



ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ ΣΤΗ ΔΙΟΙΚΗΣΗ
ΕΠΙΧΕΙΡΗΣΕΩΝ

Διπλωματική Εργασία

Εφαρμογή Agile τεχνικών για την ανάπτυξη λογισμικού

Του Τρυφέρνη Παναγιώτη Α.Μ.: mbx 22003

Επιβλέπων καθηγητής: Λουκάς Τσιρώνης

Θεσσαλονίκη 2024

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ	3
ΠΕΡΙΛΗΨΗ	4
ΕΙΣΑΓΩΓΗ.....	5
ΜΕΘΟΔΟΛΟΓΙΑ ΕΡΕΥΝΑΣ.....	6
ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΑΝΑΣΚΟΠΗΣΗ – ΕΠΙΣΚΟΠΗΣΗ.....	7
ΚΕΦΑΛΑΙΟ 1 ^ο : AGILE ΜΕΘΟΔΟΛΟΓΙΕΣ.....	7
1. ΕΙΣΑΓΩΓΙΚΑ ΣΤΟΙΧΕΙΑ	7
2. ΒΑΣΙΚΕΣ ΑΞΙΕΣ ΤΟΥ AGILE MANIFESTO	8
3. ΟΙ ΔΩΔΕΚΑ ΑΡΧΕΣ ΤΟΥ AGILE MANIFESTO	9
4. ΚΥΡΙΟΤΕΡΕΣ ΜΕΘΟΔΟΛΟΓΙΕΣ AGILE.....	10
4.1. Scrum.....	11
4.1.1. Ορισμός του Scrum.....	11
4.1.2. Θεωρία του Scrum	12
4.1.3. Ομάδα και αξίες του Scrum	13
4.1.4. Εκδηλώσεις του Scrum	17
4.1.5. Scrum Artifacts	19
4.2. Kanban.....	21
4.3. Extreme Programming.....	23
4.3.1. Οι αξίες του Extreme Programming	24
4.3.2. Πρακτικές του Extreme Programming	25
4.3.3. Οι ρόλοι του Extreme Programming	28
IV.4. Lean Software Development.....	28
4.4.1. Κεντρικές αρχές της Lean Software Development.....	29
4.5. Dynamic Systems Development Method (DSDM).....	30
4.5.1. Βασικές αρχές DSDM.....	31
4.5.2. Αξίες που διέπουν την DSDM.....	32
4.5.3. Πρακτικές που χρησιμοποιούνται στην DSDM	33
4.5.4. Οι φάσεις του DSDM.....	34
4.5.5. Οι Ρόλοι στο DSDM.....	35
4.6. Feature-Driven Development (FDD).....	37
4.6.1. Οι ρόλοι που υπάρχουν στην FDD	38
4.6.2. Οι πρακτικές που χρησιμοποιούνται στην FDD	39
4.6.3. Οι αξίες της FDD	41
4.6.4. Οι αρχές της FDD	41

4.7	Crystal Family Methodologies	42
4.7.1	Οι ρόλοι της Crystal.....	44
4.7.2	Οι πρακτικές της Crystal.....	45
4.7.3	Οι αξίες της Crystal	46
4.7.4	Οι αρχές της Crystal.....	47
4.8	Adaptive Software Development (ASD).....	48
4.8.1	Οι πρακτικές της ASD	48
4.8.2	Οι ρόλοι της ASD	49
IV.8.3	Οι αξίες της ASD	50
5	ΜΕΛΕΤΗ ΠΕΡΙΠΤΩΣΗΣ (CASE STUDY).....	51
5.1	Εισαγωγή	51
5.2	Εκπαίδευση της ομάδας.....	52
5.3	Scrum Meetings	52
5.4	Εκκίνηση του έργου	53
5.5	Ευρήματα κατά την διάρκεια των Sprints	55
5.6	Αποτελέσματα	55
5.7	Σύγκριση με την παραδοσιακή μεθοδολογία που χρησιμοποιούσαμε στο παρελθόν	56
6	ΣΥΝΕΝΤΕΥΞΕΙΣ	57
6.1	Ερωτήσεις – Απαντήσεις – Σχολιασμός.....	57
7	ΣΥΜΠΕΡΑΣΜΑΤΑ	64
	ΒΙΒΛΙΟΓΡΑΦΙΑ	65

ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ

Εικόνα 1. Επισκόπηση του Scrum Πηγή: <https://startinfinity.com/project-management-methodologies/scrum> 12
Εικόνα 2. Οι αξίες του Scrum Πηγή: Scrum.org 14
Εικόνα 3. Οι ρόλοι στην ομάδα του Scrum Πηγή: <https://www.scrum.org/resources/blog/how-do-3-scrum-roles-promote-self-organization> 17
Εικόνα 4. Product Backlog Πηγή: <https://donetonic.com/product-backlog-vs-sprint-backlog/> 20
Εικόνα 5. Scrum Artifacts Πηγή: <https://www.atlassian.com/agile/scrum/artifacts> 21
Εικόνα 6. Kanban Board Πηγή: Wikipedia.org 23
Εικόνα 7 Planning/Feedback XP loops Πηγή: [wikimedia.org](https://www.wikimedia.org) 27
Εικόνα 8 Επισκόπηση Lean Πηγή: <https://kruschecompany.com/lean-software-development/> 30
Εικόνα 9 Οι αρχές της DSDM Πηγή: <https://thecompetenza.com/agile-development-methodology/> 32
Εικόνα 10 Οι πρακτικές της DSDM Πηγή: https://www.researchgate.net/figure/521-Dynamic-Systems-Development-Methodology-DSDM-Source_fig13_344458644 35
Εικόνα 11 Οι ρόλοι της DSDM Πηγή: <https://www.agilebusiness.org/dsdm-project->

framework/roles-and-responsibilities.html 37Εικόνα 12 Οι πρακτικές της FDD
Πηγή: <https://lvivivity.com/7-things-about-feature-driven-development> 40Εικόνα
13 Κατανομή μεγέθους ομάδας με χρώμα Crystal. Πηγή:
<https://activecollab.com/blog/project-management/crystal-methods> 44Εικόνα 14
Committed vs Completed Story Points. Πηγή: Jira Server 55

ΠΕΡΙΛΗΨΗ

Αυτή η ερευνητική εργασία θα στοχεύσει στην εξερεύνηση της σημαντικότητας των agile μεθοδολογιών στην ανάπτυξη λογισμικού και κατα πόσο έχει επεκταθεί στο κόσμο των προγραμματιστών. Με το λογισμικό να παίζει όλο και περισσότερο ρόλο στην τεχνολογική ζωή μας όλες οι εταιρίες επαναπροσδιορίζουν την οργάνωση τους και ψάχνουν τους πιο αποδοτικότερους τρόπους για να διαχειριστούν τέτοιου είδους έργα ώστε να μπορούν να προσπεράσουν ή να συμβαδίσουν με τον ανταγωνισμό. Μετά την παρουσίαση των πιο δημοφιλή εύελικτων μεθοδολογιών θα προχωρήσουμε στην έρευνα μιας μελέτης περίπτωσης όπου στην εταιρία που εργάζομαι δοκιμάσαμε την μεθοδολογία Scrum + Kanban για την ανάπτυξη ενός σημαντικού έργου που σχετίζοταν με την ανάπτυξη λογισμικού και με δύσκολη προθεσμία. Ακόμα θα ακολουθήσει μια συνέντευξη από τα άτομα που συμμετείχαν ώστε να μπορείτε να δείτε ποιές ήταν οι εντυπώσεις των μελών της ομάδας που συμμετείχαν είτε για πρώτη φορά κάποιοι είτε όχι ώστε να καταλάβετε αν βοήθησε ή όχι πραγματικά τα μέλη της ομάδας.

Λέξεις κλειδιά: Scrum, Μελέτη περίπτωσης, Ευέλικτη μεθοδολογία, agile, διοίκηση έργου, Μέθοδοι διοίκησης έργων

Τρυφέρνης Παναγιώτης, 2024

ΕΙΣΑΓΩΓΗ

Στον 21ο αιώνα ζούμε σε μια εποχή συνεχών προηγμένων ψηφιακών ανακαλύψεων. Απο την τεράστια άνοδο της τεχνητής νοημοσύνης που μας απασχολεί όλο και περισσότερο μέρα με την μέρα μέχρι την εξάπλωση του cloud computing και του Internet of things. Αυτό το ασταμάτητο κύμα καινοτομίας έχει επηρεάσει σημαντικά όλες τις πτυχές της ζωής μας και ακόμα περισσότερο τον τρόπο λειτουργίας των επιχειρήσεων.

Ενας απο τους τομείς που έχει επηρεαστεί βαθιά από αυτό το μετασχηματικό τοπίο είναι η ανάπτυξη λογισμικού. Οι παραδοσιακές προσεγγίσεις στη διαχείριση έργων συχνά αντιμετωπίζει δυσκολίες και δεν μπορούν να ακολουθήσουν την συνεχόμενη τεχνολογική πρόοδο. Ως απάντηση σε αυτή την πρόοδο αναπτύχθηκαν οι agile μεθοδολογίες προσφέροντας στις επιχειρήσεις ένα ευέλικτο και προσαρμοστικό τρόπο για την αποτελεσματική διαχείριση των έργων ανάπτυξης λογισμικού.

Οι agile μεθοδολογίες περιλαμβάνουν ένα σύνολο αρχών και πρακτικών που προάγουν την συνεργασία, την προσαρμοστικότητα και την συνεχή βελτίωση στις ομάδες ανάπτυξης. Αυτες οι μεθοδολογίες βασίζονται στην επαναληπτική ανάπτυξη, στις τακτικές ανατροφοδοτήσεις και στην στενή συνεργασία των λειτουργικών ομάδων ώστε να ανταποκρίνονται γρήγορα στις μεταβαλλόμενες απαιτήσεις της αγοράς και της τεχνολογικής προόδου.

Σε μια εποχή που οι προσδοκίες των πελατών αλλάζουν συνέχεια, ο χρόνος μέχρι την κυκλοφορία είναι κρίσιμος και καινοτόμες ιδέες προκύπτουν από το πουθενά όλες οι επιχειρήσεις πρέπει να έχουν την δυνατότητα γρήγορης προσαρμογής για να επιβιώσουν και να αναπτυχθούν. Οι agile μεθοδολογίες είναι αυτές που παρέχουν την οργάνωση και τα απαραίτητα εργαλεία ώστε να αντιμετωπίσουν αυτές τις δυσκολίες και να τους επιτρέψει να έχουν το δυνατόν υψηλότερο επίπεδο αποτελεσματικότητας, παραγωγικότητας και ικανοποίησης των πελατών.

Τελικά αυτή η εργασία θέλει να τονίσει και να αποδείξει την σημασία της υιοθέτησης agile μεθοδολογιών σε όλες τις επιχειρήσεις και ιδιαίτερα αυτές που

αναπτύσουν λογισμικό για να μπορούν να ευδοκιμήσουν και να αντιμετωπίσουν την ανελέητη τεχνολογική ανάπτυξη του 21ου αιώνα.

ΜΕΘΟΔΟΛΟΓΙΑ ΕΡΕΥΝΑΣ

Η παρούσα διπλωματική εργασία έχει στηριχθεί σε 2 τρόπους εισαγωγής δεδομένων όπου θα παρουσιάζουν τα αποτελέσματα της agile μεθοδολογίας.

Συγκεκριμένα θα υπάρχουν πηγές δεδομένων ένα ερωτηματολόγιο που θα έχει σκοπό μέσα από τις απαντήσεις των ατόμων που ασχολούνται με την ανάπτυξη λογισμικού να καταλάβουμε το ποσοστό επέκτασης στην Ελλάδα αυτών των μεθοδολογιών, αν προτιμούνται σε σχέση με τον παραδοσιακό τρόπο διαχείρισης έργων, κατα πόσο βελτιώνουν την παραγωγικότητα των ίδιων, αν έχει βοηθήσει στο να επιτεύχονται οι στόχοι του έργου, αν γνωρίζουν για αυτές τις μεθόδους οι ίδιοι και αν χρησιμοποιούν κάποιες ευέλικτες μεθοδολογίες ποιες είναι αυτές;

Δεύτερον θα υπάρξει ένα case study που έγινε στην εταιρεία που εργάζομαι και για πρώτη φορά χρησιμοποιήθηκε μια μορφή Scrum – Kanban όπου την τροποποιήσαμε στα μέτρα μας για να μας βοηθάει και θα δείξουμε τα αποτελέσματα της και την βελτίωση της γενικής διαχείρισης του έργου.

Στην συνέχεια θα έχουμε εισαγωγή δεδομένων από τις συνεντεύξεις των ατόμων που συμμετείχαν στο case study ώστε να δούμε τις απόψεις τους πάνω στην μεθοδολογία που χρησιμοποιήσαμε και αν ένιωσαν βελτίωση στην παραγωγικότητα και οργάνωση του έργου.

Τέλος ελπίζω στο τέλος αυτής της ερευνητικής εργασίας να έχω βοηθήσει τον αναγνώστη να καταλάβει πόσο αποτελεσματική και σημαντική είναι η διαχείριση των έργων με την κατάλληλη μεθοδολογία για το καθένα ανάλογα και μέσω του case study να καταλάβει πως και ο ίδιος μπορεί να εφαρμόσει αυτές τις μεθοδολογίες στην ομάδα του.

ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΑΝΑΣΚΟΠΗΣΗ – ΕΠΙΣΚΟΠΗΣΗ

ΚΕΦΑΛΑΙΟ 1^ο : AGILE ΜΕΘΟΔΟΛΟΓΙΕΣ

1. ΕΙΣΑΓΩΓΙΚΑ ΣΤΟΙΧΕΙΑ

Οι μεθοδολογίες διαχείρισης έργων περιλαμβάνουν διάφορες προσεγγίσεις. Η πιο παραδοσιακή και κλασική που υπάρχει από την δημιουργία της επιστήμης διαχείρισης έργου είναι η waterfall και είναι μια διαδοχική διαδικασία όπου κάθε βήμα πρέπει να ολοκληρωθεί πριν αρχίσει το επόμενο βήμα (Καπόπουλος, Δ., Κομισόπουλος, Φ., Παλόγλου, Α. 2020). Ωστόσο αυτή η μεθοδολογία δεν είναι κατάλληλη για ένα περιβάλλον ανάπτυξης λογισμικού αφού είναι πολύ αυστηρή και πολύ λίγο προσαρμόσιμη όταν χρειάζεται κάτι να μεταβληθεί.

Λόγω της ύπαρξης μονο ενος παραδοσιακού και δυσκίνητου τρόπου διοίκησης έργου αναπτύχθηκε η ανάγκη για την δημιουργία πιο agile μεθοδολογιών. Δείγματα agile μεθοδολογιών βλέπουμε απο το 1980 όπου άρχισαν οι εταιρείες ανάπτυξης λογισμικού να καταλαβαίνουν ότι υπάρχουν πολλές αδυναμίες στην παραδοσιακή μέθοδο όταν την χρησιμοποιούν για την ανάπτυξη κώδικα.(Heeager, Svejvig, and Schlichter 2016).

Το 2001 ιδρύθηκε και επίσημα η agile μέθοδος μετά από την συνάντηση μιας ομάδας 17 ατόμων όπου εργάζονταν στον τομέα ανάπτυξης λογισμικού και όλοι μαζί δημιούργησαν το κείμενο που έγινε γνωστό ως agile manifesto. Σε αυτό το κείμενο καταγράφονται οι βασικές αρχές της μεθοδολογίας και μια πρωτοφανής αντίληψη στην διοίκηση έργων.

Σύμφωνα με την έρευνα 16th Annual state of agile report οι ομάδες που χρησιμοποιούν τις μεθόδους agile το κάνουν κατα 52% επειδή επιταχύνει τον χρόνο εισαγωγής στην αγορά, το 44% επειδή είναι πιο εύκολο να εκτιμηθεί το πως θα είναι το προϊόν στο τελικό του επίπεδο και 31% επειδή έχει μικρότερο ρίσκο σε σχέση με άλλες μεθοδολογίες. (digital.ai. (2022). 16th Annual State of Agile Report. In [https://digital.ai/resource-center/analyst-reports/state-of-agile-report/.](https://digital.ai/resource-center/analyst-reports/state-of-agile-report/))

2. ΒΑΣΙΚΕΣ ΑΞΙΕΣ ΤΟΥ AGILE MANIFESTO

Οι 4 βασικές αξίες του agile manifesto είναι οι εξής (agilemanifesto.org, 2001):

1. Τα άτομα και οι αλληλεπιδράσεις πάνω από τις διαδικασίες και τα εργαλεία
2. Το λογισμικό που λειτουργεί πάνω από την εκτενή τεκμηρίωση
3. Η συνεργασία με τον πελάτη πάνω από της συμβατικές διαπραγματεύσεις
4. Η ανταπόκριση στην αλλαγή πάνω από την πιστή εκτέλεση ενός πλάνου

Η πρώτη βασική αξία μας δείχνει ότι η agile δίνει προτεραιότητα στον άνθρωπο. Πιστεύει ότι πρέπει να αφήσουν τους ανθρώπους να παίρνουν πρωτοβουλίες να λειτουργούν μόνοι τους στις κατάλληλες συνθήκες και τότε η παραγωγικότητα τους είναι μεγαλύτερη από όταν ακολουθούν τυφλά ένα σχέδιο που τους έχει δοθεί από την αρχή για να ακολουθήσουν. Επιπλέον μιλάει και για τα εργαλεία που χρησιμοποιούνται για την οργάνωση ότι χρειάζονται αλλά δεν μπορούν να αντικαταστήσουν την συνάντηση από κοντά είτε από κάποια web conference εφαρμογή.

Η δεύτερη αξία μας μιλάει για την σημαντικότητα το παραγόμενο προϊόν να έχει καλύψει τις απαιτήσεις του πελάτη και η ομάδα να επικεντρώνεται στην παραγωγή αυτού που ζητήθηκε ακριβώς και όχι σε άλλες επιπρόσθετες εργασίες που δεν ωφελούν

Η τρίτη αξία εξηγεί ότι ο πελάτης είναι συνεργάτης και όχι αντίπαλός μας όπως γίνεται πολλές φορές. Είναι συνεργάτης και πρέπει να συμμετάσχει στη ανάπτυξη και δημιουργία του έργου συχνά, από την στιγμή όπου μπορεί να έχουμε συνεχείς αλλαγές σε ένα περιβάλλον που είναι ευέλικτο.

Η τέταρτη και τελευταία αξία μας εξηγεί τον βασικό παράγοντα ενός ευέλικτου περιβάλλοντος. Η αλλαγή είναι πάνω από το σχέδιο και την επιζητούμε στην agile. Αυτή η δυσκολίες σε αλλαγές στο waterfall τρόπο ανάπτυξης είναι που παρακίνησε την αλλαγή και την δημιουργία του agile όπου η αλλαγή είναι βασικό στοιχείο της και είναι και προβλέψιμη.

