυση των απαιτήσεων κόστισε τελικά στα μέλη της ομάδας μιας και κατά την ανάπτυξη του κώδικα της εφαρμογής κάποιες νέες διαδικασίες δεν μπορούσαν να ενσωματωθούν και να λειτουργούσαν ως σύνολο μαζί με τα υπόλοιπα τμήματα του κώδικα που είχαν μέχρι εκείνη τη στιγμή υλοποιηθεί.

Για το λόγο αυτό η ομάδα ανάπτυξης χρειάστηκε να διορθώνει αρκετές φορές μέρος του κώδικα που είχε υλοποιηθεί ή ακόμη και να μελετά και να συμπληρώνει εκ των υστέρων τις όποιες απαιτήσεις της εφαρμογής προέκυπταν. Όλα τα παραπάνω είχαν ως αποτέλεσμα το παραδοτέο πληροφοριακό σύστημα να μην ολοκληρωθεί εντός του χρονικού ορίου στο οποίο είχε αρχικά υπολογιστεί καθώς επίσης και κάποιες διαδικασίες του να μην ολοκληρωθούν όπως είχαν αρχικά σχεδιαστεί.

Τα παραπάνω προβλήματα θα μπορούσαν να είχαν αντιμετωπιστεί πιο αποτελεσματικά αν αντί για τη χρήση της μεθοδολογίας eXtreme Programming είχε χρησιμοποιηθεί η μεθοδολογία Scrum. Από την παραπάνω σύγκριση των δύο μεθοδολογιών καταλήξαμε στο συμπέρασμα ότι η μεθοδολογία XP παρουσιάζει μεγαλύτερη ευελιξία όσο αφορά τις φάσεις ανάπτυξης ενός πληροφοριακού

συστήματος ενώ η Scrum είναι πιο ευέλικτη όσο αφορά της πρακτικές που χρησιμοποιεί. Επομένως αν στη δική μας περίπτωση υπήρχαν πρακτικές αντίστοιχες με εκείνες του sprint, του ανεκτέλεστου υπόλοιπου ή της ανασκόπησης του sprint που χρησιμοποιούνται στη Scrum θα συνειδητοποιούσαμε πολύ πιο έγκαιρα κατά την ανασκόπηση του sprint ότι υπάρχουν ανεκτέλεστες εργασίες οι οποίες δεν έχουν ολοκληρωθεί καθώς επίσης ότι ο στόχος του sprint δεν έχει πλήρως επιτευχθεί. Επίσης κατά τη δοκιμαστική έκδοση του λογισμικού που χρησιμοποιούνται στη Scrum θα συνειδητοποιούσαμε πολύ πιο έγκαιρα ποια είναι τα τμήματα της εφαρμογής που πρέπει να διορθωθούν ώστε να ενσωματωθούν με επιτυχία με τα προηγούμενα τμήματα του συστήματος που έχουν ολοκληρωθεί. Επομένως με τη χρήση της Scrum θα είχαμε παραδώσει ένα πλήρως λειτουργικό σύστημα ηλεκτρονικής συνταγογράφησης on-time.

10.3 ΤΟ ΠΛΗΡΟΦΟΡΙΑΚΟ ΣΥΣΤΗΜΑ ΜΕ ΧΡΗΣΗ ΤΗΣ ΜΕΘΟΔΟΛΟΓΙΑΣ SCRUM

Στο σημείο αυτό και με τη βοήθεια των πινάκων που ακολουθούν θα καταλάβουμε ότι με την υιοθέτηση της μεθοδολογίας Scrum η παραπάνω εφαρμογή θα είχε καλύτερα αποτελέσματα. Στη μελέτη περίπτωσης περιλαμβάνονται όλα τα στάδια της Scrum που αναλύθηκαν όπως:

- Ανεκτέλεστο υπόλοιπο προϊόντος.
- Συνάντηση σχεδιασμού Sprint.
- Ανεκτέλεστο υπόλοιπο Sprint.
- Καθημερινή συνάντηση Scrum.
- Συνάντηση ανασκόπησης Sprint.

Το ανεκτέλεστο υπόλοιπο του προϊόντος είναι μια λίστα που περιέχει όλη την επιθυμητή λειτουργικότητα της εφαρμογής. Στο σημείο αυτό καταγράφονται οι βασικότερες απαιτήσεις της εφαρμογής και καθορίζεται επίσης και ο χρόνος της κάθε επανάληψης.

A/A	Καθήκοντα	Λεπτομέρειες	Προτεραιότητα	Αρχή	Τέλος	Κατάσταση
1	Περίγραμμα εφαρμογής	Περιεχόμενο πάνω, κάτω, δεξιά και αριστερά της εφαρμογής	Υψηλή			Ανενεργή
2	Menu εφαρμογής	Καθορισμός menu ανάλογα με απαιτήσεις	Υψηλή			Ανενεργή
3	Links	Καθορισμός συνδέσμων εφαρμογής	Υψηλή			Ανενεργή
4	Σχεδιασμός Β.Δ.	Όλα τα φάρμακα, όλες οι κατηγορίες	Υψηλή			Ανενεργή
5	Κατηγορίες	Εμφάνιση φαρμάκων ανά κατηγορία	Μεσαία			Ανενεργή
6	Αναζήτηση	Μπάρα αναζήτησης φαρμάκων ανά κατηγορία	Χαμηλή			Ανενεργή
Sprint 1 (Διάρκεια 2 Εβδομάδες)						
7	Διαχειριστής συστήματος	Περιβάλλον διαχειριστή	Υψηλή			Ανενεργή
8	Εισαγωγή στοιχείων	Καταχώριση φαρμάκων,	Υψηλή			Ανενεργή

		κατηγοριών και τιμών				
9	Φόρμα στοιχείων ασθενών	Όλα τα απαραίτητα στοιχεία του ασθενή	Υψηλή			Ανενεργή
10	Εισαγωγή ασθενή	Έλεγχος username και password	Μεσαία			Ανενεργή
11	Φόρμα στοιχείων ιατρού	Όλα τα απαραίτητα στοιχεία του χρήστη	Υψηλή			Ανενεργή
12	Εισαγωγή ιατρού	Έλεγχος username και password	Μεσαία			Ανενεργή
13	Φόρμα στοιχείων φαρμακείου	Όλα τα απαραίτητα στοιχεία του χρήστη	Υψηλή			Ανενεργή
14	Εισαγωγή φαρμακοποιού	Έλεγχος username και password	Μεσαία			Ανενεργή
Sprint 2 (Διάρκεια 2 εβδομάδες)						
15	Καλάθι αγορών	Προσθήκη φαρμάκων στο καλάθι	Υψηλή			Ανενεργή
16	Μερική ή ολική εκτέλεση συνταγής	Τι γίνεται σε περίπτωση μερικής εκτέλεσης;	Υψηλή			Ανενεργή

17	Υλοποίηση Πληρωμής	Συμμετοχή ασθενή και ταμείου	Υψηλή			Ανενεργή
18	Τρόπος Πληρωμής	Μετρητά ή πιστωτική	Υψηλή			Ανενεργή
Sprint 3 (Διάρκεια 2 εβδομάδες)						
19	...					
...	...					
...	...					

Πίνακας 24: Το Ανεκτέλεστο Υπόλοιπο του Προϊόντος

Εφόσον έχουμε τελειώσει με την καταγραφή των βασικότερων ενεργειών για το ανεκτέλεστο υπόλοιπο του προϊόντος, προχωράμε στη συνέχεια στη συνάντηση σχεδιασμού του sprint. Ο ιδιοκτήτης του Π.Σ. θα περιγράψει στην ομάδα τις λειτουργίες της εφαρμογής δίνοντας ιδιαίτερη έμφαση στις σημαντικότερες εργασίες. Ο ιδιοκτήτης του συστήματος δεν χρειάζεται να αναλύσει με κάθε λεπτομέρεια όλες τις εργασίες που αναφέρονται στο ανεκτέλεστο υπόλοιπο του προϊόντος.