3. ΟΙ ΔΩΔΕΚΑ ΑΡΧΕΣ ΤΟΥ AGILE MANIFESTO

Οι δώδεκα αρχές του Agile manifesto είναι (agilemanifesto.org, 2001):

1. Η υψηλότερη προτεραιότητα είναι η ικανοποίηση του πελάτη μέσα από ταχύς και συνεχή παράδοση πολύτιμου λογισμικού.
2. Οι αλλαγές στα προαπαιτούμενα είναι καλοδεχούμενες οποιαδήποτε στιγμή ακόμα και αν είναι αργά στην ανάπτυξη. Οι agile διεργασίες πάντα χρησιμοποιούν την αλλαγή προς ανταγωνιστικό όφελος του πελάτη.
3. Παραδώστε λογισμικό που λειτουργεί συνέχεια με συχνότητα από μερικές εβδομάδες μέχρι μερικούς μήνες με προτίμηση τα μικρότερα χρονοδιαγράμματα πάντα.
4. Οι προγραμματιστές και τα στελέχη της εταιρείας πρέπει να συνεργάζονται καθημερινά κατά την διάρκεια του έργου.
5. Κάνε έργα γύρω από ανθρώπους που είναι παρακινημένοι. Δώσε τους το περιβάλλον και την υποστήριξη που χρειάζονται και εμπιστευσου τους να κάνουν το έργο.
6. Ο πιο αποτελεσματικός και αποδοτικός τρόπος για να μεταφερθεί η πληροφορία προς και μέσα στην ομάδα ανάπτυξης είναι η συζήτηση πρόσωπο με πρόσωπο.
7. Η βασική μέτρηση προόδου είναι η λειτουργικότητα του λειτουργικού.
8. Οι διεργασίες του agile προωθούν την βιώσιμη ανάπτυξη. Οι χρηματοδότες, οι προγραμματιστές και οι χρήστες πρέπει να είναι σε θέση να διατηρούν ένα σταθερό ρυθμό για πάντα.
9. Η συνεχής προσοχή στην τεχνολογική τελειότητα και στον καλό σχεδιασμό ενισχύει την ευελιξία.
10. Η τέχνη του να μεγιστοποιεί την ποσότητα της δουλειάς που δεν χρειάζεται να γίνει είναι αναγκαιότητα. Η απλότητα.
11. Οι καλύτερες αρχιτεκτονικές, απαιτήσεις και σχέδια αναδύονται από ομάδες που αυτό-οργανώνονται.

12. Ανα συγκεκριμένα διαστήματα η ομάδα πρέπει να αναρωτιέται πως μπορούν να γίνουν πιο παραγωγικοί και μετά να ρυθμίσουν και να προσαρμόσουν τη συμπεριφορά τους ανάλογα τα ευρήματα.

Αυτές είναι οι αρχές που οποιοσδήποτε θέλει να ασχοληθεί με τον χώρο της agile μεθοδολογία πρέπει να ξέρει και να ακολουθεί. Όλες οι agile μεθοδολογίες έχουν χτιστεί πάνω σε αυτές τις αρχές και τις βασικές αξίες που αναφέραμε προηγουμένως. Παρακάτω θα δείτε μερικές από αυτές.

4. ΚΥΡΙΟΤΕΡΕΣ ΜΕΘΟΔΟΛΟΓΙΕΣ AGILE

Υπάρχουν πολλές μεθοδολογίες agile αλλά θα σας αναλύσω μερικές από τις κύριες που υπάρχουν εδώ. Οι δύο κύριες είναι η Scrum και η Kanban μεθοδολογία. Θα αναλύσουμε αρκετά διεξοδικά την Scrum μιας και είναι αυτή που χρησιμοποιεί το 87% των agile ομάδων(digital.ai. (2022). 16th Annual State of Agile Report. In <https://digital.ai/resource-center/analyst-reports/state-of-agile-report/>).

Θα αναλύσουμε συγκεκριμένα 8 από αυτές όπου θα τις αναλύσουμε προς τα βασικά τους στοιχεία που περιγράφουμε τις agile μεθοδολογίες τα οποία είναι:

- Ορισμός
- Βασικές αρχές
- Αξίες που διέπουν
- Πρακτικές που χρησιμοποιούνται
- Οι φάσεις τους
- Οι ρόλοι των μελών

Σε κάποια θα μιλήσουμε για όλα σε κάποια άλλα όχι για να αποφύγουμε τυχόν επαναλήψεις σε κάποια από αυτά που εκπροσωπούν 2 ή 3 βασικά τους στοιχεία.

4.1. Scrum

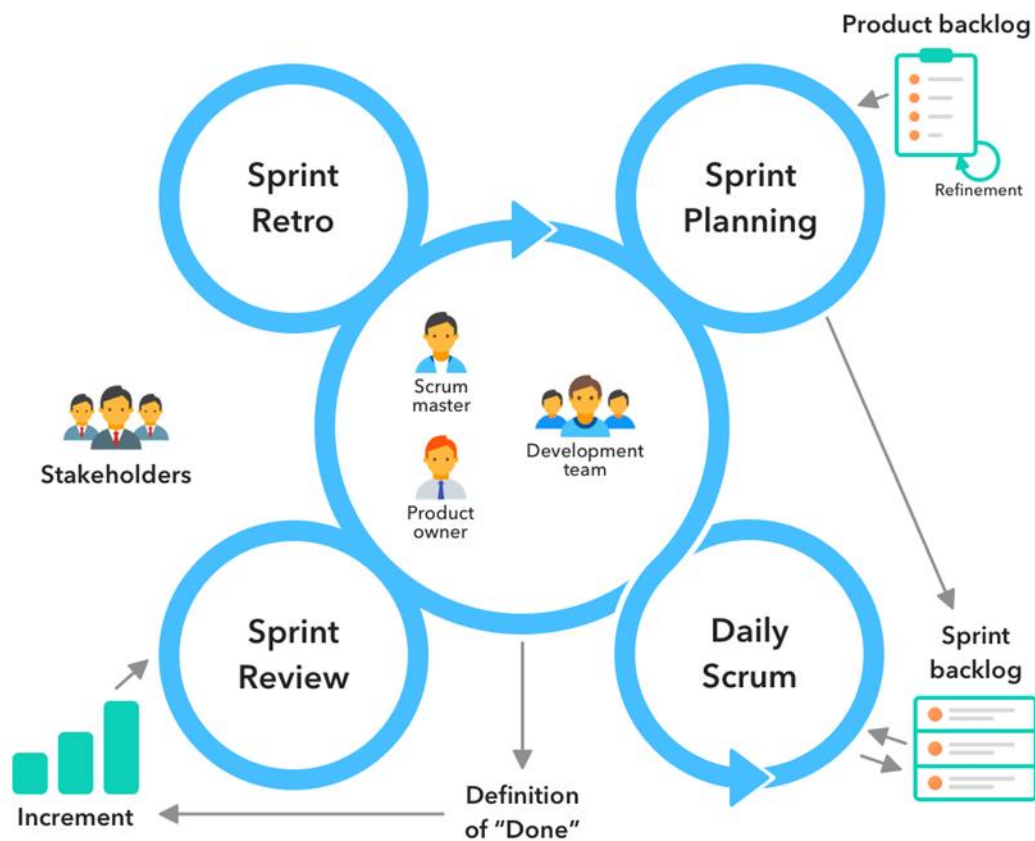
4.1.1. Ορισμός του Scrum

Η Scrum έχει δημιουργηθεί από τον Ken Schwaber & Jeff Sutherland. Στις αρχές του 1990 άρχισε αλλά επίσημα η πρώτη έκδοση του οδηγού τους βγήκε το 2010 (scrumguides.org) όπου έχει ανανεωθεί αρκετές φορές μέχρι την πιο πρόσφατη έκδοση τον Νοέμβριο του 2020. Εκεί μέσα εξηγούν τα βασικά συστατικά του Scrum. Το όνομα "Scrum" προέρχεται από τον όρο του ράγκμπι που χρησιμοποιείται για να περιγράψει τον τρόπο που οι ομάδες συνεργάζονται για να κερδίσουν τη μπάλα.

Ο κεντρικός στόχος του Scrum είναι να βελτιώσει την αποτελεσματικότητα της ομάδας και της διαδικασίας ανάπτυξης, εστιάζοντας στην αυξημένη παραγωγικότητα και την ικανότητα προσαρμογής σε αλλαγές. Η μέθοδος Scrum βασίζεται σε μια σειρά από ρόλους, διαδικασίες και εκδόσεις προϊόντος. Περιληπτικά η scrum χρειάζεται έναν Scrum Master όπου θα δημιουργήσει ένα περιβάλλον όπου θα έχουμε:

- Έναν Product Owner όπου θα βάζει τις εργασίες που πρέπει να γίνουν και την προτεραιότητα τους μέσα στο Product Backlog.
- Την δημιουργική ομάδα που θα δημιουργεί αξία μέσω των ενεργειών που θα επιλέγονται να γίνονται μέσα σε ένα διάστημα sprint.
- Η ομάδα και όλα τα ενδιαφερόμενα μέρη της επιθεωρούν τα αποτελέσματα και προσαρμόζονται για το επόμενο sprint.
- Επανάληψη όλων πάλι.

Η Scrum είναι απλή. Ο οδηγός γενικώς είναι σκόπιμα ελλιπές μιας και βασίζεται στην συλλογική εμπειρία και νοημοσύνη των ανθρώπων που το χρησιμοποιούν. Σε αυτό το πλαίσιο μπορεί να συνδυαστεί με πολλές διαδικασίες, τεχνικές και μεθόδους. (The Scrum Guide. Ken Schwaber and Jeff Sutherland, 2010(2020 Updated))



Εικόνα 1. Επισκόπηση του Scrum Πηγή: <https://startinfinity.com/project-management-methodologies/scrum>

4.1.2. Θεωρία του Scrum

Η Scrum έχει βασιστεί πάνω στον εμπειρισμό και την λιτή σκέψη. Ο εμπειρισμός υποστηρίζει ότι η γνώση μας προέρχεται από εμπειρία και λήψη αποφάσεων σύμφωνα με όλα αυτά που παρατηρούμε. Η λιτή σκέψη από την άλλη εστιάζει στα βασικά και μειώνει την σπατάλη χρόνου αλλά και χρήματος.

Χρησιμοποιεί μια επαναληπτική, σταδιακή προσέγγιση για την βελτίωση της πρόβλεψης και του ελέγχου του κινδύνου. Δεσμεύει μια ομάδα να έχουν συλλογικά όλες τις δεξιότητες που χρειάζονται και την τεχνογνωσία ώστε να γίνει η δουλειά.

Βασίζεται πάνω στους πυλώνες της διαφάνειας, της επιθεώρησης και προσαρμογής που χρησιμοποιεί 4 γεγονότα για να συμβούν αυτά που προσαρμόζονται γύρω από το χρονικό διάστημα που ονομάζουμε Sprint.

- **Διαφάνεια:** Όλες οι διαδικασίες και η εργασία πρέπει να είναι ορατή σε όλους και σε αυτούς που την εκτελούν αλλά και σε αυτούς που την λαμβάνουν. Οι

σημαντικές αποφάσεις βασίζονται στα 3 επίσημα artifacts (Product Backlog, Increment, Sprint Backlog) όπου υπάρχει διαφάνεια και επιτρέπει την επιθεώρηση αλλιώς θα ήταν παραπλανητική και σπατάλη χρόνου.

- **Επιθεώρηση:** Τα artifacts και η πρόοδος προς του στόχους πρέπει να επιθεωρούνται συνεχώς και με προσοχή για να εντοπίζονται γρήγορα τυχόν αποκλίσεις ή προβλήματα. Η επιθεώρηση επιτρέπει να υπάρχει προσαρμογή διότι αλλιώς θα ήταν άσκοπη.
- **Προσαρμογή:** Αν κάποια διαδικασία ή το προϊόν που βγαίνει δεν είναι αποδεκτό η προσαρμογή πρέπει να γίνει το συντομότερο δυνατό για να μειωθεί οποιαδήποτε περαιτέρω απόκλιση μπορεί να δημιουργηθεί. Η ομάδα μας είναι έτοιμη οποιαδήποτε στιγμή εντοπιστεί μια απόκλιση για να αντιδράσουν και να κάνουν την προσαρμογή που χρειάζεται σε αντίθεση με κάποια ομάδα που δεν είναι εξουσιοδοτημένη ή αυτοδιαχειριζόμενη για να το αντιμετωπίσει.

4.1.3. Ομάδα και αξίες του Scrum

- **Δέσμευση:** Δεσμεύονται να πετύχουν τους στόχους και να υποστηρίζει ο ένας τον άλλον.
- **Συγκέντρωση:** Συγκέντρωση στο έργο του Sprint και στην καλύτερη επίτευξη αυτού.
- **Ειλικρίνεια:** Η ομάδα είναι ειλικρινείς σε ότι έχει σχέση με την δουλειά και τις προκλήσεις που δημιουργούνται.
- **Σεβασμός:** Όλα τα μέλη της ομάδας σέβονται το ένα το άλλο ως άτομα που έχουν τις δεξιότητες που χρειάζονται για να κάνουν την εργασία που τους έχει ανατεθεί και σαν ανεξάρτητα άτομα.
- **Κουράγιο:** Κουράγιο για να κάνουν αυτό που πρέπει και να δουλέψουν σε δύσκολα προβλήματα που μπορεί να χρειαστεί να λύσουν στο έργο.



Εικόνα 2. Οι αξίες του Scrum Πηγή: Scrum.org

Αυτές οι αξίες δίνουν κατεύθυνση στην Ομάδα Scrum όσον αφορά τη δουλειά, τις ενέργειες και τη συμπεριφορά τους. Οι αποφάσεις που λαμβάνονται, τα βήματα που λαμβάνονται και ο τρόπος που χρησιμοποιείται το Scrum θα πρέπει να ενισχύουν αυτές τις αξίες, όχι να τις μειώνουν ή να τις υπονομεύουν. Ο συνδυασμός αυτών των αξιών μαζί με τους 3 πυλώνες που είπαμε πριν χτίζει την εμπιστοσύνη μέσα στην ομάδα έργου.

Μια ομάδα Scrum συνήθως είναι μια μικρή ομάδα μέχρι 10 άτομα το πολύ ώστε να μπορεί να είναι ευκίνητη αλλά ταυτόχρονα αρκετά μεγάλη για να ολοκληρώσει ότι χρειάζεται και αποτελείται από έναν Scrum Master, Product Owner και τα μέλη της ομάδας που συνήθως είναι προγραμματιστές. Όλα τα μέλη είναι εξοπλισμένα με τις απαραίτητες δεξιότητες που χρειάζεται το έργο για να ολοκληρωθεί και πολλές φορές μπορούν πολλά άτομα να κάνουν τα ίδια αλλά αποφασίζουν ποιος θα κάνει τι με στόχο την ολοκλήρωση του προϊόντος. Αν οι ομάδες είναι μεγαλύτερες θα πρέπει να εξετάσουν αν θα πρέπει να σπάσουν σε μικρότερες ομάδες με κοινό στόχο αλλά διαφορετικές εργασίες ώστε να υπάρχει καλύτερη οργάνωση.

Η ομάδα είναι υπεύθυνη για όλα όσα σχετίζονται με το έργο και είναι εξουσιοδοτημένοι από την εταιρία να διαχειρίζονται όπως πιστεύουν αυτοί το έργο. Η εργασία τους σε Sprint τους βοηθάει να βελτιώνουν τον ρυθμό και την συνέπεια της

ομάδας αλλά ταυτόχρονα είναι και υπεύθυνοι για την παραγωγή ενός χρήσιμου παραδοτέου σε κάθε Sprint που ολοκληρώνεται.

- I. **Προγραμματιστές – Μέλη της ομάδας:** Τα άτομα της ομάδας που δημιουργούν το προϊόν και είναι υπεύθυνα για το παραδοτέο στο τέλος κάθε sprint. Είναι υπεύθυνοι για:
 - Δημιουργία και σχεδιασμός του Sprint. (Sprint backlog)
 - Εξασφάλιση ποιότητας με την βοήθεια του τι έχουν ορίσει σαν παραδοταίο.
 - Προσαρμόζουν τις εργασίες κάθε μέρα προς τον στόχο του Sprint.
 - Λογοδοτεί ο ένας στον άλλο σαν επαγγελματίες
- II. **Product Owner:** Είναι υπεύθυνος για την μεγιστοποίηση του προϊόντος που βγαίνει από την ομάδα. Είναι επίσης υπεύθυνος για την διαχείριση του Product Backlog αλλά είναι υπεύθυνος και για τα παρακάτω:
 - Ανάπτυξη και επικοινωνία του στόχου του προϊόντος που παράγουμε
 - Δημιουργία και κατανοητή διασύνδεση των αντικειμένων του Product Backlog.
 - Ιεράρχηση των εργασιών που πρέπει να γίνουν από το αυτό που πρέπει να γίνει νωρίτερα από το επόμενο μέχρι το τελευταίο.
 - Εξασφαλίζει ότι το Product Backlog είναι ορατό σε όλους και κατανοητό.

Οποιαδήποτε αλλαγή είναι να γίνει στο έργο και στις εργασίες το πρέπει να περάσουν πρώτα από αυτόν οποιος και αν θέλει να τις κάνει. Για να έχει επιτυχία ο ρόλος του πρέπει όλοι να σέβονται τις αποφάσεις του και να τις ακολουθούν.

- III. **Scrum Master:** Είναι το άτομο που είναι υπεύθυνο να δημιουργήσει το Scrum στην ομάδα και που ορίζεται επικεφαλής της Scrum. Βοηθάει όλη την ομάδα να κατανοήσει την θεωρία και το πρακτικό κομμάτι της Scrum. Είναι αυτός που είναι υπεύθυνος για την αποτελεσματικότητα της ομάδας και της αύξησης της μέσω της βελτίωσης το διαδικασιών της μεθοδολογίας. Είναι οι ηγέτες που υπηρετούν την ομάδα.

Ο Scrum Master βοηθάει την ομάδα με διάφορους τρόπους:

- Καθοδηγεί τα μέλη σε θέματα διαλειτουργικότητας και αυτοδιαχείρισης

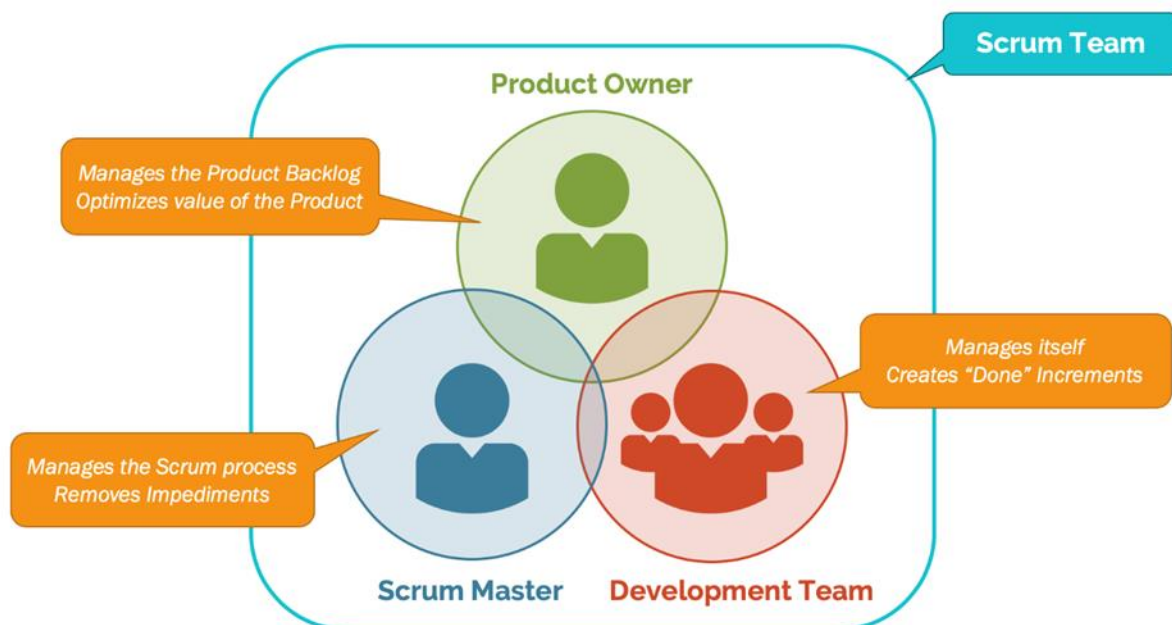
- Βοηθάει την ομάδα να είναι συγκεντρωμένη στην δημιουργία υψηλής αξίας παραδοτέων
- Είναι υπεύθυνος να εξαφανίζει τα εμπόδια που μπορεί να επηρεάζουν την πρόοδο της ομάδας
- Διασφαλίζει ότι τηρούνται όλα τα συμβάντα της Scrum και γίνονται με παραγωγικό τρόπο εντός χρονομετρημένων ορίων.

Μια απο τις υποχρεώσεις του είναι να βοηθάει και τον Product Owner σε μια πληθώρα πραγμάτων όπως:

- Βοηθάει στην εύρεση τεχνικών για τον σωστό καθορισμό στόχων και διαχείρισης του Product Backlog
- Βοηθάει την ομάδα να κατανοήσει ποσο σημαντικό είναι να είναι σαφή και συνοπτικά τα στοιχεία στο Product Backlog
- Βοηθάει στον εμπειρικό σχεδιασμό του προϊόντος σε ένα σύνθετο περιβάλλον
- Διευκολύνει την συνεργασία όλων των ενδιαφερόμενων μερών

Επίσης εντός του οργανισμού εξυπηρετεί σε πολλά θέματα όπως:

- Ηγεσία, καθοδήγηση και εκπαίδευση της εταιρείας στην Scrum
- Σχεδιασμός και συμβουλές πάνω στην εφαρμογή της Scrum
- Βοηθάει τα ενδιαφέροντα μέλη και τους εργαζόμενους να καταλάβουν την εμπειρική προσέγγιση της
- Εξαφανίζει τα εμπόδια μεταξύ των ενδιαφερόμενων μερών και των ομάδων Scrum



Εικόνα 3. Οι ρόλοι στην ομάδα του Scrum Πηγή: <https://www.scrum.org/resources/blog/how-do-3-scrum-roles-promote-self-organization>

4.1.4. Εκδηλώσεις του Scrum

Όλα τα συμβάντα της Scrum έχουν ως στόχο την ελαχιστοποίηση εξωτερικών συναντήσεων και έχουν σχεδιαστεί να γίνονται με τέτοιο τρόπο ώστε να υπάρχει η απαιτούμενη διαφάνεια.

I. Sprint

Είναι η εκδήλωση που εμπεριέχει όλες τις άλλες ή αλλιώς η καρδιά της Scrum. Έχει σταθερή διάρκεια όπου αποφασίζεται στην αρχή από την ομάδα και είναι από ένα μήνα και κάτω για να δημιουργηθεί συνέπεια. Ένα νέο Sprint ξεκινάει κατευθείαν μετά από το άλλο που θα κλείσει. Όλη η εργασία και όλα τα άλλα γεγονότα γίνονται κατά την διάρκεια του. Επίσης κατά την διάρκεια του:

- Δεν γίνονται αλλαγές που μπορεί να θέσουν σε κίνδυνο τον σκοπό του
- Η ποιότητα του δεν μειώνεται
- Το Product Backlog βελτιώνεται απο τον Product Owner όπως απαιτείται
- Ο στόχος του μπορεί να επαναδιαπραγματευτεί με τον Product Owner καθώς μαθαίνουν περισσότερα

Κάθε Sprint θα μπορούσε να θεωρηθεί ένα μικρό έργο. Αν και υπάρχουν πολλοί τρόποι να κάνεις πρόβλεψη για το μέλλον ο εμπειρισμός πάντα έχει σημαντική θέση και χρησιμοποιείται. Ο μόνος που έχει δύναμη να ακυρώσει ένα sprint σε περίπτωση που ο στόχος είναι παρωχημένος είναι ο Product Owner.

II. Sprint Planning

Το Sprint Planning είναι αυτό που ξεκινάει το Sprint παρουσιάζοντας την εργασία που έχει να γίνει μέσα σε αυτό. Αυτό προκύπτει από την συζήτηση ολόκληρης της ομάδας. Ο Product Owner έχει φροντίσει όλοι ξέρουν για τι πρέπει να έχουν προετοιμάσει για την συνάντηση και ομάδα έχει την δυνατότητα να καλέσει και άτομα εκτός της ομάδας για να δώσουν τις συμβουλές τους.

Στην αρχή ο Product Owner προτείνει πως να αυξήσουν την αξία του προϊόντος σε αυτό το Sprint και αποφασίζεται ο στόχος του Sprint και γιατί ο το συγκεκριμένο Sprint είναι πολύτιμο στα ενδιαφερόμενα μέρη. Μετά η συζήτηση συνεχίζεται και αποφασίζεται τι από τις εργασίες που είναι να γίνουν θα μπει στο συγκεκριμένο sprint και επίσης μπορεί κατά την διάρκεια αυτής της ενέργειας να τα βελτιώνουν και να διορθώνουν αν χρειάζεται κάπου ταυτόχρονα. Το δύσκολο σε αυτό είναι ο όγκος εργασιών που θα πρέπει να γίνει σε αυτό το διάστημα. Ωστόσο όσο πιο πολύ γνωρίζει η ομάδα από το παρελθόν τις δυνατότητές της είτε από προηγούμενα sprint του ίδιου έργου αυτό γίνεται όλο και πιο ακριβές. Κάποιες φορές μπορεί να χρειαστεί κάποιες εργασίες να διασπαστούν σε μικρότερες εργασίες στο τέλος του meeting μπαίνουν όσες εργασίες είναι να γίνουν στο Sprint Backlog όπου χρησιμοποιείται από όλους για να βλέπουν τι έχει ανατεθεί στους ίδιους.

Το συγκεκριμένο συμβάν χρειάζεται το περισσότερο 8 ώρες για ένα sprint που είναι διάρκειας μήνα και όσο μικρότερο το sprint τόσο μικρότερο πρέπει να είναι και το planning.

III. Daily Scrum

Ο σκοπός του είναι να βλέπει όλη η ομάδα την πρόοδο προς το στόχο του Spring και να προσαρμόζουν ότι πρέπει ανάλογα. Διαρκεί 15 λεπτά και για να μειωθεί η πολυπλοκότητα του συνήθως προτείνεται να είναι την ίδια ώρα κάθε εργάσιμη μέρα και θα μπορούσε να πραγματοποιείται και χωρίς τον Product Owner και τον Scrum Master αφού απασχολεί μόνο τους προγραμματιστές εκτός αν έχουν κάποιο

παραδοτέο και οι ίδιοι. Επίσης εδώ μοιράζονται τα εμπόδια που μπορεί να αντιμετωπίζει ο καθένας και τα μεταφέρουν στον Scrum Master για τους βοηθήσει να τα αντιμετωπίσουν.

IV. Sprint Review

Επιθεωρεί το αποτέλεσμα το Sprint και καθορίζει τι αλλαγές θα κάνουμε στα μελλοντικά. Επίσης παρουσιάζεται από την ομάδα η δουλειά που έχει γίνει στα ενδιαφέροντα μέρη και συζητείται η γενική πρόοδο προς το προϊόν. Το Product Backlog επίσης προσαρμόζεται ανάλογα με αυτά που ειπώθηκαν στο Review. Χρονολογείται ότι πρέπει να είναι το περισσότερο 4 ώρες για ένα μηνιαίο Sprint και λιγότερο για μικρότερης διάρκειας Sprints.

V. Sprint Retrospective

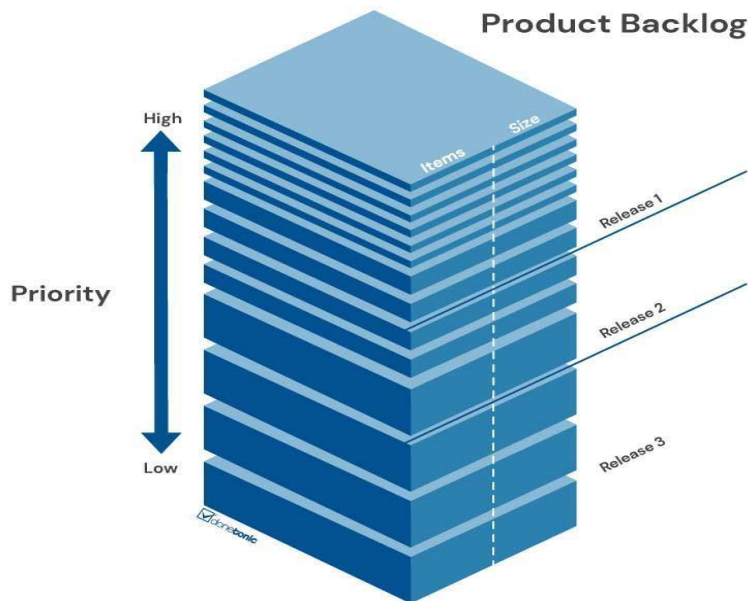
Γίνεται με στόχο να βελτιωθεί η ποιότητα και η αποτελεσματικότητα της ομάδας. Η ομάδα επιθεωρεί πως πήγε το Sprint σε σχέση με όλες τις μεταβλητές και συζητάνε τι πήγε καλά σε αυτό το Sprint και ποια προβλήματα αντιμετώπισανε και πως επιλύθηκαν ή ποια είναι αυτά που δεν επιλύθηκαν. Έτσι εντοπίζουν χρήσιμες αλλαγές που χρειάζονται να γίνουν ώστε να υπάρχει βελτίωση. Προτείνετε να έχει διάρκεια 3 ώρες για ένα μηνιαίο Sprint και λιγότερο ανάλογα για τα μικρότερα sprint.

4.1.5. Scrum Artifacts

Τα Scrum Artifacts είναι δημιουργήματα που αντιπροσωπεύουν το έργο και έχουν σχεδιαστεί για να μεγιστοποιούν την διαφάνεια των πληροφοριών. Καθημερινά όλοι να επιβλέπουν και προσαρμόζουν τα πάντα ανάλογα. Έχουμε 3 Scrum Artifacts:

I. Product Backlog

Είναι μια λίστα όπου ταξινομημένα μπαίνουν οι εργασίες που πρέπει να γίνουν για την ανάπτυξη του προϊόντος. Είναι η μοναδική πηγή εργασιών για την ομάδα. Από εδώ επιλέγονται οι δουλειές που είναι να γίνουν σε ένα Sprint.



Εικόνα 4. Product Backlog Πηγή: <https://donetonic.com/product-backlog-vs-sprint-backlog/>

II. Sprint Backlog

Το Sprint Backlog αποτελείται από τον στόχο του Sprint, το σύνολο των εργασιών που επιλέχθηκαν από το Product Backlog για να μουν σε αυτό και ένα σχέδιο για το πως θα γίνουν οι εργασίες ώστε να φτάσουν στο παραδοτέο. Σε αυτό φαίνεται ξεκάθαρα μια εικόνα για τον πραγματικό χρόνο των εργασιών που πρέπει να γίνουν μέσα στο Sprint για να ολοκληρωθεί.

III. Increment

Είναι το σκαλοπάτι που γίνεται σε κάθε επανάληψη (Increment) του Spring για να φτάσει στο στόχο του προϊόντος. Κάθε επανάληψη (Increment) πρέπει να είναι χρήσιμη και να παράγει έργο. Μπορεί σε ένα Sprint να γίνουν πολλαπλές όπου παρουσιάζονται σε κάθε Sprint review. Για να θεωρηθεί μια εργασία ότι είναι μέρος της πρέπει να καλύπτει τον ορισμό ολοκλήρωσης μιας εργασίας που έχει θέσει η ομάδα από την αρχή.

SCRUM ARTIFACTS



Εικόνα 5. Scrum Artifacts Πηγή: <https://www.atlassian.com/agile/scrum/artifacts>

4.2. Kanban

Το Kanban δημιουργήθηκε από την Toyota στην Ιαπωνία κατά τη διάρκεια της δεκαετίας του 1940 και του 1950. Η βασική ιδέα του Kanban προέκυψε από τις πρακτικές της "σούπερ-αγοράς" (supermarket) στην Toyota, όπου προμηθευτές προσφέραν ανταλλάξιμα αντικείμενα σε κάρτες με βάση την πραγματική κατανάλωση.

Το Kanban είναι μια μέθοδος οπτικής διαχείρισης που προήλθε από τις ιαπωνικές πρακτικές κατασκευής, ειδικά από την Toyota, όπου χρησιμοποιήθηκε για τη βελτίωση της αποδοτικότητας στις διαδικασίες παραγωγής. Ο όρος "Kanban" μεταφράζεται ως "οπτικό σήμα" ή "κάρτα" στα ιαπωνικά.

Στην ουσία του, το Kanban είναι ένα σύστημα που επιτρέπει σε ομάδες ή οργανισμούς να διαχειρίζονται το έργο τους, είτε πρόκειται για κατασκευή, ανάπτυξη λογισμικού, διαχείριση έργων ή οποιοδήποτε άλλο πεδίο, διαθέτοντας μια οπτική

αναπαράσταση της ροής εργασίας και χρησιμοποιώντας οπτικά σήματα για να υποδείξουν την κατάσταση των εργασιών ή αντικειμένων. Η βασική ιδέα πίσω από το Kanban είναι η δημιουργία μιας οπτικής αναπαράστασης της διαδικασίας εργασίας για να ενισχύσει τη διαφάνεια, την επικοινωνία και τη συνεργασία μέσα σε μια ομάδα. (Raut, Laukik & Wakode, Rajat & Talmale, Pravin. (2015). Overview on Kanban Methodology and its Implementation. International Journal for Scientific Research & Development. 03. 2518-2521.)

Τα βασικά συστατικά του συστήματος Kanban είναι:

- I. **Πίνακας Kanban:** Αυτός είναι ένας οπτικός προσανατολισμός της ροής εργασίας, συνήθως χωρισμένος σε στήλες που αντιπροσωπεύουν διάφορα στάδια της εργασίας. Κάθε στήλη αντιπροσωπεύει ένα βήμα στη διαδικασία, όπως "Προς Κατασκευή", "Σε Εξέλιξη" και "Ολοκληρωμένο".
- II. **Κάρτες:** Οι κάρτες αντιπροσωπεύουν μεμονωμένες εργασίες, αντικείμενα ή μονάδες εργασίας. Περιέχουν σχετικές πληροφορίες για την εργασία, όπως η περιγραφή της, η προτεραιότητα και ο Kanban.
- III. **Στήλες:** Οι στήλες στον πίνακα Kanban αντιπροσωπεύουν τα διάφορα στάδια της ροής εργασίας. Οι κάρτες μετακινούνται από αριστερά προς τα δεξιά ανάμεσα στις στήλες καθώς προχωρούν στη διαδικασία.
- IV. **Όρια Εργασίας σε Εξέλιξη (WIP):** Το Kanban τονίζει τον περιορισμό του αριθμού των εργασιών που μπορούν να βρίσκονται σε εξέλιξη ταυτόχρονα. Αυτό βοηθά στην αποφυγή συμφορήσεων και υπερφόρτωσης των μελών της ομάδας, εξασφαλίζοντας μια πιο ομαλή ροή εργασίας.
- V. **Οπτικά Σήματα:** Οπτικές υποδείξεις, συχνά αντιπροσωπευόμενες από κάρτες ή άλλα σήματα, υποδεικνύουν την κατάσταση των εργασιών ή των αντικειμένων. Για παράδειγμα, η μετακίνηση μιας κάρτας από τη στήλη "Προς Κατασκευή" στη στήλη "Σε Εξέλιξη" υποδεικνύει ότι η εργασία έχει αρχίσει.
- VI. **Συνεχής Βελτίωση:** Το Kanban ενθαρρύνει τις ομάδες να αξιολογούν τακτικά τη ροή εργασίας και τις διαδικασίες τους για να εντοπίσουν περιοχές προς βελτίωση. Αυτό περιλαμβάνει την πραγματοποίηση προσαρμογών στη

διαδικασία και τη ροή εργασίας βάσει των πραγματικών δεδομένων απόδοσης.

Το Kanban προσφέρει πολλά οφέλη, συμπεριλαμβανομένης της αυξημένης ορατότητας στο έργο, την ταχύτερη ανταπόκριση στις αλλαγές, τη βελτιωμένη συνεργασία και την εστίαση στην παροχή αξίας στους πελάτες. Είναι γνωστό για την προσαρμοστικότητά του και την ευελιξία του, καθιστώντας το κατάλληλο για διάφορες βιομηχανίες και τύπους έργων. Ενώ το Kanban έχει τις ρίζες του στην κατασκευή, έχει ευρέως υιοθετηθεί στην ανάπτυξη λογισμικού, τη διαχείριση έργων, το μάρκετινγκ και πολλά άλλα πεδία ως ένα ισχυρό εργαλείο για την αποτελεσματική διαχείριση της εργασίας.



Εικόνα 6. Kanban Board Πηγή: Wikipedia.org

4.3 Extreme Programming

Το μοντέλο του ακραίου προγραμματισμού δημιουργήθηκε από τους Kent Beck, Ward Cunningham και άλλους το 1996 αλλά παρουσιάστηκε πρώτη φορά επίσημα το 1999 ως λύση στα προβλήματα που παρουσίαζαν οι παραδοσιακές μεθόδους. Αν και δεν χρησιμοποιείται τόσο πολύ πια αφού είναι κάτω του 10% (16th Annual state of agile report 2022) είναι από τις πιο γνωστές μεθοδολογίες στον χώρο της ανάπτυξης λογισμικού.

Ο στόχος του είναι να παράγει μεγάλης ποιότητας λογισμικό και να πετύχει υψηλότερη ποιότητας ζωής στην ομάδα ανάπτυξης όσο δουλεύει.

Είναι εφαρμόσιμο όταν υπάρχουν συνεχείς δυναμικές αλλαγές στα προαπαιτούμενα του λογισμικού, όταν έχουμε ρίσκα από έργα σταθερού χρόνου που χρησιμοποιούν νέες τεχνολογίες, απο ομάδες μικρές που συστεγάζονται στον ίδιο χώρο και όταν η τεχνολογία που χρησιμοποιεί μια ομάδα επιτρέπει αυτοματισμούς και λειτουργικές δοκιμές. (Beck, K. (1999). Extreme Programming explained: Embrace change.)

4.3.1 Οι αξίες του Extreme Programming

Το μοντέλο Extreme programming βασίζεται σε 5 αξίες:

I. Επικοινωνία

Δίνεται μεγάλη έμφαση στην επικοινωνία σε αυτό το μοντέλο μιας και η ανάπτυξη λογισμικού είναι ομαδική υπόθεση. Τονίζει την μεταφορά γνώσης με την βοήθεια ενός πίνακα ή κάποιου εργαλείου σχεδιασμού.

II. Απλότητα

Ακολουθεί τον κανόνα της απλότητας οπου σημαίνει ότι προσπαθούν πάντα να κάνουν το πιο απλό που θα λειτουργήσει, να αποφύγουν την σπατάλη ώστε να γίνουν μόνο τα απαραίτητα για να είναι και πιο εύκολη η συντήρηση του και να μην υπάρχει προσπάθεια πρόβλεψης του τι θα χρειαστεί στο μέλλον να γίνεται μόνο ότι ακριβώς έχει ζητηθεί από την ομάδα.

III. Ανατροφοδότηση

Με εργαλείο την ανατροφοδότηση υπάρχει συνεχής βελτίωση απο τις προηγούμενες προσπάθειες που έχει κάνει η ομάδα και αναθεωρούν τους τομείς που έχουν κάνει λάθη ώστε να προσαρμοστεί στο μέλλον το προιον τους.

IV. Θάρρος

Ο δημιουργός του Extreme Programming Kent με την λέξη θάρρος ήθελε να δείξει ότι πρέπει να δρούμε ενάντια στον φόβο που μπορεί να έχουμε για να σταματήσουμε κάτι που κάνουμε μέχρι τώρα και μπορεί να είναι επιβλαβές στην ομάδα ή κάτι που είναι δύσκολο να το αποδεχτούμε για να προχωρήσουμε πιο γρήγορα. Μερικές φορές μέσα από το θάρρος για αλλαγή βγαίνει και η απλότητα που χρειάζεται γιατί έχουμε

την τάση πολλές φορές να κάνουμε τα πάντα πιο περίπλοκα με αποτέλεσμα να γίνονται πιο επιβλαβή στην ομάδα.