Ο Scrum Master με τα μέλη της ομάδας θα πρέπει στη συνέχεια να προχωρήσουν στην υλοποίηση των εργασιών της πρώτης επανάληψης, διατηρώντας πάντοτε ενημερωμένο το ανεκτέλεστο υπόλοιπο του Sprint.

A/A	Καθήκοντα	Κατάσταση	Πρόοδος	Προβλήματα	Ημερομηνία
1	Περίγραμμα εφαρμογής	Ενεργή	25%	Κανένα	
2	Περίγραμμα εφαρμογής	Ενεργή	40%	Ασυμβατότητα με Opera browser	
3	Περίγραμμα εφαρμογής	Ολοκληρώθηκε	100%	Κανένα	

4	Menu εφαρμογής	Ενεργή	40%	Κανένα	
5	Menu εφαρμογής	Ολοκληρώθηκε	100%	Κανένα	
6	Links	Ολοκληρώθηκε	100%	Κανένα	
7	Σχεδιασμός Β.Δ.	Ενεργή	20%	Πρόβλημα σε queries της Β.Δ.	
8	Σχεδιασμός Β.Δ.	Ενεργή	45%	Κανένα	
9	Σχεδιασμός Β.Δ.	Ενεργή	65%	Πρόβλημα στην κατηγοριοποίηση	
10	Σχεδιασμός Β.Δ.	Ολοκληρώθηκε	100%	Κανένα	
11	Αναζήτηση	Ολοκληρώθηκε	100%	Κανένα	
1^ο Sprint Ολοκληρώθηκε					

Πίνακας 25: Το Ανεκτέλεστο Υπόλοιπο του 1^{ου} Sprint

Σε κάθε περίπτωση τα μέλη της κάθε ομάδας πρέπει να συμμετέχουν στις καθημερινές Scrum συναντήσεις αναφέροντας το έργο που ολοκληρώθηκε την προηγούμενη ημέρα, τα εμπόδια που προέκυψαν και τον τρόπο που τα αντιμετώπισαν. Στο τέλος της δεκαπεντάλεπτης αυτής συνάντησης θα πρέπει επίσης να συζητήσουν για τις εργασίες που πρόκειται να ολοκληρώσουν τη σημερινή ημέρα.

Εφόσον η πρώτη επανάληψη έχει ολοκληρωθεί η Scrum ομάδα θα πρέπει να έχει έτοιμο ένα τμήμα από τη συνολική εφαρμογή. Καλό θα είναι λοιπόν το τμήμα αυτό του λογισμικού να δοκιμάζεται, έτσι ώστε να έχουμε μια πιο πλήρη εικόνα σε σχέση με το επιθυμητό από το πραγματικό αποτέλεσμα.

Δοκιμή	Ενέργεια	Κατάσταση	Σχόλια
Έλεγχος συμβατότητας	Mozilla Firefox	Επιτυχία	Σωστή λειτουργία
Έλεγχος συμβατότητας	Internet Explorer	Επιτυχία	Σωστή λειτουργία
Έλεγχος συμβατότητας	Opera	Αποτυχία	Πρόβλημα στην ανάλυση
Έλεγχος συμβατότητας	Google Chrome	Επιτυχία	Σωστή λειτουργία
Έλεγχος Β.Δ.	Εισαγωγή	Επιτυχία	Σωστή λειτουργία
Έλεγχος Β.Δ.	Διαγραφή	Επιτυχία	Σωστή λειτουργία
Έλεγχος Β.Δ.	Ενημέρωση	Επιτυχία	Σωστή λειτουργία
Έλεγχος Β.Δ.	Σύνθετο query	Αποτυχία	Πρόβλημα σε κάποιους πίνακες
Έλεγχος αναζήτησης	Λέξη	Επιτυχία	Σωστή λειτουργία
Έλεγχος αναζήτησης	Πρόταση	Επιτυχία	Σωστή λειτουργία