V. Σεβασμός

Βασικό συστατικό για να πετύχουν όλα είναι ο σεβασμός ανάμεσα στα μέλη της ομάδας δεν γίνεται αλλιώς να υπάρξει σωστή επικοινωνία, ανατροφοδότηση και να βρεθούν απλές λύσεις στα προβλήματα.

4.3.2 Πρακτικές του Extreme Programming

Ακολουθούν οι 12 αρχικές πρακτικές σύμφωνα με την πρώτη έκδοση του οδηγού του Extreme programming απο τον Kent.

- The Planning Game
- Small Releases
- Metaphor
- Simple Design
- Testing
- Refactoring
- Pair Programming
- Collective Ownership
- Continuous Integration
- 40-hour week
- On-site Customer
- Coding Standard

Θα αναλύσουμε λίγο τις πιο δημοφιλείς πρακτικές που μερικές από αυτές αναπτύχθηκαν μετά τον οδηγό και χρησιμοποιούνται πια όπως αναφέρεται στην 2^η έκδοση του οδηγού του Kent. (Beck, K., & Andres, C. (2004). Extreme Programming explained: Embrace Change (2nd Edition).

I. Καθίστε Μαζί

Μια πρακτική που εστιάζει στην επικοινωνία και βασίζεται στο ότι η ομάδα θα βρίσκεται σε έναν ανοιχτό χώρο όλοι μαζί για να υπάρχει η καλύτερη μορφή επικοινωνίας που είναι πρόσωπο με πρόσωπο.

II. Όλη η ομάδα

Μια ομάδα που υπάρχουν όλοι οι ρόλοι που χρειάζονται για να γίνει το έργο και δουλεύουν καθημερινά με σκοπό την επιτυχία του.

III. Ενημερωτικός χώρος εργασίας

Χώρος εργασίας όπου βοηθάει πολύ την επικοινωνία πρόσωπο με πρόσωπο αλλά υπάρχει και ο απαραίτητος ιδιωτικός χώρος για τον καθένα ώστε να εργαστεί.

Προωθεί πολύ την διαφάνεια ως προς την δουλειά της ομάδας σε όλα τα ενδιαφερόμενα μέρη ακόμα και αυτά που δεν είναι στην ομάδα.

IV. Δουλειά με ενέργεια

Επικεντρώνεται στο καταφέρνει να δουλεύουν τα άτομα στην ομάδα χωρίς περισπασμούς και να πετυχαίνουν όσο υψηλότερο βαθμό συγκέντρωσης γίνεται.

Αυτό σημαίνει να είσαι σωματικά ξεκούραστος και διανοητικά ώστε να μπορεί να γίνει.

V. Προγραμματισμός ζευγών (Pair Programming)

Είναι μια απο τις πιο δημοφιλής πρακτικές, σε αυτή ο κώδικας αναπτύσσεται από 2 άτομα που κάθονται στον ίδιο υπολογιστή με την λογική ότι 4 μάτια και 2 μυαλά είναι πιο αποτελεσματικά στο τέλος διότι υπάρχει συνεχής αναθεώρηση και λύνονται πολύ πιο γρήγορα προβλήματα που μπορεί να βασανίζουν ένα άτομο και να χρειάζεται να σταματήσει την παραγωγή του. Οι έρευνες έδειξαν ότι δεν χρειάζονται διπλάσιο χρόνο αφού είναι ικανοί να λύσουν τα προβλήματα πολύ πιο γρήγορα και γλιτώνουν χρόνο απο την αναθεώρηση που γίνεται κατευθείαν εκείνη την στιγμή από το άτομο που δεν γράφει.

VI. Εβδομαδιαίος κύκλος

Στον εβδομαδιαίο κύκλο η ομάδα συναντάται κάθε πρώτη μέρα της εβδομάδας για να μιλήσει για την πρόοδο που έγινε και για να τους ενημερώσει ο πελάτης τι θέλει να γίνει μέχρι την επόμενη εβδομάδα με σκοπό μέχρι τέλος της εβδομάδας να βγουν αυτά που έχει ζητήσει λειτουργικά και δοκιμασμένα.

VII. Χαλαρότητα

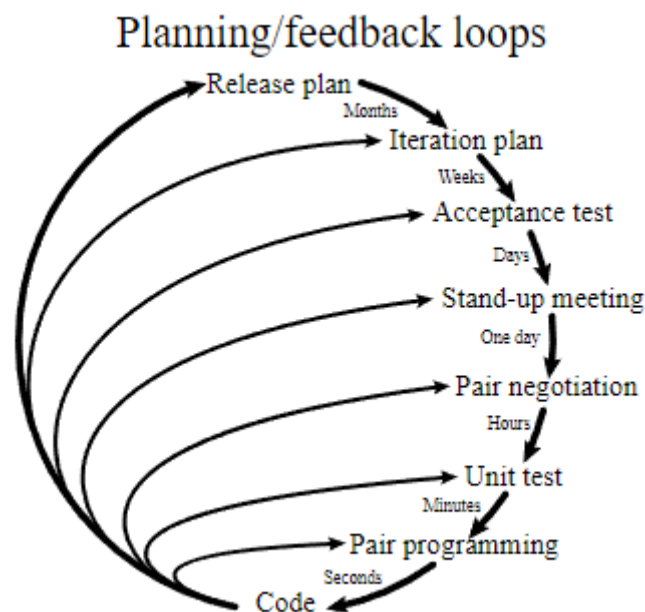
Η λογική πίσω από την χαλαρότητα είναι να προστίθενται σε κάθε κύκλο εργασιών κάποιες εργασίες χαμηλής προτεραιότητας ώστε να διαγράφονται όταν η ομάδα καθυστερεί στα πιο βασικά με αποτέλεσμα να μπορείς να έχεις μια μεταβλητότητα στις προβλεψεις σου για να είσαι πάντα σε θέση να τις εκπληρώσεις.

VIII. Συνεχής ενσωμάτωση

Η συνεχής ενσωμάτωση στοχεύει στο να δοκιμάζονται αμέσως οι αλλαγές κώδικα στην μεγαλύτερη βάση κώδικα που πρόκειται να καταλήξει μετά διότι μπορούν να εντοπιστούν νωρίτερα προβλήματα και να διορθωθούν. Οι πιο πολλές ομάδες φοβούνται την ενοποίηση με την μεγάλη βάση λόγω των πολλών συγκρούσεων που μπορεί να εμφανιστούν και το καθυστερούν συνεχώς. Η σκέψη πίσω από αυτή την πρακτική είναι ότι όσο πιο νωρίς γίνεται θα έχεις να αντιμετωπίσεις λιγότερα προβλήματα με αποτέλεσμα κάθε αλλαγή να είναι πιο εύκολη να γίνει.

IX. Επαυξητικός σχεδιασμός

Προτείνει να γίνεται λίγη δουλειά στην αρχή ώστε να μπορεί να βοηθήσει στο να γίνει καλύτερα κατανοητή ή ευρεία προοπτική του έργου ώστε να υπάρχουν λιγότερες διορθώσεις αργότερα. Έτσι μειώνεται το κόστος αλλαγών και επιτρέπετε να παρθούν αποφάσεις σε πολλά σημεία του έργου με περισσότερες πληροφορίες.



Εικόνα 7 Planning/Feedback XP loops Πηγή: wikimedia.org

4.3.3 Οι ρόλοι του Extreme Programming

Η Μεθοδος Extreme Programming δεν καθορίζει επίσημα κάπου ρόλους ωστόσο θα παρουσιάσουμε παρακάτω τους 4 πιο συνηθισμένους ρόλους που υπάρχουν συνήθως στις ομάδες που την χρησιμοποιούν:

I. Ο πελάτης

Είναι ο υπευθυνος για να πάρει όλες τις επιχειρηματικές αποφάσεις σχετικά με το έργο όπως το τι πρέπει να κάνει το λογισμικό, πως θα καταλάβουμε ότι έχουμε ολοκληρώσει το λογισμικό, πόσα μπορούμε να ξοδέψουμε, τι πρέπει να κάνουμε στην συνέχεια και οποιαδήποτε άλλη απορία προκύψει κατα το έργο. Ο πελάτης συμμετάσχει ενεργα σαν άτομο στο έργο και το ιδανικό είναι να είναι μέλος της ομάδας.

II. Μέλη της ομάδας ανάπτυξης

Όλα τα άτομα που ασχολούνται και δεν είναι ο πελάτης ή κάποιος από τους πιο κάτω ρόλους θεωρείται μέλος της ομάδας ανάπτυξης και είναι υπεύθυνοι για να φτάσουν το λογισμικό στο επίπεδο που επιθυμεί ο πελάτης.

III. Ο ιχνηλάτης

Είναι ένας ρόλος που σε πολλές ομάδες δεν υπάρχει και αν υπάρχει συνήθως είναι κάποιο άτομο από την ομάδα ανάπτυξης που αφιερώνει κάποιον χρόνο στην ημέρα του για να παρακολουθεί μετρήσεις και την ανάπτυξη που έχει η ομάδα κάθε εβδομάδα ώστε να μπορούν να βρεθούν αδυναμίες αν υπάρχουν.

IV. Ο προπονητής

Ο προπονητής συνήθως είναι ένας ρόλος που υπάρχει σε ομάδες που είναι πρώτη φορά που χρησιμοποιούν Extreme Programming ώστε να βοηθήσει ως εξωτερικός σύμβουλος στις πρακτικές του XP λόγω του ότι έχει ξαναπεράσει λάθη που μπορεί να έχουν γίνει στην αρχή και να βοηθήσει στην αποφυγή τους.

IV.4 Lean Software Development

Η Lean Software Development δεν έχει δημιουργηθεί από κάποιον συγκεκριμένο ιδρυτή ή σε κάποια συγκεκριμένη ημερομηνία όπως οι άλλες μεθόδους αλλά βασική επιρροή ήταν η Toyota που πάρθηκαν στοιχεία της απο το πως δούλευε και τα χρησιμοποίησαν στην ανάπτυξη λογισμικού.

Είναι μια μεθοδολογία που βασίζεται στις αρχές του Lean Management όπου ανέπτυξε η Toyota στον τομέα της κατασκευής. Η μεθοδολογία Lean επιδιώκει να βελτιστοποιήσει την αποδοτικότητα, την ποιότητα και παραγωγικότητα στην ανάπτυξη λογισμικού εφαρμόζοντας το Lean Management. Lean σημαίνει να είσαι λιτός και απλός δηλαδή να συγκεντρώνεσαι στον βασικό στόχο και να μην γίνονται πράγματα που δεν προσθέτουν αξία στο προϊόν.

4.4.1 Κεντρικές αρχές της Lean Software Development

I. Αφαίρεση «σκουπιδιών»

Εδώ προσπαθούν να γίνει η αναγνώριση των δραστηριοτήτων και διαδικασιών που δεν είναι απαραίτητες ώστε να αφαιρεθούν. Πολλές φορές παράγοντες για να βρεθούν αυτές είναι όταν έχουν υπερβολική τεκμηρίωση, αναμονές και γενικώς όταν χαρακτηρίζεται από την λέξη υπερβολική.

II. Δημιουργία αξίας

Αυτή η αρχή επικεντρώνεται στο να αναγνωρίσει ποιές δραστηριότητες είναι αυτές που δίνουν αξία στο λογισμικό και έχουν πραγματική σημασία για τους χρήστες μας ώστε να τους δοθεί προτεραιότητα.

III. Δημιουργία γνώσης

Μεγάλη σημασία έχει η ανταλλαγή γνώσης μεταξύ των μελών της ομάδας διότι η συνεργασία και η κοινοποίηση ιδεών βοηθούν στην βελτίωση της κατανόησης των εργασιών που είναι να γίνουν και ενισχύει την απόδοση της ομάδας.

IV. Ενδυνάμωση της ομάδας

Γίνεται ενδυνάμωση της ομάδας μέσω της αυτοοργάνωσης που υπάρχει. Δηλαδή η ομάδα παίρνει τις αποφάσεις για όλα τα ζητήματα και όχι κάποιο ανώτερο στέλεχος. Αυτό ενθαρρύνει την αυτονομία, την ενεργό συμμετοχή και την ευθύνη στα μέλη της ομάδας.

V. Γρήγορη παράδοση

Γίνεται παράδοση μικρών εκδόσεων του λογισμικού πολύ συχνά ώστε να είναι πιο ταχεία η ανταπόκριση σε αλλαγές και με μεγαλύτερη δυνατότητα βελτίωσης βάσει των ανατροφοδοτούμενων πληροφοριών.

Συνοψίζοντας, η Lean Software Development έχει ως σκοπό την δημιουργία λογισμικού υψηλής ποιότητας με την εξάλειψη μη απαραίτητων δραστηριοτήτων, την παροχή πραγματικής αξίας στους χρήστες, την δημιουργία γνώσης στην ομάδα, την ενδυνάμωση της ομάδας και με την συνεχή προσαρμογή στις αλλαγές μέσω των γρήγορων παραδόσεων και γρήγορων ανατροφοδοτήσεων.



Εικόνα 8 Επισκόπηση Lean Πηγή: <https://kruschecompany.com/lean-software-development/>

4.5 Dynamic Systems Development Method (DSDM)

Η Dynamic Systems Development Method δημιουργήθηκε το 1994 από Project Managers που συμμετείχαν οι εταιρείες τους στην κοινοπραξία DSDM με βασικούς αρχηγούς τους Arie van Bennekum, Martin Barnes, Malcolm Evans, Gerry Coleman, Daryl Kulak και Kurt Bittner. Τώρα πια έχει μετονομαστεί σε Agile Business κοινοπραξία. Η DSDM αναπτύχθηκε ως μια από τις πρώτες μεθοδολογίες για την διαχείριση έργων ανάπτυξης λογισμικού με στόχο την βελτίωση της διαδικασίας ανάπτυξης, την αύξηση της συνεργασίας μέσα στην ομάδα και την ταχεία παράδοση με ταυτόχρονα την σωστή ποιότητα του λογισμικού. (Jennifer Stapleton. 1997. Dsdm: The Method in Practice. Addison-Wesley Longman Publishing Co., Inc., USA.)

4.5.1 Βασικές αρχές DSDM

Υπάρχουν 8 βασικές αρχές στην DSDM που πρέπει να ακολουθούνται για να είναι ευέλικτη, συνεργατική και αποδοτική.

I. Συνεργασία και συμμετοχή

Ενθαρρύνεται καθ' όλη την διάρκεια του έργου να υπάρχει επικοινωνία αλλά και συμμετοχή από όλους τους ενδιαφερόμενους (Stakeholders) ακόμα και αυτών που δεν αναπτύσσουν το έργο.

II. Ενδυνάμωση και ενεργοποίηση

Οι ομάδες πάντα παροτρύνονται να λαμβάνουν μόνοι τους αποφάσεις και να αναλαμβάνουν τις ευθύνες. Γενικώς σε όλα τα επίπεδα υπάρχει υποστήριξη για την λήψη αποφάσεων.

III. Στόχος στο να παράγει αξία για την επιχείρηση

Το λογισμικό πρέπει συνέχεια να παράγει αξία για την επιχείρηση. Οπότε όλη η καθοδήγηση του έργου πρέπει να γίνεται ευθυγραμμισμένα με τις πραγματικές ανάγκες της επιχείρησης.

IV. Παράδοση στην ώρα που πρέπει

Πρέπει να τηρείται αυστηρά στο να γίνεται η παράδοση των αποτελεσμάτων εντός των περιθωρίων που έχουν τεθεί χρονικά.

V. Ποτέ δεν θυσιάζουμε ποιότητα

Δεν γίνεται ποτέ αποδεκτή η μείωση της ποιότητας αντιθέτως πρέπει συνέχεια να διατηρείται και να βελτιώνεται.

VI. Χτίζοντας παροδικά πάνω σε γερές βάσεις

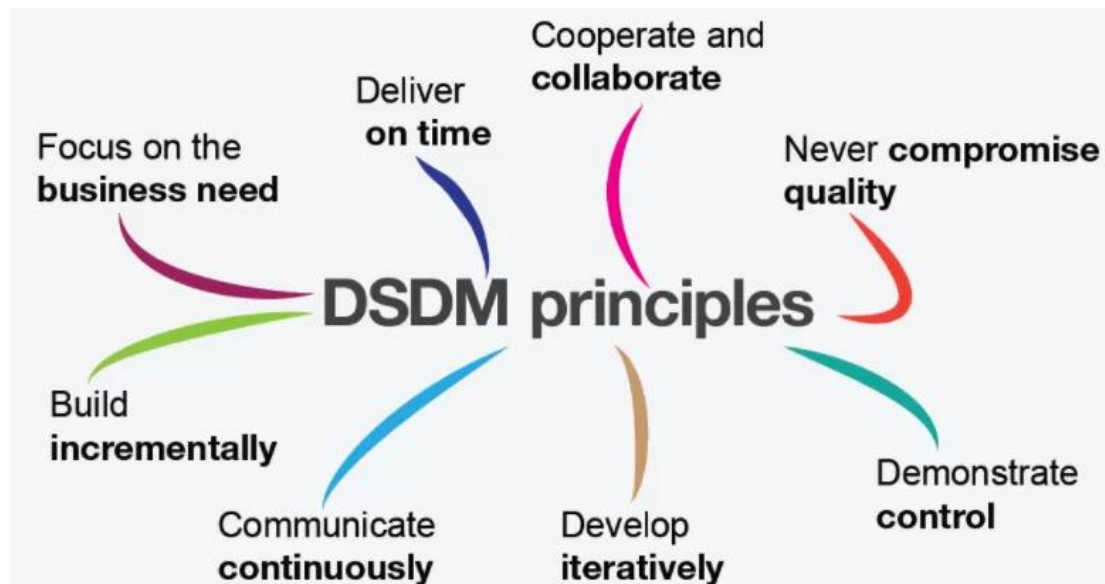
Η ανάπτυξη γίνεται πάντα πάνω σε σταθερές γερές βάσεις και η ομάδα προσθέτει εκεί κάθε μεταβολή.

VII. Αναπτύσσουμε επαναλαμβανόμενα

Βοηθάει στην εξέλιξη και την ευκολότερη ενσωμάτωση των μεταβολών στην βάση.

VIII. Συνεχής επικοινωνία και καθαρή

Για να πετύχει η DSDM χρειάζεται συνεχής και σαφής επικοινωνίας ώστε να επιτευχθεί συνεργασία και κατανόηση ανάμεσα στα μέλη της ομάδας.



Εικόνα 9 Οι αρχές της DSDM Πηγή: <https://thecompetenza.com/agile-development-methodology/>

4.5.2 Αξίες που διέπουν την DSDM

Οι αξίες της DSDM αντιπροσωπεύουν τις βασικές πτυχές που καθοδηγούν την προσέγγιση της και προβάλλουν την σημασία τους στην διαδικασία ανάπτυξης λογισμικού.

I. Επικοινωνία

Η συνεχής και διαφανείς επικοινωνία μεταξύ όλων των ενδιαφερομένων του έργου είναι απαραίτητη για την σωστή κατανόηση των αναγκών και απαιτήσεων το έργου.

II. Συνεργασία

Η ενεργό συμμετοχή όλων των ενδιαφερόμενων μερών και της ομάδας είναι κρίσιμη για την επιτυχία του έργου όπως και η συνεργασία ανάμεσα όλων για να επιτευχθούν αποτελέσματα.

III. Υπευθυνότητα

Οι ομάδες ανάπτυξης πρέπει να αναλαμβάνουν ευθύνες και να είναι σε θέση να λαμβάνουν αποφάσεις για το έργο.