Πίνακας 26: Δοκιμή της πρώτης επανάληψης της εφαρμογής

Από τον έλεγχο που έγινε για το πρώτο τμήμα της εφαρμογής που ολοκληρώθηκε προέκυψαν κάποια σφάλματα τα οποία δεν είχαν αντιληφθεί τα μέλη της ομάδας στη διάρκεια του πρώτου Sprint. Στη μεθοδολογία Scrum αυτό που γενικότερα εφαρμόζεται είναι να αφιερώσουμε λίγο χρόνο για να διορθώσουμε σφάλματα από το προηγούμενο Sprint πριν ξεκινήσει το επόμενο. Με τον τρόπο αυτό εξασφαλίζουμε ένα πλήρως λειτουργικό τμήμα του συνολικού συστήματος, χωρίς να συσσωρεύονται σφάλματα που μπορούν να αποδειχθούν δύσκολα και χρονοβόρα στην επίλυσή τους. Τα πραγματικά αποτελέσματα από το 1^ο Sprint φαίνονται στον πίνακα που ακολουθεί.

Καθήκοντα	Επιθυμητή πρόοδος	Πραγματική πρόοδος
Περίγραμμα εφαρμογής, menu & links	100%	100%
Σχεδιασμός Β.Δ.	100%	85%
Αναζήτηση	100%	100%

Πίνακας 27: Επιθυμητή και Πραγματική πρόοδος

Εφόσον όλα τα παραπάνω έχουν ολοκληρωθεί, αυτό που απομένει είναι να πραγματοποιηθεί η συνάντηση ανασκόπησης του Sprint. Στη συνάντηση αυτή συμμετέχουν ο Scrum Master, η ομάδα Scrum και ο ιδιοκτήτης του προϊόντος. Κύριος άξονας της ανασκόπησης αποτελεί η παρουσίαση από την ομάδα Scrum για όσα επιτεύχθηκαν στο προηγούμενο sprint. Αυτό γίνεται κυρίως με την παρουσίαση ενός τμήματος από το σύνολο της εφαρμογής. Η διαδικασία που αναλύσαμε μέχρι τώρα (από το ανεκτέλεστο υπόλοιπο προϊόντος μέχρι και τη συνάντηση ανασκόπησης του sprint)επαναλαμβάνεται για το σύνολο των επαναλήψεων μέχρι τελικά να φτάσουμε στην ολοκλήρωση του συστήματός μας.

A/A	Καθήκοντα	Κατάσταση	Πρόοδος	Προβλήματα	Ημερομηνία
1	Περιβάλλον διαχείρισης	Ολοκληρώθηκε	100%	Κανένα	
2	Εισαγωγή στοιχείων διαχειριστή	Ενεργή	35%	Στην κατηγοριοποίηση των φαρμάκων	
3	Εισαγωγή στοιχείων διαχειριστή	Ολοκληρώθηκε	100%	Κανένα	
4	Φόρμα στοιχείων ασθενή	Ολοκληρώθηκε	100%	Κανένα	
5	Εισαγωγή στοιχείων ασθενή	Ενεργή	40%	Το λάθος συνθηματικό δεν εμφανίζει μήνυμα	
6	Εισαγωγή στοιχείων ασθενή	Ολοκληρώθηκε	100%	Κανένα	
7	Φόρμα στοιχείων ιατρού	Ολοκληρώθηκε	100%	Κανένα	

8	Εισαγωγή στοιχείων ιατρού	Ενεργή	40%	Το λάθος συνθηματικό δεν εμφανίζει μήνυμα	
9	Εισαγωγή στοιχείων ιατρού	Ολοκληρώθηκε	100%	Κανένα	
10	Φόρμα στοιχείων φαρμακείου	Ολοκληρώθηκε	100%	Κανένα	
11	Εισαγωγή στοιχείων φαρμακείου	Ολοκληρώθηκε	100%	Κανένα	
2^ο Sprint Ολοκληρώθηκε					

Πίνακας 28: Το Ανεκτέλεστο Υπόλοιπο του 2^{ου} Sprint

Δοκιμή	Ενέργεια	Κατάσταση	Σχόλια
Πιστοποίηση διαχειριστή	Έλεγχος username και password	Επιτυχία	Σωστή λειτουργία
Πιστοποίηση ασθενή	Έλεγχος username και password	Επιτυχία	Σωστή λειτουργία
Πιστοποίηση ιατρού	Έλεγχος username και password	Επιτυχία	Σωστή λειτουργία
Πιστοποίηση φαρμακοποιού	Έλεγχος username και password	Επιτυχία	Σωστή λειτουργία
Αναζήτηση φαρμάκων με βάση την κατηγορία	Επικοινωνία με Β.Δ.	Αποτυχία	Πρόβλημα σε sql ερώτημα

Πίνακας 29: Δοκιμή της δεύτερης επανάληψης της εφαρμογής

Καθήκοντα	Επιθυμητή πρόοδος	Πραγματική πρόοδος
Περιβάλλον διαχείρισης	100%	100%
Εισαγωγή στοιχείων διαχειριστή	100%	80%
Φόρμες στοιχείων ασθενών, ιατρών, φαρμακείων	100%	100%

Πίνακας 30: Επιθυμητή και Πραγματική πρόοδος

A/A	Καθήκοντα	Κατάσταση	Πρόοδος	Προβλήματα	Ημερομηνία
1	Καλάθι αγορών	Ολοκληρώθηκε	100%	Κανένα	
2	Μερική ή ολική εκτέλεση	Ενεργή	20%	Το Π.Σ. δεν κάνει μερική εκτέλεση συνταγής	
3	Μερική ή ολική εκτέλεση	Ενεργή	50%	Κανένα	
4	Μερική ή ολική εκτέλεση	Ολοκληρώθηκε	100%	Κανένα	
5	Υλοποίηση πληρωμής	Ενεργή	15%	Ίδια συμμετοχή σε όλα τα ταμεία	
6	Υλοποίηση πληρωμής	Ενεργή	40%	Κανένα	
7	Υλοποίηση πληρωμής	Ολοκληρώθηκε	100%	Κανένα	
8	Τρόπος πληρωμής	Ενεργή	30%	Πρόβλημα στην πληρωμή με πιστωτική	
9	Τρόπος πληρωμής	Ολοκληρώθηκε	100%	Κανένα	
3^o Sprint Ολοκληρώθηκε					

Πίνακας 31: Το Ανεκτέλεστο Υπόλοιπο του 3^{ου} Sprint

Δοκιμή	Ενέργεια	Κατάσταση	Σχόλια
Έλεγχος του καλαθιού αγοράς	Προσθήκη φαρμάκων στο καλάθι	Επιτυχία	Σωστή λειτουργία
Ολική εκτέλεση συνταγής	Προσθήκη φαρμάκων στο καλάθι	Επιτυχία	Σωστή λειτουργία
Μερική εκτέλεση συνταγής	Εκτέλεση μέρους της συνταγής	Αποτυχία	Το Π.Σ. ζητάει και τα υπόλοιπα φάρμακα
Υπολογισμός ποσού πληρωμής	Καταχώριση συνταγής	Αποτυχία	Ίδια συμμετοχή σε όλα τα ταμεία
Πληρωμή με πιστωτική	Ταυτοποίηση στοιχείων κάρτας	Επιτυχία	Σωστή λειτουργία