IV. Ανατροφοδότηση

Η συνεχής ανάπτυξη με την συνεχή ανατροφοδότηση προάγουν την εξέλιξη του λογισμικού για να φτάσουμε στις πραγματικές ανάγκες του έργου.

V. Διαχείριση πόρων και χρόνου

Εννοείται ένα από τα σημαντικότερα για να φτάσουμε στην επιτυχία ενός έργου είναι η σωστή και αποτελεσματική διαχείριση των πόρων και του χρόνου που έχουμε στην διάθεση μας.

4.5.3 Πρακτικές που χρησιμοποιούνται στην DSDM

Η DSDM χρησιμοποιεί μια σειρά από πρακτικές και τεχνικές που οδηγούν στην ευέλικτη και αποδοτική ανάπτυξη λογισμικού. Μερικές από αυτές είναι:

I. MoSCoW προτεραιοποίηση

Αυτή η πρακτική βασίζεται στις κατηγορίες “Must Have”, “Should Have”, “Could Have” και “Won’t Have”. Έτσι βοηθάει στο να καταλαβαίνει η ομάδα την προτεραιότητα και σημαντικότητα των απαιτήσεων στο έργο.

II. Timeboxing

Είναι η τεχνική του να δίνεται ένα περιορισμένο διάστημα για κάθε ολοκλήρωση συγκεκριμένης εργασίας με αποτέλεσμα να διασφαλίζεται ότι θα παραδοθεί στο συγκεκριμένο πλαίσιο χρόνου.

III. Επαναληπτική ανάπτυξη

Όπως και σχεδόν σε όλες τις μεθοδολογίες που μιλήσαμε ως τώρα υπάρχει και εδώ η επαναληπτική ανάπτυξη όπου όλη η ανάπτυξη γίνεται σε επαναλαμβανόμενα χρονικά πλαίσια όπου στο τέλος της καθε μιας προσθέτει αξία στο έργο.

IV. Πρωτότυπο

Η δημιουργία αρχικών προτύπων του λογισμικού προκειμένου να γίνει καλύτερη διατύπωση των απαιτήσεων και να συλλέξει σχόλια από τους χρήστες

V. Δοκιμές καθ’όλη την διάρκεια

Να εκτελούν δοκιμές καθ’όλη την διάρκεια του έργου ώστε να διασφαλίζεται η ποιότητα και η λειτουργικότητα του λογισμικού.

VI. Συνεργατικά εργαστήρια

Η διεξαγωγή συνεργατικών εργαστηρίων με τη συμμετοχή των ενδιαφερομένων προκειμένου να ανταλλαχθούν ιδέες, να διευκρινιστούν απαιτήσεις και να λυθούν προβλήματα.

VII. Διευκολυνόμενα εργαστήρια

Η επιμέλεια και διευκόλυνση των εργαστηρίων από έναν εμπειρογνώμονα προκειμένου να διασφαλιστεί η αποτελεσματική διαδικασία.

VIII. Μοντελοποίηση και πρωτοτυποποίηση

Η χρήση μοντελοποίησης και δημιουργίας πρωτοτύπων για την καλύτερη κατανόηση των απαιτήσεων και του σχεδιασμού.

4.5.4 Οι φάσεις του DSDM

Οι φάσεις της Dynamic Systems Development Method (DSDM) αντικατοπτρίζουν τη δομή της ανάπτυξης λογισμικού με τη χρήση αυτής της μεθοδολογίας. Οι βασικές φάσεις της DSDM είναι:

I. Σκοπιμότητα

Σε αυτή την φάση βλέπουμε την εφικτότητα του έργου μας και με μια αρχική ανάλυση των απαιτήσεων και των στόχων προσδιορίζουμε τις λύσεις και την σκοπιμότητα του.

II. Επιχειρηματική έρευνα

Εδώ γίνεται περαιτέρω ανάλυση στα παραπάνω και δημιουργείται το επιχειρησιακό μοντέλο ώστε να υπάρξει αξία από το έργο.

III. Επανάληψη λειτουργικού μοντέλου

Κατά τη διάρκεια αυτής της φάσης γίνεται η επιλογή των πιο κρίσιμων λειτουργιών και δημιουργείται ένα επαναλαμβανόμενο λειτουργικό μοντέλο για το σύστημα.

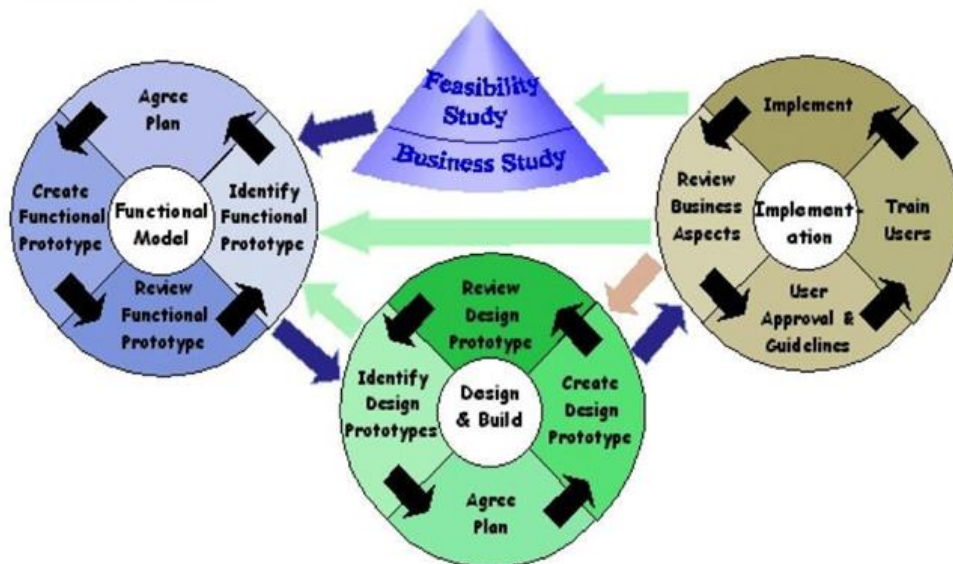
IV. Επανάληψη σχεδιασμού και κατασκευής

Εδώ γίνεται ο σχεδιασμός και η κατασκευή του έργου σε επανάληψη μέχρι να βρεθεί το κατάλληλο όπως επίσης εδώ δημιουργούνται τα πρώτα πρωτότυπα.

V. Εκτέλεση

Σε αυτή την φάση εκτελούνται όλα όσα έχουν αποφασιστεί στις προηγούμενες φάσεις.

The DSDM Development Process



Εικόνα 10 Οι πρακτικές της DSDM Πηγή: https://www.researchgate.net/figure/521-Dynamic-Systems-Development-Methodology-DSDM-Source_fig13_344458644

4.5.5 Οι Ρόλοι στο DSDM

Η Dynamic Systems Development Method (DSDM) καθορίζει διάφορους ρόλους που συμμετέχουν στη διαδικασία ανάπτυξης λογισμικού. Κάθε ρόλος έχει συγκεκριμένες ευθύνες και καθήκοντα που συντελούν στην επιτυχή υλοποίηση του έργου.

Ορισμένοι από τους κύριους ρόλους που συναντάμε στην DSDM είναι:

I. Εκπρόσωπος των χρηστών

Είναι αυτός που πρέπει να καθοδηγήσει την ομάδα της ανάπτυξης ώστε να ικανοποιούνται οι ανάγκες και απαιτήσεις των χρηστών.

II. Εκπρόσωπος της επιχείρησης

Αντιπροσωπεύει τα συμφέροντα της επιχείρησης και εξασφαλίζει το έργο να ευθυγραμμίζεται και με τους στόχους της ώστε να υπάρχει αξία για αυτήν.

III. Οραματιστής της επιχείρησης

Ο οραματιστής είναι αυτός που καθορίζει το όραμα το επιχειρησιακό πλαίσιο και τους στόχους του έργου.

IV. Προγραμματιστής

Οι προγραμματιστές είναι υπεύθυνοι για την εκτέλεση των δραστηριοτήτων που πρέπει να γίνουν για να γίνει η ανάπτυξη του λογισμικού και για την πραγματοποίηση των αλλαγών που θέλουν οι άλλοι ρόλοι του έργου.

V. Διαχειριστής του περιβάλλοντος του έργου

Αναλαμβάνει την διασφάλιση του αναγκαίου περιβάλλοντος και εργαλείων που χρειάζεται η ομάδα για να γίνει η ανάπτυξη και ο έλεγχος του έργου.

VI. Αρχηγός της ομάδας εργασίας

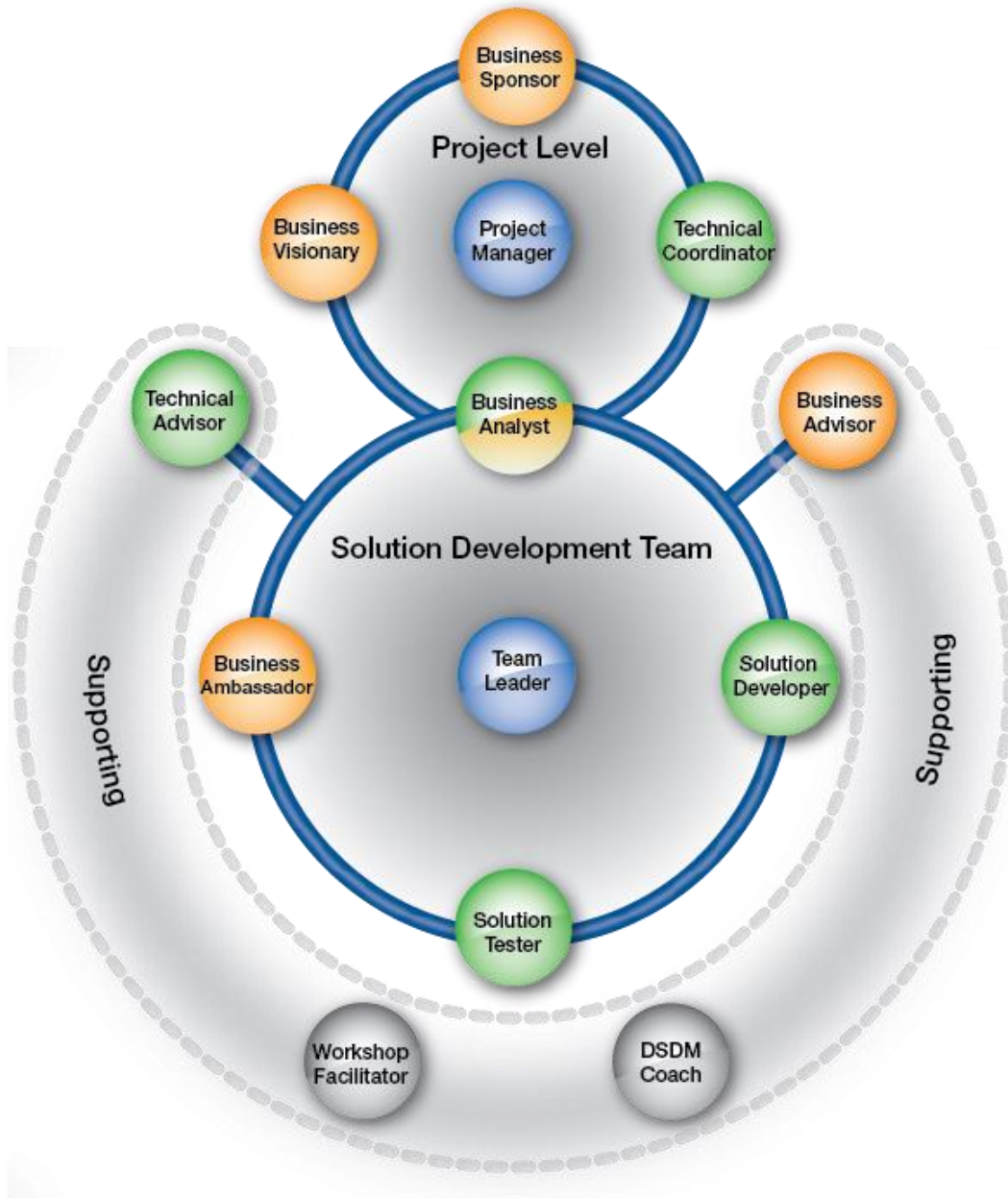
Ο αρχηγός οργανώνει και κατευθύνει την ομάδα ώστε να μπορέσει να προχωρά το έργο σύμφωνα με το χρονοδιάγραμμα και τις απαιτήσεις.

VII. Δοκιμαστής

Υπεύθυνος για να κάνει τις δοκιμές όταν πρέπει και να εξασφαλίσει την ποιότητα που χρειάζεται το έργο μέσω της ανατροφοδότησης που θα δίνει στην ομάδα.

VIII. Συντονιστής των εργασιών που έχει η ομάδα να κάνει

Συντονίζει και διαχειρίζεται τις προσπάθειες της ομάδας για να γίνουν όλα στην ώρα τους που πρέπει χωρίς να δημιουργούν εμπόδια για άλλους.



Εικόνα 11 Οι ρόλοι της DSDM Πηγή: <https://www.agilebusiness.org/dsdm-project-framework/roles-and-responsibilities.html>

4.6 Feature-Driven Development (FDD)

Η Feature-Driven Development (FDD) αρχικά ξεκίνησε από τον Ivar Jacobson και τον Peter Coad. Στην αρχή ο Ivar ανέπτυξε το θεωρητικό και μεθοδολογικό κύκλο της FDD ενώ στην συνέχεια ο Peter βοήθησε με την πρακτική υλοποίηση της μεθοδολογίας.

Ξεκίνησε η ανάπτυξη της το 1990 και πιο συγκεκριμένα το 1994 ο Ivar παρουσίασε το βασικό πλαίσιο της στο βιβλίο του “object-Oriented Software Engineering – A Use case Driven Approach”. Στην συνέχεια το 1997 ο Peter και η ομάδα του δημοσίευσε το βιβλίο “Java Modeling in Color with UML” όπου περιέγραφε πιο αναλυτικά την μεθοδολογία FDD και παρουσιάζει και σενάρια εφαρμογής.

Αργότερα ο Peter ανέπτυξε ένα σύνολο πρακτικών και διαδικασιών που συμβάδιζαν με τις αρχές της FDD. Η FDD μετά αποτέλεσαι μία από τις πολλές μεθοδολογίες ανάπτυξης λογισμικού που δημιουργήθηκαν κατα την διάρκεια αυτής της δεκαετίας και έχει χρησιμοποιηθεί με επιτυχία σε πάρα πολλά έργα ανάπτυξης λογισμικού. (Steve R. Palmer and Mac Felsing. 2001. A Practical Guide to Feature-Driven Development (1st. ed.). Pearson Education.)

4.6.1 Οι ρόλοι που υπάρχουν στην FDD

Στην FDD υπάρχουν πολλοί ρόλοι που χρειάζεται να συνεργαστούν για να διεξαχθεί αποτελεσματικά η διαδικασία ανάπτυξης. Αυτοί οι ρόλοι περιλαμβάνουν:

I. Ο Διαχειριστής του έργου

Είναι αυτός που είναι υπεύθυνος για όλο τον συντονισμό των ομάδων και μελών που δραστηριοποιούνται στο έργο. Πρέπει να έχει καταλάβει ακριβώς τις ανάγκες του πελάτη και ανάλογα να διαχειρίζεται το πρόγραμμα, τον προϋπολογισμό και τους πόρους. Επίσης το σημαντικότερο είναι υπεύθυνος για να βγει όλο το έργο στην ώρα που έχει συμφωνηθεί και ακριβώς όπως έχει συμφωνηθεί.

II. Αρχιτέκτονας του έργου

Υπεύθυνος για τον γενικό σχεδιασμό του λογισμικού οπότε είναι υποχρεωτικό να έχει κατανοήσει απολύτως τις απαιτήσεις, την δομή του συστήματος που θέλει ο πελάτης και να επιλέξει τις βέλτιστες λύσεις και τεχνολογίες για να γίνει αυτό.

III. Υπεύθυνος ανάπτυξης

Ο Υπεύθυνος ανάπτυξης επιβλέπει την ομάδα συνεχώς και εξασφαλίζει ότι όλα εκτελούνται όπως τα έχει σχεδιάσει ο αρχιτέκτονας και ο διαχειριστής του έργου ώστε να βγουν όλα στην ώρα που πρέπει. Καθοδηγεί τις διαδικασίες σχεδιασμού και υλοποίησης του κώδικα ενώ ταυτόχρονα είναι αυτός που επιβλέπει για να υπάρχει η σωστή ποιότητα και ότι τηρούν τα πρότυπα κώδικα που έχουν συμφωνηθεί.

IV. Ο Υπεύθυνος των προγραμματιστών

Ο υπεύθυνος της ομάδας των προγραμματιστών είναι αυτός που συντονίζει την ομάδα ώστε έχουν μοιραστεί σωστά οι αρμοδιότητες ανάμεσα στην ομάδα και να μην υπάρχουν εμπόδια λόγω λάθος προγραμματισμού δραστηριοτήτων. Επίσης είναι ο πρώτος που επιβλέπει την ώρα που γράφεται ο κώδικας αν τηρούνται τα πρότυπα.

V. Εξειδικευμένες γνώστης ενός πεδίου

Εδώ ανήκουν τα άτομα που έχουν έρθει στην ομάδα για να προσφέρουν την συμβουλή τους και να βοηθήσουν σε ένα συγκεκριμένο πεδίου που γνωρίζουν πολύ καλά ώστε να διασφαλίσουν ότι το σύστημα καλύπτει όλους τους στόχους σε αυτό το πεδίο.

VI. Υπεύθυνοι εκτέλεσης κάποιον κομματιών

Μέσα στην ομάδα υπάρχουν και κλάσσεις όπου μπορεί να έχουν δοθεί σε συγκεκριμένα άτομα και ομάδες ώστε να τις ολοκληρώσουν και απλά να συγχωνευτούν με το κεντρικό έργο στο τέλος.

VII. Ομάδες χαρακτηριστικών

Μέσα στην ομάδα επίσης μπορεί να υπάρχουν ομάδες όπου είναι ειδικές στην ανάπτυξη συγκεκριμένων χαρακτηριστικών σε τέτοια συστήματα με αποτέλεσμα να δουλεύουν σε όλη την διάρκεια του πάνω σε αυτά και να παρουσιάζονται στο τέλος για συγχώνευση με το έργο.

4.6.2 Οι πρακτικές που χρησιμοποιούνται στην FDD

I. Σχεδιασμός ενός γενικού μοντέλου

Κατα το πρώτο στάδιο δημιουργείται ένα γενικό μοντέλο του συστήματος που πρέπει να φτιαχτεί. Μετά με αυτό το μοντέλο ως βάση γνωρίζουν τις βασικές αλληλεπιδράσεις που χρειάζονται να υπάρχουν και χτίζουν πάνω του.

II. Δημιουργία λίστας χαρακτηριστικών

Δημιουργείται μια λίστα με όλα τα χαρακτηριστικά που πρέπει να έχει το έργο. Το καθένα αντιστοιχεί σε μια συγκεκριμένη λειτουργία και αρχίζει η ομάδα να τα δουλεύει ένα ένα.

III. Προγραμματισμός ανάλογα τα χαρακτηριστικά

Η ομάδα κάνει τον προγραμματισμό της ανάπτυξης του έργου σύμφωνα με τα χαρακτηριστικά που χρειάζονται. Το καθένα υποβάλλεται για ανάλυση στην ομάδα ώστε να προσδιοριστούν οι απαιτήσεις, η σχεδίαση και ο προγραμματισμός του.

IV. Σχεδιασμός ανάλογα τα χαρακτηριστικά

Στην συνέχεια της προηγούμενης πρακτικής πραγματοποιείται ο σχεδιασμός των κλάσεων που θα δουλεύουν και των δομών που πρέπει να γίνουν ώστε να μπορεί να γίνει η ανάπτυξη των χαρακτηριστικών.