Πίνακας 32: Δοκιμή της τρίτης επανάληψης της εφαρμογής

Καθήκοντα	Επιθυμητή πρόοδος	Πραγματική πρόοδος
Καλάθι αγοράς	100%	100%
Εκτέλεση συνταγής	100%	50%
Τρόπος και ποσό πληρωμής	100%	80%

Πίνακας 33: Επιθυμητή και Πραγματική πρόοδος

Η μεθοδολογία Scrum είναι μια μεθοδολογία διαχείρισης πληροφοριακών συστημάτων και ανήκει σε μια ευρέως διαδεδομένη οικογένεια μεθοδολογιών που ονομάζονται ευέλικτες. Αποτελεί μια πλήρως οργανωμένη προσέγγιση με βασικό της χαρακτηριστικό την ανάπτυξη ποιοτικού και λειτουργικού λογισμικού μέσα από ένα αριθμό επαναλήψεων. Στο τέλος κάθε επανάληψης πρέπει να έχει παραχθεί ένα πλήρως λειτουργικό τμήμα από το σύνολο της εφαρμογής, μέχρι να φτάσουμε στην τελική έκδοση του προϊόντος.

Περιλαμβάνει δοκιμασμένες πρακτικές και μεθόδους ελέγχου που εγγυώνται την κατασκευή ποιοτικού λογισμικού εντός προκαθορισμένων χρονικών διαστημάτων. Επίσης διατηρεί πάντοτε ενημερωμένη τη λίστα του ανεκτέλεστου υπολοίπου της εφαρμογής και του ανεκτέλεστου υπολοίπου του sprint και απαιτεί την ενεργή και ουσιαστική συμμετοχή όλων των μελών της ομάδας καθώς επίσης και των όλων των εμπλεκόμενων μερών (ιδιοκτήτης, τελικοί χρήστες). Σημαντικό επίσης είναι και το γεγονός ότι μπορεί να εφαρμοστεί σε έργα οποιουδήποτε μεγέθους.

Η εφαρμογή της μεθοδολογίας Scrum σε ένα σύγχρονο επιχειρησιακό περιβάλλον μπορεί να κρύβει κινδύνους αλλά όχι δυσάρεστα αποτελέσματα για τον ιδιοκτήτη του συστήματος, την ομάδα Scrum ή τους τελικούς χρήστες. Η ενθάρρυνση για τη συμμετοχή όλων των εμπλεκόμενων μερών καθ' όλη τη διάρκεια του έργου, εξασφαλίζει λογισμικό υψηλής ποιότητας και λειτουργικότητας. Επίσης η εκτίμηση των αποτελεσμάτων από τη χρήση των scrum τεχνικών μας επιτρέπει οποιαδήποτε χρονική στιγμή να γνωρίζουμε με ακρίβεια το τι υλοποιήθηκε σε σχέση με το τι θα έπρεπε να είχε υλοποιηθεί.

Είναι γεγονός ότι τα παραδοσιακά μοντέλα αποτελούν καλή επιλογή σε έργα χωρίς σημαντική πολυπλοκότητα. Αντίθετα οι ευέλικτες μέθοδοι αναπτύχθηκαν για να ανταποκρίνονται αποτελεσματικά σε δυναμικά περιβάλλοντα όπου η αναπροσαρμογή του αρχικού σχεδίου αποτελεί το ζητούμενο και όχι το απροσδόκητο. Στα σύγχρονα επιχειρησιακά περιβάλλοντα είναι σχεδόν απίθανο να

μπορούν να αποτυπωθούν με ακρίβεια από την αρχή όλες οι προδιαγραφές του συστήματος χωρίς να χρειαστεί να τροποποιηθούν. Για το λόγο αυτό θα πρέπει να χρησιμοποιούνται μεθοδολογίες οι οποίες επιτρέπουν ενδεχόμενες αλλαγές και αναπροσαρμογές με βάση το αρχικό πλάνο.