V. Υλοποίηση ανάλογα τα χαρακτηριστικά

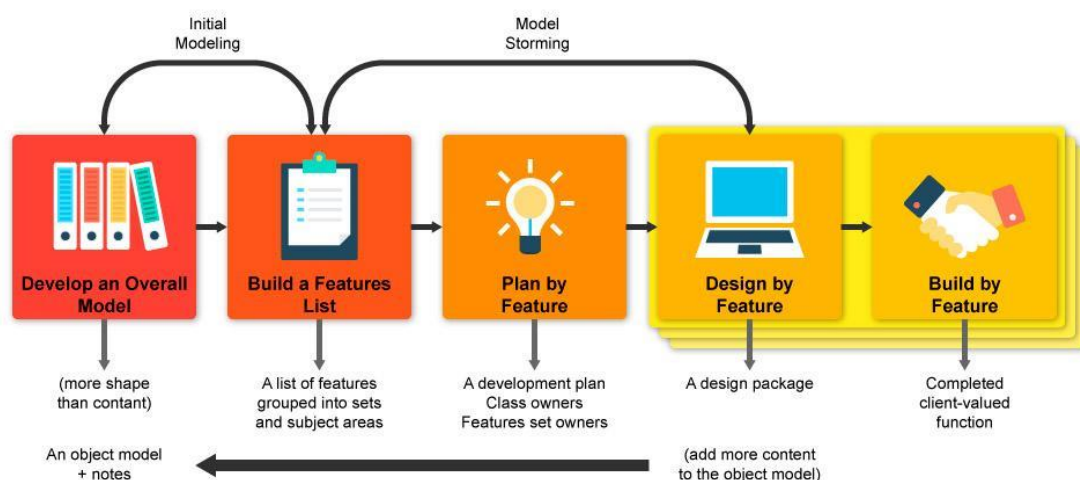
Η κάθε ομάδα αναλαμβάνει μια κλάση να υλοποιήσει και είναι υπεύθυνη για τον προγραμματισμό και την υλοποίηση του συγκεκριμένου χαρακτηριστικού στα χρονικά περιθώρια που πρέπει.

VI. Επαλήθευση ανάλογα τα χαρακτηριστικά

Το κάθε χαρακτηριστικό υποβάλλεται συνεχώς σε δοκιμές για να γίνει απόλυτα σίγουρο ότι λειτουργούν σωστά και όπως το έχουν ζητήσει οι πελάτες.

VII. Συνεχής πρόοδος

Κατά την διάρκεια όλως των παραπάνω παρακολουθείτε η γενική πρόοδος και διασφαλίζεται συνεχώς ότι το έργο είναι στο στάδιο που θα έπρεπε να είναι στην συγκεκριμένη χρονική στιγμή ώστε να τελειώσει στην προθεσμία που έχει ζητηθεί.



Εικόνα 12 Οι πρακτικές της FDD Πηγή: <https://lvivivity.com/7-things-about-feature-driven-development>

4.6.3 Οι αξίες της FDD

Οι αξίες της FDD δείχνουν τις βασικές αρχές και την φιλοσοφία που καθοδηγούν την FDD και την ομάδα ανάπτυξης προς την σωστή ολοκλήρωση των έργων. Οι αξίες είναι:

I. Απλότητα

Μια από τις βασικές αξίες της FDD είναι η απλότητα. Γενικώς η μεθοδολογία ενθαρρύνει την σχεδίαση και υλοποίηση απλών λύσεων όπου αυτό οδηγεί και σε πιο εύκολη συντήρηση και ανάπτυξη του λογισμικού.

II. Ποιότητα

Ενώ η μεθοδολογία προωθεί την απλότητα με τίποτα δεν θυσιάζει την ποιότητα του έργου. Για την FDD είναι ζωτικής σημασίας να υπάρχουν οι βέλτιστες πρακτικές και με τήρηση των υψηλότερων προτύπων κώδικα προκειμένου να διασφαλιστεί η ποιότητα του κώδικα και η σωστή λειτουργικότητα του συστήματος.

III. Συνεχής βελτίωση

Επίσης ένα από τα βασικά στοιχεία της είναι η συνεχή βελτίωση. Η μεθοδολογία επιτρέπει την εξέλιξη του λογισμικού και την προσαρμογή κατά την διάρκεια του έργου βασιζόμενη στην συνεχή ανατροφοδότηση και ανάδραση.

Οι παραπάνω αξίες συμβάλλουν στην δημιουργία ενός περιβάλλοντος που ευνοεί την ανάπτυξη υψηλής ποιότητας λογισμικού, την προσαρμογή σε μεταβαλλόμενες απαιτήσεις και την επίτευξη εξαιρετικών αποτελεσμάτων.

4.6.4 Οι αρχές της FDD

Οι αρχές της FDD καθοδηγούν την μεθοδολογία και την προσέγγιση της στην ανάπτυξη λογισμικού με βάση τα χαρακτηριστικά που υπάρχουν. Οι αρχές αυτές περιλαμβάνουν τις εξής:

I. Επικεντρωμένη στα χαρακτηριστικά

Η FDD έχει ως βασική αρχή όλη η μεθοδολογία να είναι γύρω από τα χαρακτηριστικά που χρειάζονται να φτιαχτούν στο λογισμικό. Το καθένα τους είναι αντιστοιχισμένο με μια λειτουργία ή δυνατότητα που θέλουμε να έχει το λογισμικό.

II. Σύντομοι επαναλαμβανόμενοι κύκλοι φάσεων

Η ανάπτυξη γίνεται μέσα σε σύντομους, επαναλαμβανόμενους κύκλους φάσεων. Κάθε κύκλος αντιστοιχεί σε μια συγκεκριμένη φάση της ανάπτυξης, όπως η ανάλυση, ο σχεδιασμός και η υλοποίηση, και συμβάλλει στην συνεχή πρόοδο του έργου.

III. Ομαδική εργασία

Η FDD βασίζεται σε συνεργασία και ομαδική εργασία. Οι διάφοροι ρόλοι στη μεθοδολογία συνεργάζονται στενά για την επίτευξη των στόχων ανάπτυξης. Οι ομάδες χαρακτηριστικών αναλαμβάνουν την υλοποίηση των χαρακτηριστικών με συνεργασία και συντονισμό.

IV. Επιθεώρηση

Κατά τη διάρκεια των διαφόρων φάσεων της ανάπτυξης, πραγματοποιούνται συστηματικές επιθεωρήσεις για τον έλεγχο της ποιότητας του κώδικα, την τήρηση των προτύπων και την εξασφάλιση της ορθότητας των όσων έχουν αναπτυχθεί μέχρι τότε.

V. Τακτική ανάπτυξη και τακτικές δοκιμές

Έχουμε συνεχή ανάπτυξη του λογισμικού όπως και συνεχή δοκιμές ώστε να βοηθούν στην παρακολούθηση του έργου και να γίνεται η ανίχνευση προβλημάτων στα αρχικά στάδια για να αντιμετωπίζονται πιο εύκολα και γρήγορα.

4.7 Crystal Family Methodologies

Η οικογένεια μεθοδολογιών Crystal δημιουργήθηκε από τον Alistair Cockburn. Αρχικά, το 1992, ο Cockburn δημοσίευσε το βιβλίο "Structured Design" στο οποίο παρουσίασε μία μεθοδολογία για την ανάπτυξη λογισμικού με βάση την δομημένη σχεδίαση.

Αργότερα, ο Cockburn εξέλιξε τις ιδέες του για την ανάπτυξη λογισμικού, ενσωματώνοντας πολλές από τις προηγούμενες αρχές και προσεγγίσεις του, και δημιούργησε την οικογένεια μεθοδολογιών Crystal. Οι πρώτες μορφές της Crystal παρουσιάστηκαν περίπου από το 1995 και μετά. (Agile Software Development (2000) Alistair Cockburn)

Η Crystal είναι μια οικογένεια μεθοδολογιών ανάπτυξης λογισμικού που έχει ως στόχο την ευέλικτη και αποτελεσματική ανάπτυξη λογισμικού, λαμβάνοντας υπόψη τον χαρακτήρα και τις απαιτήσεις κάθε έργου. Η οικογένεια μεθοδολογιών Crystal περιλαμβάνει διάφορες εκδοχές, γνωστές ως "Crystals", καθεμία από τις οποίες αντιπροσωπεύει μια προσέγγιση και ένα σύνολο πρακτικών για την ανάπτυξη λογισμικού. Οι πιο γνωστές μεθοδολογίες της οικογένειας Crystal περιλαμβάνουν:

I. Crystal clear

Είναι μια χαλαρή μεθοδολογία όπου είναι κατάλληλη για μικρές ομάδες έως 8 άτομα και μικρά έργα. Βασίζεται στην επικοινωνία, την συνεργασία, την ευελιξία και την απλότητα.

II. Crystal yellow

Η Crystal yellow στοχεύει σε μεγαλύτερες ομάδες ανάπτυξης με σκοπό να βελτιώσει την επικοινωνία μεταξύ τους και τον συντονισμό τους. Ο σκοπός της είναι όλες οι ομάδες τις να φτάσουν να είναι αυτοοργανωμένες.

III. Crystal orange

Η orange προσφέρεται πιο πολύ για μεγάλες ομάδες όπου χρειάζονται περισσότερο έλεγχο και πιο δομημένη προσέγγιση στα προβλήματα της. Αυξάνει πολύ την βαρύτητα που δίνεται στον συντονισμό της και έχει πιο αυστηρά πρότυπα και διαδικασίες σε σχέση με τις 2 προηγούμενες.

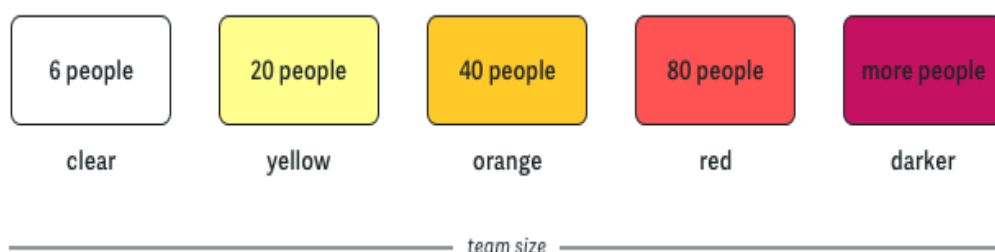
IV. Crystal red

Είναι κατάλληλη για πολύ μεγάλες ομάδες επειδή επικεντρώνεται στην επεξεργασία της πολυπλοκότητας μέσω διαίρεσης του έργου σε πολλές υποομάδες και την δημιουργία μικρότερων ομάδων όπου είναι η καθεμία υπεύθυνη για μια άλλη πτυχή του έργου.

V. Crystal maroon

Η πιο πολύπλοκη έκδοση της οικογένειας Crystal, κατάλληλη για πολύ μεγάλα έργα που απαιτούν υψηλό βαθμό δομημένης προσέγγισης και διαδικασιών.

Κάθε μια από αυτές τις μεθοδολογίες έχει τα δικά της χαρακτηριστικά, προσεγγίσεις και πρακτικές, προσαρμοσμένα στις συγκεκριμένες ανάγκες του έργου και την κουλτούρα της ομάδας ανάπτυξης.



Εικόνα 13 Κατανομή μεγέθους ομάδας με χρώμα Crystal. Πηγή: <https://activecollab.com/blog/project-management/crystal-methods>

4.7.1 Οι ρόλοι της Crystal

Η μεθοδολογία Crystal διαθέτει ένα σύνολο ρόλων που αναλαμβάνουν διάφορες ευθύνες κατά τη διάρκεια της ανάπτυξης λογισμικού. Οι βασικοί ρόλοι που συναντώνται στην οικογένεια μεθοδολογιών Crystal είναι οι εξής:

I. Σύμβουλος

Ένα άτομο εκτός της ομάδας όπου παρέχει γνώσεις, συμβουλές και καθοδήγηση γενικώς για την εκτέλεση του έργου. Δεν χρειάζεται να υπάρχει καθ'όλη την διάρκεια του αλλά μπορεί να συμμετέχει οπότε χρειάζεται.

II. Συντονιστής

Ο συντονιστής είναι υπεύθυνος για την οργάνωση της ομάδας ανάπτυξης. Καθοδηγεί την ομάδα, διαχειρίζεται τις εργασίες που έχει να κάνει η ομάδα και διασφαλίζει την συνεχή πρόοδο μέσω σωστού συντονισμού των ενεργειών που πρέπει να γίνουν.

III. Παραγωγός

Ο παραγωγός είναι υπεύθυνος σε καθημερινή βάση να ελέγχει ότι η λειτουργία της ομάδας είναι στα επίπεδα που πρέπει και εξασφαλίζει την ομαλή λειτουργία της μέσω επίλυσης των προβλημάτων που μπορεί να μειώσουν την παραγωγικότητα τους.

IV. Εκπρόσωπος χρηστών

Εκπροσωπεί τους χρήστες του λογισμικού και διασφαλίζει ότι οι ανάγκες τους θα ληφθούν υπόψη και θα καλυφθούν.

V. Χρήστες

Οι χρήστες είναι οι τελικοί χρήστες που θα χρησιμοποιήσουν το λογισμικό οπότε συμμετέχουν στο έργο παρέχοντας συνεχή ανατροφοδότηση αξιολογώντας το λογισμικό που φτάνει στα χέρια τους.

Αυτοί οι ρόλοι αντανakλούν την έμφαση της μεθοδολογίας Crystal στη συνεργασία, την ευελιξία και την ανθρώπινη πτυχή της ανάπτυξης λογισμικού.

4.7.2 Οι πρακτικές της Crystal

Η μεθοδολογία Crystal υποστηρίζει τη χρήση διαφόρων πρακτικών που επικεντρώνονται στην ευελιξία, τη συνεργασία και την ποιότητα. Ανάλογα με τη συγκεκριμένη έκδοση της οικογένειας Crystal, οι πρακτικές αυτές μπορεί να διαφοροποιούνται, αλλά γενικά συμπεριλαμβάνουν:

I. Κοινή Κατανόηση

Η ομάδα πρέπει να έχει κοινή κατανόηση των στόχων, των απαιτήσεων και των προκλήσεων του έργου.

II. Συνεργασία

Η συνεργασία μεταξύ των μελών της ομάδας είναι κρίσιμη. Οι μέθοδοι ενθαρρύνουν την ανοικτή επικοινωνία, την ανταλλαγή ιδεών και τη συνεργατική επίλυση προβλημάτων.

III. Διασπασμένη εργασία

Το έργο διασπάται σε μικρές και πιο διαχειρίσιμες εργασίες. Αυτό βοηθά στην καλύτερη παρακολούθηση της προόδου και τη διασφάλιση της αποδοτικότητας.

IV. Αυτοοργάνωση

Οι ομάδες ανάπτυξης αναλαμβάνουν την αυτοοργάνωση των δραστηριοτήτων τους. Κάθε μέλος έχει τη δυνατότητα να αναλάβει δράση και να συμβάλλει στο έργο.

V. Διαρκής παρακολούθηση

Η πρόοδος και η ποιότητα του έργου παρακολουθούνται συνεχώς, με στόχο τη βελτίωση και την αντιμετώπιση προβλημάτων άμεσα.

VI. Κύκλοι ανάπτυξης

Η ανάπτυξη γίνεται μέσω συνεχών και σύντομων κύκλων ανάπτυξης, γνωστών και ως iterations ή increments. Κάθε κύκλος παράγει ένα λειτουργικό κομμάτι του λογισμικού.

VII. Τεστ και ποιότητα

Η δοκιμή και η επιδίωξη της υψηλής ποιότητας αποτελούν βασικό μέρος της ανάπτυξης. Η διασφάλιση ότι το λογισμικό λειτουργεί σωστά είναι κρίσιμη.

VIII. Αξιολόγηση και προσαρμογή

Μετά από κάθε κύκλο ανάπτυξης, γίνεται αξιολόγηση των αποτελεσμάτων και προσαρμογή της στρατηγικής ή των πρακτικών, αν απαιτείται.

Αυτές είναι μερικές από τις κύριες πρακτικές που εφαρμόζονται στη μεθοδολογία Crystal, ανάλογα με τη συγκεκριμένη έκδοση που χρησιμοποιείται.

4.7.3 Οι αξίες της Crystal

Οι αξίες της μεθοδολογίας Crystal επικεντρώνονται σε θεμελιώδεις αρχές που διαμορφώνουν τον τρόπο που προσεγγίζεται η ανάπτυξη λογισμικού. Οι κύριες αξίες της μεθοδολογίας Crystal περιλαμβάνουν:

I. Συνεργασία

Η Crystal επικεντρώνεται στη συνεργασία μεταξύ των μελών της ομάδας ανάπτυξης, καθώς και με τους τελικούς χρήστες. Η ανοικτή επικοινωνία, η ανταλλαγή ιδεών και η συνεργατική προσέγγιση συμβάλλουν στην ανάπτυξη υψηλής ποιότητας λογισμικού.

II. Ευελιξία

Η Crystal προωθεί την ευελιξία στην ανάπτυξη λογισμικού. Αντιλαμβάνεται ότι κάθε έργο έχει μοναδικές απαιτήσεις και συνθήκες, και προσαρμόζει τη μεθοδολογία ανάλογα.

III. Επαγγελματισμός

Η μεθοδολογία ενθαρρύνει την προσήλωση στην ποιότητα, την τήρηση προθεσμιών και την παροχή του βέλτιστου αποτελέσματος για τον πελάτη.

IV. Ανθρωποκεντρικότητα

Η Crystal αναγνωρίζει τη σημασία του ανθρώπινου παράγοντα στην ανάπτυξη λογισμικού. Κατανοεί ότι η ευημερία των μελών της ομάδας συνδέεται με την επιτυχία του έργου.

V. Σταδιακή ενσωμάτωση

Η Crystal προωθεί την ενσωμάτωση του λογισμικού σε μικρά και συχνά διαστήματα. Αυτό επιτρέπει στους χρήστες να δουν πρόοδο και να παρέχουν ανατροφοδότηση πιο συχνά.

Αυτές οι αξίες διαμορφώνουν την προσέγγιση της μεθοδολογίας Crystal και συμβάλλουν στη δημιουργία ενός περιβάλλοντος ανάπτυξης λογισμικού που επιδιώκει την ευελιξία, τη συνεργασία και την ανθρωποκεντρικότητα.

4.7.4 Οι αρχές της Crystal

Οι αρχές της μεθοδολογίας Crystal αντικατοπτρίζουν τις βασικές ιδέες και προσεγγίσεις που καθοδηγούν την ανάπτυξη λογισμικού. Αν και αυτές οι αρχές μπορεί να διαφέρουν μεταξύ των διαφόρων εκδοχών της μεθοδολογίας Crystal, εδώ είναι μερικές από τις κύριες αρχές που συχνά συνδέονται με αυτήν:

I. Συνεργασία με τον πελάτη

Οι ανάγκες του πελάτη είναι κρίσιμες για την επιτυχία του έργου. Η συνεργασία με τον πελάτη κατά τη διάρκεια της ανάπτυξης είναι απαραίτητη για τη διασφάλιση ότι το τελικό προϊόν θα ανταποκρίνεται στις ανάγκες του.

II. Αυτοοργάνωση της ομάδας

Η ομάδα ανάπτυξης έχει την αυτονομία να αποφασίζει για τον τρόπο με τον οποίο θα εκτελεστεί το έργο. Οι μελλοντικές αποφάσεις λαμβάνονται από την ομάδα.

III. Ανάπτυξη με σταδιακή ενσωμάτωση

Η ανάπτυξη του λογισμικού γίνεται με βάση την σταδιακή ενσωμάτωση σε μικρά και συχνά διαστήματα. Αυτό επιτρέπει την ανάπτυξη λειτουργικού λογισμικού που

μπορεί να χρησιμοποιηθεί από τους χρήστες με λιγότερα προβλήματα κατά την διάρκεια της ανάπτυξης του.

IV. Εστίαση στην Ποιότητα

Η ποιότητα του λογισμικού αποτελεί κεντρική ανησυχία. Οι μεθοδολογίες Crystal προωθούν την εφαρμογή τεχνικών δοκιμής και παρακολούθησης για τη διασφάλιση της ποιότητας.

V. Διασπασμένη Εργασία

Το έργο διασπάται σε μικρές εργασίες που μπορούν να υλοποιηθούν ανεξάρτητα. Αυτή η αρχή επιτρέπει την ανάπτυξη λογισμικού με ευελιξία και αυξάνει την επιτυχία της παράδοσης.

VI. Επαναλαμβανόμενοι κύκλοι ανάπτυξης

Οι κύκλοι ανάπτυξης είναι επαναλαμβανόμενοι και συχνοί. Μετά από κάθε κύκλο, αξιολογείται η απόδοση και γίνονται βελτιώσεις για την επόμενη φάση.

Αυτές οι αρχές αντικατοπτρίζουν τη φιλοσοφία της Crystal και τον τρόπο που προσεγγίζει την ανάπτυξη λογισμικού, βασισμένη σε συνεργασία, ευελιξία και ποιότητα.

4.8 Adaptive Software Development (ASD)

Η μεθοδολογία Adaptive Software Development (ASD) δημιουργήθηκε από τον Jim Highsmith, έναν κορυφαίο ειδικό στον τομέα της ανάπτυξης λογισμικού και της διαχείρισης έργων. Η ASD αναπτύχθηκε στη δεκαετία του 1990 και ανακοινώθηκε αρχικά στη δημοσίευση του Jim Highsmith με τίτλο "Adaptive Software Development: A Collaborative Approach to Managing Complex Systems" το 2000. Στη συνέχεια, η μεθοδολογία ASD εξελίχθηκε και ενσωματώθηκε στην οικογένεια των μεθοδολογιών Agile, που επικεντρώνονται σε ευελιξία, συνεργασία και προσαρμοστικότητα στην ανάπτυξη λογισμικού. (Adaptive Software Development (1999) James A. Highsmith)

4.8.1 Οι πρακτικές της ASD

Οι πρακτικές της μεθοδολογίας Adaptive Software Development (ASD) επικεντρώνονται στην προσαρμογή, τη συνεργασία και την επικοινωνία. Αν και οι πρακτικές μπορεί να διαφέρουν ανάλογα με το περιβάλλον και τις ανάγκες, εδώ είναι μερικές από τις βασικές πρακτικές που σχετίζονται με την ASD:

I. Κύκλοι προσαρμογής

Η ASD χρησιμοποιεί κύκλους προσαρμογής, που είναι επαναλαμβανόμενες φάσεις ανάπτυξης. Κάθε κύκλος περιλαμβάνει την ανάλυση, τον σχεδιασμό, την υλοποίηση και τη δοκιμή ενός μικρού κομματιού λογισμικού.

II. Δοκιμαστική ανάπτυξη

Η ASD ενθαρρύνει τη δοκιμαστική ανάπτυξη, όπου οι δοκιμές γίνονται πριν από τον κώδικα. Αυτό βοηθά στη διασφάλιση ότι το λογισμικό πληροί τις απαιτήσεις και λειτουργεί σωστά.

III. Συνεχής επικοινωνία

Οι μέθοδοι επικοινωνίας, όπως συναντήσεις ομάδας και ανατροφοδοτήσεις από τους πελάτες, είναι ουσιαστικές για τη συνεχή προσαρμογή του έργου.

IV. Επαναλαμβανόμενοι πειραματισμοί

Η ASD ενθαρρύνει την ενσωμάτωση νέων ιδεών και λύσεων με κάθε κύκλο προσαρμογής, προκειμένου να δοκιμάσει και να εξετάσει διαφορετικές προσεγγίσεις.

V. Αυτοοργανωμένες ομάδες

Οι ομάδες ανάπτυξης έχουν την αυτονομία να λαμβάνουν αποφάσεις για τον τρόπο εκτέλεσης του έργου, να προσαρμόζονται στις αλλαγές και να ανταποκρίνονται στις ανάγκες.

VI. Συνεχής Βελτίωση

Η ASD προωθεί την ανασκόπηση των διαδικασιών και τη βελτίωση τους μετά από κάθε κύκλο προσαρμογής.

4.8.2 Οι ρόλοι της ASD

Γενικά η ASD δεν έχει συγκεκριμένους ρόλους παρόλα αυτά υπάρχουν μερικοί βασικοί που υπάρχουν συνήθως όπως:

I. Μέλη της Ομάδας Ανάπτυξης

Αυτά τα άτομα αποτελούν τον πυρήνα της ομάδας ανάπτυξης και είναι υπεύθυνα για τη σχεδίαση, την υλοποίηση, τις δοκιμές και την προσαρμογή του λογισμικού.

II. Πελάτης

Ο πελάτης είναι υπεύθυνος για την παροχή απαιτήσεων και ανατροφοδότησης σχετικά με το πώς πρέπει να αναπτυχθεί το λογισμικό. Ο διαρκής διάλογος και η συνεργασία με τον πελάτη είναι κρίσιμοι.

III. Συντονιστής

Το μέλος της ομάδας που έχει ως αποστολή να συντονίζει τις δραστηριότητες και την επικοινωνία εντός της ομάδας. Ο συντονιστής βοηθάει στην εξασφάλιση της συνεχούς ροής της ανάπτυξης.

IV. Μέλη της Ομάδας Ποιότητας

Είναι υπεύθυνα για τη διασφάλιση της ποιότητας του λογισμικού μέσω των δοκιμών, της αξιολόγησης και της παρακολούθησης.

Είναι σημαντικό να σημειωθεί ότι η ASD επικεντρώνεται στην ευελιξία και τη συνεργασία, επιτρέποντας στα μέλη της ομάδας να αναλαμβάνουν πολλούς ρόλους και να προσαρμόζονται στις ανάγκες του έργου.

IV.8.3 Οι αξίες της ASD

I. Συνεργασία

Η ASD προωθεί τη στενή συνεργασία μεταξύ των μελών της ομάδας ανάπτυξης, αλλά και με τους πελάτες και τους χρήστες. Η ανοιχτή επικοινωνία και η συνεχής ανταλλαγή απόψεων βοηθούν στην ανάπτυξη ενός βελτιστοποιημένου λογισμικού.

II. Προσαρμοστικότητα

Η ικανότητα να προσαρμόζεται σε αλλαγές και ανατροφοδοσίες είναι ζωτικής σημασίας για την ASD. Αυτή η αξία ενθαρρύνει την ομάδα να είναι ευέλικτη και να προσαρμόζεται γρήγορα σε νέες απαιτήσεις.

III. Διαφάνεια

Η ASD προωθεί τη διαφάνεια στην ανάπτυξη και στην επικοινωνία. Οι πληροφορίες σχετικά με την πρόοδο, τις δυσκολίες και τις αποφάσεις είναι προσβάσιμες για όλα τα μέλη της ομάδας.

IV. Επικέντρωση στον άνθρωπο

Η ASD αξιοποιεί τις ικανότητες και τις απόψεις των μελών της ομάδας. Αντί να επικεντρώνεται μόνο στα εργαλεία και τη διαδικασία, δίνει έμφαση στη συνεργασία και την ανάπτυξη των ανθρώπινων δεξιοτήτων.

V. Συνεχής βελτίωση

Η αξία αυτή ενθαρρύνει την ομάδα να αναλογιστεί και να βελτιώνει συνεχώς τις διαδικασίες και την απόδοσή της.

5 ΜΕΛΕΤΗ ΠΕΡΙΠΤΩΣΗΣ (CASE STUDY)

5.1 Εισαγωγή

Θα αναφερθούμε σε ένα έργο που έγινε στην εταιρία Lan Communications Ε.Π.Ε. όπου εργάζεται ο συγγραφέας της διπλωματικής εργασίας ως Project & Operations Manager. Το έργο που θα ερευνηθεί αφορά ένα πληροφοριακό σύστημα χονδρικής πώλησης οπτικών ινών με 3 μήνες προθεσμία και 20 μέρες περίοδο testing μαζί με τον πελάτη.

Λόγω της κρισιμότητας του έργου αποφασίστηκε ότι θα χρειαστεί διαφορετική διαχειριστική μέθοδο από άλλα project ώστε να βγει όπως πρέπει στον χρόνο που έχει συμφωνηθεί. Μέχρι εκείνη την στιγμή η εταιρία δούλευε μόνο με Waterfall προσέγγιση. Οπότε παρουσίασα στην διοίκηση την ευέλικτη μέθοδο διαχείρισης έργων Scrum όπου και ενθουσιάστηκαν και έδωσαν την έγκρισή τους να προχωρήσουμε.

Η εταιρία έχει τμήμα ανάπτυξης λογισμικού 8 άτομα που εμπλέκονται σε πολλά project που αναπτύσσονται αλλά και συντηρούνται συνεχώς οπότε καταλαβαίνετε ότι θα ήταν υποχρεωτικό να φτιάξουμε την δική μας ευέλικτη παραλλαγή του Scrum για να δουλέψει στα μέτρα μας.

5.2 Εκπαίδευση της ομάδας

Αρχικά δημιουργήθηκαν από τον Project manager – Scrum Master παρουσιάσεις για να εξηγήσουν στην ομάδα της αρχές ευελιξίας από το agile manifesto και τις βασικές αρχές της ευελιξίας ώστε να καταλάβει η ομάδα τα θεμέλια από αυτά που πάμε να εφαρμόσουμε.

Έπειτα έγιναν παρουσιάσεις για το Scrum ώστε να εκπαιδευτούν τα άτομα που θα συμμετέχουν. Οι παρουσιάσεις περιλάμβαναν αναλυτική περιγραφή για το τι είναι το Scrum, τα Scrum events, τα Scrum Artifacts, οι ρόλοι που υπάρχουν και το estimation που γίνεται με τα story points. Τους δόθηκε μαζί με τις παρουσιάσεις και ο Scrum Guide που έχει γραφτεί από τους Ken Schwaber & Jeff Sutherland που δημιούργησαν την συγκεκριμένη μεθοδολογία. Σε αυτό το κομμάτι αναφέρθηκαν και τι προσδοκίες και υποχρεώσεις έχει ο καθένας ανάλογα τον ρόλο του που πρέπει να κάνει κατά την διάρκεια του Sprint.

Μετά από την πρώτη επαφή της ομάδας με την Scrum έγινε η πρώτη συνάντηση στις 21/11/2022 με agenda τα αιτήματα του πελάτη, τον στόχο του έργου και μοιράστηκαν οι ρόλοι του καθενός. Την ομάδα την αποτελούσαν 5 άτομα οπού είχαμε 1 άτομο Scrum Master, 1 άτομο Product Owner, 1 άτομο Dev leader και 2 άτομα Development team.

5.3 Scrum Meetings

Τα Meetings που πρέπει να γίνουν στο Scrum και κάναμε και εμείς είναι τα εξής:

I. Sprint Planning

Πραγματοποιούνται κάθε πρωί την πρώτη μέρα του Sprint με σκοπό να συζητηθεί η δουλειά που θα γίνει κατά την διάρκεια του και οτιδήποτε πρέπει να ειπωθεί για να κυλήσει ομαλά η δουλειά

II. Spring Ending

Γινόταν κάθε απόγευμα την τελευταία μέρα του Sprint με σκοπό να παρουσιαστεί από τον καθένα η δουλειά που έγινε να μιλήσουμε τυχόν κωλύματα και να ενημερωθούν όλοι για την πρόοδο που έγινε.

III. Retrospective

Ακολουθούσε μετά το Sprint Ending και βλέπαμε τα προβλήματα που υπήρξαν στην ροή της εργασίας και τι εμπόδια μπορεί να είχαμε. Επίσης ήταν ευκαιρία να μέλη να προτείνουν τις ιδέες τους για βελτίωση του Scrum όμως και να ευχαριστήσουν κάποιο μέλη για την εργασία του και την προσφορά του στο project.

IV. Daily Stand up

Είχαμε αποφασίσει συγκεκριμένη ώρα της ημέρα να μπαίνουμε στο εργαλείο επικοινωνίας που χρησιμοποιήσαμε χωρίς τον Product Owner ώστε να ενημερώνονται ο Dev leader και το Development team μεταξύ τους για τα εξείς πράγματα.

- Με τι ασχολήθηκαν χθες;
- Με τι θα ασχοληθούν σήμερα;
- Αν υπάρχει κάποιο εμπόδιο που δεν τους αφήνει να προχωρήσουν και αν χρειάζονται κάτι απο την ομάδα;
- Πόσο κοντά είναι στο να τελειώσει τα Task που έχει για το Sprint;

Αυτό γινόταν με σκοπό την διαφάνεια της εργασίας μας ώστε να γνωρίζουν όλοι σε τι κατάσταση είμαστε και για να μπορεί ο Scrum Master να ενημερώνετε ώστε να βοηθάει να αντιμετωπιστούν τυχον κολλήματα που μπορεί να έχει η ομάδα.

V. Backlog Refinement

Αν και δεν είναι επίσημο Scrum meeting γινόταν μια φορά την εβδομάδα μεταξύ των Product Owner, Scrum Master και Dev Leader ώστε να βάζουν προτεραιότητες και να συζητάνε για το τι θα έπρεπε να γίνει στο επόμενο Sprint από την ομάδα.

5.4 Εκκίνηση του έργου

Αποφασίστηκε ότι η πρώτη φάση των 3 μηνών που θα χρειαστούν θα γίνει με sprint της 1 εβδομάδας όπου θα παρακολουθούνται μέσω του προγράμματος Jira ATlassian όπου είναι από τα πιο γνωστά εργαλεία διαχείρισης έργου η εταιρία το λειτουργούσε και πριν αλλά χωρίς να γίνεται χρήση του Scrum ή του Kanban που περιείχε το είχαν απλά για καταγραφή των task. Τα Sprints θα αρχίσουν από 22/11/2022 και θα γίνεται την Δευτέρα κάθε εβδομάδα συνάντηση για την οργάνωση του sprint και την Παρασκευή κάθε εβδομάδα συνάντηση για το κλείσιμο του και την ανατροφοδότηση. Επιπλέον κάθε μέρα το πρωί στις 10 είχαμε daily stand up όπου βρισκόντουσαν όλοι εκτός από τον Product Owner για να δει η ομάδα τι έγινε την

προηγούμενη μέρα, τι θα γίνει σήμερα και αν υπάρχει κάποιο εμπόδιο που δεν τους αφήνει να ολοκληρώσουν το sprint τους όπως έχει συμφωνηθεί. Οπότε μέσα στο Jira υπήρχε το Backlog ιεραρχημένο με την σειρά της προτεραιότητας που πρέπει να γίνουν τα tasks, Sprint Backlog, Sprint Duration και στόχος, Kanban πίνακας του κάθε Sprint για το τι δεν έχει γίνει, τι δουλεύετε τώρα και τι είναι για έλεγχο ώστε να παραδοθεί.

Στην αρχή της πρώτες 2-3 εβδομάδες δεν μπορούσαμε να κάνουμε σωστή εκτίμηση των story points ώστε να καταλάβουμε την δουλειά που πρέπει να γίνει κάθε εβδομάδα αλλά με το που δούλεψε λίγο λύθηκαν τα άτομα της ομάδας με αποτέλεσμα να είμαστε πολύ πιο αποτελεσματικοί και ανά εβδομάδα όλο και πιο ακριβείς αφού βάλουμε και worklog και user stories για να μπορούμε να δούμε πόσο μας παίρνει το κάθε task.

Τα Sprints που τρέξαμε ήταν τα εξής:

Sprint 1: 22/11/2022 - 02/12/2022

Sprint 2: 05/12/2022 - 09/12/2022

Sprint 3: 12/12/2022 - 16/12/2022

Sprint 4: 19/12/2022 - 23/12/2022

Sprint 5: 03/01/2023 - 13/01/2023

Sprint 6: 16/01/2023 - 20/01/2023

Sprint 7: 23/01/2023 - 27/01/2023

Sprint 8: 30/01/2023 - 03/02/2023

Sprint 9: 06/02/2023 - 10/02/2023

Sprint 10: 13/02/2023 - 17/02/2023

Sprint 11: 20/02/2023 - 24/02/2023

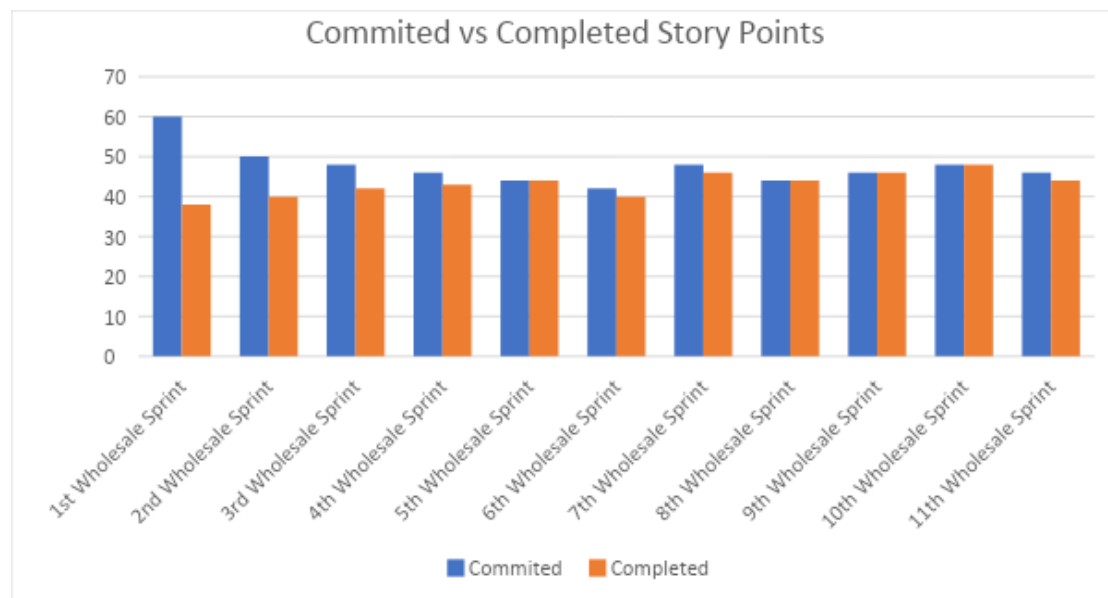
Κάθε Sprint ήταν και Version.

Μέχρι να παραδοθεί και το έργο και μετά έγιναν 2 των 2 εβδομάδων για να φτιάχνονται τα πρώτα ευρήματα του testing τα οποία ήταν στις 28/02/2023 – 10/03/2023 και 13/03/2023 – 24/03/2023 όπου και παραδόθηκε τελικά.