Παρόλο που η μεθοδολογία Scrum οριοθετείται από σαφή και διακριτά βήματα, ίσως υστερεί στο γεγονός ότι είναι λίγο δύσκολη η πρακτική εφαρμογή της και αυτό γιατί αποτελεί ριζική αλλαγή στον τρόπο λειτουργίας των ομάδων. Όμως, το συνεργατικό πνεύμα μεταξύ των μελών της ομάδας καθώς επίσης και η συμμετοχή όλων των εμπλεκόμενων μερών στη συνολική διαδικασία, διαμορφώνουν μια εξαιρετική προοπτική που αιτιολογεί την επιλογή και τη χρήση της Scrum μεθοδολογίας. Η καθημερινή ανταλλαγή απόψεων και η αναθεώρηση των παραδοσιακών προσεγγίσεων με τον ιδιοκτήτη να έχει ρόλο εξωτερικού παρατηρητή, έχει σαν αποτέλεσμα τη δημιουργία μιας ομάδας με κύριο στόχο την ανάπτυξη λειτουργικών εφαρμογών.

Συμπερασματικά στην ερώτηση ποια είναι η καταλληλότερη μεθοδολογία η απάντηση μπορεί να είναι ιδιαίτερα πολύπλοκη, μιας και η κάθε εφαρμογή παρουσιάζει τις δικές της ιδιαιτερότητες. Όμως, οι ευέλικτες μέθοδοι παρέχουν πρακτικές που ανταποκρίνονται με επιτυχία στα συνεχώς μεταβαλλόμενα επιχειρησιακά περιβάλλοντα.

Ωστόσο, προσωπική μου επιθυμία είναι η παρούσα εργασία να δώσει επαρκή ερεθίσματα σε επαγγελματίες του κλάδου της πληροφορικής για προσθετική υιοθέτηση και χρήση της Scrum μεθοδολογίας. Γιατί η κατασκευή ποιοτικών και άρτιων πληροφοριακών συστημάτων είναι τόσο επιτακτική όσο ποτέ άλλωστε ιδιαίτερα στην εποχή που ζούμε.

Jamie Allsop, “Foundations of agile development”, 2008.

Dan Rawsthorne, “Comparing/Combining RUP, XP, and Scrum – mixing the Process Cocktail”, 2003.

Ken Schwaber and Jeff Sutherland, “The scrum guide developed and sustained”, 2010.

Ken Schwaber and Jeff Sutherland, “The Definitive Guide to Scrum: The Rules of the Game”, 2011.

Henrik Kniberg, “Scrum and XP from the Trenches”, 2006.

Asif Qumer, Brian Henderson – Sellers “Comparative Evaluation of XP and Scrum Using The 4D Analytical Tool (4 – DAT)”, 2006.

Jeff Sutherland and Ken Schwaber “The Scrum Papers: Nut, Bolts and Origins of an Agile development”, 2011.

The Scrum Papers: Origins of an Agile Framework by Jeff Sutherland and Ken Schwaber.

Mike Beedle, Martine Devos, Yonat Sharon, Ken Schwaber, Jeff Sutherland, “SCRUM an extension pattern language for hyperproductive software development”.

User Stories Applied For Agile Software Development.

Ken Schwaber, “Scrum Development Process”.

<http://www.mountangoatsoftware.com/topics/scrum>

<http://www.mountangoatsoftware.com/scrum/a-reusable-scrum-presentation>

[http://en.wikipedia.org/wiki/Scrum_\(development\)](http://en.wikipedia.org/wiki/Scrum_(development))

<http://scrummethodology.com/>

<http://www.selectbs.com/process-maturity/what-is-scrum-development>

<http://www.codeproject.com/Articles/4798/What-is-SCRUM>

<http://www.netobjectives.com/>

<http://agilemanifesto.org/>