5.5 Ευρήματα κατά την διάρκεια των Sprints

Τα πρώτα Sprints σε μια εταιρία – ομάδα που δουλεύει πρώτη φορά Scrum δεν είναι πολύ αξιόπιστες πηγές συμπερασμάτων μέχρι να εξοικειωθεί η ομάδα με τα εργαλεία του. Οπότε είχαμε αρκετές αλλαγές στο Velocity και στην ανατροφοδότηση κάθε sprint αλλαγές εκτιμήσεων μέχρι να φτάσουμε σε ένα επιθυμητό στάδιο με την βοήθεια της ακολουθίας Fibonacci ώστε να μπορέσουμε να βάλουμε κάποιες βάσεις. Από το 3ο Sprint άρχισε να φαίνεται ότι η ομάδα είναι πολύ πιο άνετη με την χρήση των εργαλείων οπότε βλέποντας την πορεία μέχρι τότε φτάσαμε στο συμπέρασμα ότι το Velocity της ομάδας μας είναι 44 Story Points. Από τότε αρχίσαμε να βάζουμε στόχο την εβδομάδα 40-46 πόντους διαμοιρασμένους στην ομάδα

Βελτιώνοντας από Sprint σε Sprint την εκτίμηση των χρόνων των εργασιών μας φτάσαμε στο συμπέρασμα στο 3ο Sprint ότι το Velocity μας είναι 44 Story Points και από τότε αρχίσαμε να βάζουμε στόχο την εβδομάδα 40-46 πόντους διαμοιρασμένους στην ομάδα ώστε να βγαίνουν σε μια εβδομάδα και σωστά. Όπως παρατηρείτε και στο γράφημα από κάτω στην αρχή είχαμε βάλει πιο μεγάλες προσδοκίες προς τα committed αλλά με την πρώτη ανατροφοδότηση βρήκαμε που ήταν το πρόβλημα και είμαστε όλο και πιο κοντά στο να βγαίνουν όπως είχαν σχεδιαστεί.



Εικόνα 14 Committed vs Completed Story Points. Πηγή: Jira Server

5.6 Αποτελέσματα

Το Έργο παραδόθηκε στην ώρα του όπως ακριβώς ζητήθηκε με πολύ θετικά σχόλια και απο την μεριά της διοίκησης μας αλλά και απο του πελάτη. Συμπεράναμε μετά

από το τελικό μας Retrospective και συζήτηση ότι μας βοήθησε πάρα πολύ στην ευελιξία που είχαμε των task, τα χρονικά πλαίσια που μας έβαζαν τα sprints μαζί με την πιο σωστή τοποθέτηση του καθε task με πιο ξεκάθαρα ζητούμενα που έβγαιναν μέσα από την συζήτηση και όχι απλά απο μια απαίτηση, βελτιώθηκε αρκετά η ομαδικότητα της ομάδας στο πώς να μπαίνουν προτεραιότητες για να μην υπάρχουν εμπόδια σε κάποιον άλλο ώστε να δουλεύουν πιο αποτελεσματικά και παραγωγικά και τέλος βοήθησε στην παραγωγικότητα και ότι υπήρχε ξεκάθαρος στόχος στο τέλος κάθε εβδομάδας που θέλαμε να φτάσουμε και σε θέμα παρακίνησης και διαφάνειας του τι θέλει η ομάδα και ο πελάτης.

Η αρχική εφαρμογή agile μεθοδολογιών πιο συγκεκριμένα στο παράδειγμα μας της Scrum έχει πάντα και υψηλό κίνδυνο για όλους τους εμπλεκόμενους. Όμως όπως και ο ορισμός του agile το φέραμε στα μέτρα μας για αρχή μέχρι να φτάσουμε εκεί που πρέπει και δούλεψε πολύ αποτελεσματικά για την δική μας ομάδα.

5.7 Σύγκριση με την παραδοσιακή μεθοδολογία που χρησιμοποιούσαμε στο παρελθόν

Σε σύγκριση με παλαιότερα έργα στο συγκεκριμένο μας βοήθησε πολύ η επαναληπτική και σταδιακή προσέγγιση ενώ όπου μπορούσαμε να επαναπροσδιορίσουμε πολύ γρήγορα κάποια πράγματα λόγω της υψηλής ευκαμψίας που υπάρχει σε αντίθεση με την παραδοσιακή όπου βγαίνει ένα σχέδιο γραμμικό στην αρχή και απλά εκτελούνταν μέχρι το τέλος όπου παίρναμε την ανατροφοδότηση με αποτέλεσμα να έχουν παρεκκλίνει απο αυτό που ο πελάτης ήθελε πραγματικά. Με αυτή την μεθοδολογία γλυτώσαμε πολύ χρόνο στην ανάπτυξη και λόγω του ότι ο πελάτης συμμετείχε ενεργά και κάθε εβδομάδα παίρναμε ανατροφοδότηση για την πρόοδό μας όπου λόγω των εβδομαδιαίων συναντήσεων είχαμε και μεγάλη ικανότητα προσαρμογής με αποτέλεσμα να μην έχουμε πολλές καθυστερήσεις για διορθώσεις. Επιπλέον υπήρχε μεγάλη ορατότητα του έργου στην ομάδα με αποτέλεσμα να υπάρχει πολύ καλύτερη συνεργασία κώδικα και μοίρασμα ιδεών που οδήγησε σε πολύ καλύτερο προγραμματισμό και ανάπτυξη κώδικα. Τέλος όσοι κίνδυνοι εμφανίστηκαν αντιμετωπίστηκαν χωρίς να δημιουργήσουν μεγάλα προβλήματα λόγω της συνεχούς επαναληπτικής ανάπτυξης όπου βλέπαμε συνέχεια τα εμπόδια μέσω των συναντήσεων μας και συζητήσεων.

6 ΣΥΝΕΝΤΕΥΞΕΙΣ

Για την καλύτερη κατανόηση των αποτελεσμάτων και ευρημάτων του Case Study αλλά και για ανατροφοδότηση γενικώς ώστε να δούμε αν ήταν πετυχημένο για να ξανά ακολουθήσουμε την ίδια μεθοδολογία ενδοεταιρικά έγινε μια τυποποιημένη συνέντευξη με ένα μη δομημένο ερωτηματολόγιο που περιείχε 11 ερωτήσεις όπου είχαμε τις ακόλουθες απαντήσεις από τα πέντε άτομα που συμμετείχαν. Θα αναφέρονται παρακάτω για λόγους απόκρυψης προσωπικών δεδομένων με τον ρόλο τους. Έχουμε τον Product Owner, Scrum master, Dev Leader – Development member και 2 άτομα Development member μόνο.

Κάτω από κάθε ερώτηση θα υπάρχει το συμπέρασμα που βγαίνει από τις απαντήσεις της ομάδας αλλά σας παρέχω και αυτούσιες τις απαντήσεις των μελών ώστε να μπορείτε να βγάλετε και τα δικά σας συμπεράσματα ανάλογα τον ρόλο, την εμπειρία και την θέση του καθενός στο project.

6.1 Ερωτήσεις – Απαντήσεις – Σχολιασμός

I. Ήταν η πρώτη φορά που χρησιμοποιήσατε Agile;

- Product Owner: Ναι
- Scrum Master: Όχι
- Dev Leader – Development Member: Όχι
- Development Member 1: Ναι
- Development Member 2: Ναι

Σχολιασμός: Βλέπουμε ότι η ομάδα κατα τα 3/5 είναι η πρώτη τους φορά που χρησιμοποιούν Agile μεθοδολογία οπότε θα είναι πολύ ενδιαφέρον να δούμε πως θα τους φανεί όπως επίσης και την σύγκριση των 2 πιο έμπειρων μελών στην agile σε σχέση με τα προηγούμενα τους project.

II. Πώς θα αξιολογούσατε την αποτελεσματικότητα του Scrum στο έργο μας;

(Κλίμακα: Πολύ αποτελεσματική, Κάπως αποτελεσματική, Ουδέτερη, Κάπως αναποτελεσματική, Αναποτελεσματική)

- Product Owner: Πολύ αποτελεσματική
- Scrum Master: Πολύ αποτελεσματική
- Dev Leader – Development Member: Πολύ αποτελεσματική

- Development Member 1: Πολύ αποτελεσματική
- Development Member 2: Πολύ αποτελεσματική

Σχολιασμός: Μπορούμε να δούμε ότι η ομάδα ήταν σε απόλυτη αρμονία σε σχέση με την αποτελεσματικότητα του Scrum όπου έβαλαν όλοι ότι ήταν πολύ αποτελεσματική. Αυτό μας δείχνει ότι ήταν ευχαριστημένη με το αποτέλεσμα και την διαχείριση του έργου.

III. Ποια θεωρείτε ότι είναι τα μεγαλύτερα οφέλη από τη χρήση του Scrum στο έργο μας;

- Product Owner: Θεωρώ ότι βοήθησε πολύ στην ευελιξία της ομάδας σχέση αποφάσεων και αλλαγών όπως και στην συνεχή βελτίωση της αποδοτικότητας της ομάδας εβδομάδα με εβδομάδα.
- Scrum Master: Βοήθησε πολύ στην διαφάνεια της ομάδας όπως και στην αυτοοργάνωση της βελτιώνοντας την αποτελεσματικότητά τους.
- Dev Leader – Development Member: Βοηθάει στο να σπάσεις τα deliverables σε μικρότερα task και να τα αποδώσεις σε άτομα, να παρακολουθείς καλύτερα την πορεία του project και να βάζεις πιο σωστά deadlines,
- Development Member 1: Οργάνωση χρόνου, ξεκάθαρα task
- Development Member 2: Η επικοινωνία μεταξύ των μελών της ομάδας (product owner, scrum master, development team κλπ). Η δυνατότητα παρακολούθησης της εξέλιξης του project καθώς και ο ρυθμός ολοκλήρωσης των task από τα επιμέρους μέλη της ομάδας. Η συνεχής καθοδήγηση και ανάλυση κατά την ανάπτυξη ή βελτίωση ενός product

Σχολιασμός: Απο τις απαντήσεις μπορούμε να κατανοήσουμε ότι το κάθε μέλος έχει διαφορετικά οφέλη που θεώρησε σημαντικότερα στην χρήση του Scrum αλλά βλέπουμε ότι κυριαρχεί η οργάνωση του project που οδηγεί και σε καλύτερα αποτελεσματικότητα, η ευελιξία όπως και η παρακολούθηση της συνεχούς προόδου στο έργο από όλους κατευθείαν με το που ολοκληρώνεται κάτι.

IV. Ποιες προκλήσεις ή εμπόδια αντιμετωπίσατε κατά τη διάρκεια της εργασίας σας με το Scrum;

- Product Owner: Υπήρχε δυσκολία στο να διατηρούμε συνέχεια μια ανοδική πορεία μέσω των Retrospective όπως και λίγο δυσκολία με τα αυστηρά ωράρια που θα έπρεπε να έχουμε στην οργάνωση του χρόνου μας για τις συναντήσεις.
- Scrum Master: Όπως είναι φυσιολογικό σε μια ομάδα που το δοκιμάζει για πρώτη φορά στην αρχή υπήρχε μια αντίσταση στην αλλαγή υποσυνείδητη μέχρι να φανούν τα αποτελέσματα της τεχνικής αλλά αντιμετωπίστηκαν όλα σε ένα ικανοποιητικό επίπεδο.
- Dev Leader – Development Member: Αυτό που μας δυσκόλεψε ήταν ότι δεν είχαμε ξεκάθαρο το τι ακριβώς έπρεπε να έχει το αποτέλεσμα, αυτό ήταν κακό γιατί δεν μπορούσαμε να κάνουμε σωστό long term planning αλλά μας βοήθησε το scrum καθώς σε κάθε sprint επαναπροσδιορίζαμε τις ανάγκες οπότε δεν μπορούσε να ξεφύγει το έργο
- Development Member 1: Σε κάποια σημεία υπήρχε υπερανάλυση σπαταλώντας χρόνο.
- Development Member 2: Σε σημεία, υπέρ ανάλυση προβλημάτων. Η ανάγκη της σωστής και οργανωμένης σχεδίασης. Η απόδοση σωστών estimation.

Σχολιασμός: Κατανοούμε από την ομάδα ότι υπήρχε πρόβλημα με την υπερ ανάλυση κάποιων προβλημάτων στα meetings που μπορεί να σημαίνει ότι χρειάζεται βελτίωση η οργάνωση του χρόνου στις συναντήσεις όπως βλέπουμε να υπάρχουν και θέματα στην μακροχρόνια σχεδίαση του έργου που εμπόδιζε να φτιαχτεί ένας σωστός προγραμματισμός για τα μελλοντικά sprints.

- V.** Βρήκατε τις τελετές της Scrum (όπως καθημερινά stand-ups, προγραμματισμό Sprint και αναδρομικά) χρήσιμες για τη βελτίωση της συνεργασίας και της επικοινωνίας;
- Product Owner: Ναι ήταν απαραίτητες σε ένα έργο σαν αυτό που είχε έντονα στοιχεία συνεχών αλλαγών.
 - Scrum Master: Εννοείται υπήρχε μεγάλη διαφορά στην διαφάνεια της πληροφορίας και την μετάδοση της. Όπως και βελτίωση του χρόνου που χανόταν για ερωτήσεις κατανόησης κάποιων task την εβδομάδα μιας και λυνόντουσαν τα πιο πολλά στην αρχή του Sprint.

- Dev Leader – Development Member: Νομίζω ότι λόγω του μεγέθους της ομάδας μας 2+1 devs + product owner τα daily δεν βοηθήσαν πολύ καθώς υπήρχε ήδη επικοινωνία μέσα στην ημέρα για το πως είναι η ομάδα και που βρίσκεται. Το καλό είναι ότι επειδή κάναμε weekly sprints μπορούσαμε να δούμε στο σύνολο τι έγινε μέσα στο sprint.
- Development Member 1: Ναι
- Development Member 2: Ναι. Η συνεχής επικοινωνία και ανάλυση των προβλημάτων παροδικά, καθώς επίσης και η καθημερινή ανάλυση των task βοηθά στην αποτελεσματική ανάπτυξη/βελτίωση ενός product. Δημιουργεί ένα κλίμα ομαδικότητας καθώς επίσης κάνει και εύκολη την παρακολούθηση της τρέχουσας κατάστασης ολοκλήρου του προϊόντος.

Σχολιασμός: Η ομάδα επιβεβαιώνει την σημαντικότητα των τελετών του Scrum μέσα από τις απαντήσεις τους και το κάθε μέλος έχει να αναφέρει κάτι διαφορετικό σαν σημαντικότερη χρησιμότητα τους για τους ίδιους από την διαφάνεια, καλύτερη διαχείριση των συνεχών αλλαγών, καλύτερη συνολική εικόνα και πιο διεξοδική ανάλυση των task.

- VI.** Πιστεύετε ότι τα artifacts της Scrum (όπως το product backlog, το sprint backlog και τα user stories) ήταν χρήσιμα στην καθοδήγηση της διαδικασίας ανάπτυξης;
- Product Owner: Ναι ήταν τα βασικά θεμέλια του Scrum που μας βοήθησαν να οργανώσουμε σωστά την δουλειά που έπρεπε να γίνει για να φτάσουμε στο επιθυμητό αποτέλεσμα.
 - Scrum Master: Εννοείται μιας και εκεί υπάρχει όλη η πληροφορία διαθέσιμη που μπορεί να χρειαστεί η ομάδα οποιαδήποτε στιγμή και με ταξινομημένη προτεραιότητα. Τα user stories επίσης στο τέλος βοήθησαν πολύ στο να κάνουμε estimation και να γνωρίζουμε πόσα tasks πρέπει να μπουν μέσα στα sprints.
 - Dev Leader – Development Member: Ναι αν γίνονται σωστά. Δηλαδή να φροντίζει ο product owner να έχει τον προγραμματισμό στο backlog για να μπορεί μετά να μπει στο sprint.
 - Development Member 1: Ναι

- Development Member 2: Ναι και αυτό γιατί κάνουν εύκολη την όλη διαδικασία της ανάπτυξης, του backtracking, της επικοινωνίας και την παρακολούθηση της εξέλιξης ικανοτήτων κατά άτομο αλλά και κατά ομάδα.

Σχολιασμός: Βλέπουμε χωρίς κανέναν δισταγμό η ομάδα μας κάνει να καταλάβουμε πόσο σημαντικά ήταν καθ'όλη την διάρκεια του έργου τα artifacts της scrum όπου κρατιόταν όλο το ιστορικό των εξελίξεων, όλες οι πληροφορίες που είχαν συζητηθεί και όλες οι εργασίες που ήταν να γίνουν για φτάσουν στο επιθυμητό επίπεδο το λογισμικό τους.

VII. Πώς επηρέασε το Scrum την παραγωγικότητά σας και την ικανότητά σας να τηρείτε τις προθεσμίες του έργου;

- Product Owner: Επηρέασε την ομάδα πολύ θετικά μιας και ήταν όλα πολύ πιο οργανωμένα και γραμμικά στο σχέδιο με αποτέλεσμα να μην έχουμε απώλειες παραγωγικότητας.
- Scrum Master: Όπως φάνηκε και απο το αποτέλεσμα βοήθησε να είναι όλα στην ώρα τους και ακριβώς όπως είχαν σχεδιαστεί.
- Dev Leader – Development Member: Όπως είπα μας βοήθησε να σχεδιάσουμε καλύτερα τον χρόνο υλοποίησης του έργου και να γνωρίζει καλύτερη ολη η ομάδα το που βρισκομαστε και που θελουμε να πάμε.
- Development Member 1: Την επηρέασε θετικά καθώς ήταν ξεκάθαρο το τι πρέπει να κάνω κατά την διάρκεια του sprint.
- Development Member 2: Η όλη διαδικασία βοήθησε σε μεγάλο βαθμό στην παραγωγικότητα αλλά και στην τήρηση των προθεσμιών. Παρότι κάποιες φορές μπορεί να γινόταν over ή under estimation στους χρόνους παράδοσης, το τελικό αποτέλεσμα είχε θετικό αντίκτυπο στους χρόνους παράδοσης. Και δίνοντας στην ομάδα στόχους και milestones που θα έπρεπε να φτάσουν, η παραγωγικότητα αυξήθηκε σε σημαντικό βαθμό.

Σχολιασμός: Συμπεραίνουμε απο τις απαντήσεις ότι η ομάδα βελτίωσε πολύ την παραγωγικότητα της με την Scrum λόγω ότι ήταν όλα πιο οργανωμένα, σχεδιασμένα με σωστές προθεσμίες ακόμα και όταν κάποιες εκτιμήσεις ξέφευγαν ήταν πολύ εύκολο να μπουν ξανά στο πρόγραμμα, βοήθησε πολύ να γνωρίζουν τα μέλη τι έχουν να κάνουν μέσα σε συγκεκριμένες προθεσμίας με αποτέλεσμα να μην σταματάνε για

να παίρνουν οδηγίες για την συνέχεια όλη την ώρα και η διαφάνεια του στόχου και των milestones είχε ως αποτέλεσμα να ξέρει η ομάδα που θέλει να φτάσει και να παρακινείται περισσότερο.

- VIII.** Πιστεύετε ότι το Scrum έχει διευκολύνει την καλύτερη προβολή και διαφάνεια στην πρόοδο και τα παραδοτέα του έργου;
- Product Owner: Ναι
 - Scrum Master: Εννοείται αυτό είναι και ένα από τα σημαντικότερα γνωρίσματα του Scrum.
 - Dev Leader – Development Member: Ναι, μας βοήθησε να ξεκαθαρίσουμε τι θα κάνει ποιο άτομο της ομάδας δίνοντας observability μεταξύ της και μεταξύ των stakeholders του project.
 - Development Member 1: Ναι
 - Development Member 2: Ναι. Η χρήση σωστών εργαλείων scrum (πχ Jira ή Trello), η απονομή των task, η οργάνωση της ομάδας και ο σχεδιασμός του πλάνου αποτελούν μέρος του scrum. Τα παραπάνω επιτυγχάνουν την καλύτερη προβολή και διαφάνεια στην πρόοδο του project καθώς και τους καλύτερους χρόνους παράδοσης.

Σχολιασμός: Ένα από τα βασικά γνωρίσματα της Scrum είναι η διαφάνεια των πληροφοριών ενός έργου ανάμεσα στα μέλη της ομάδας όπως φαίνεται και από την ομάδα συμφωνούν απόλυτα και βοήθησε πολύ στην επικοινωνία με τους stakeholders αλλά και την ομάδα για να γνωρίζει ακριβώς την κατεύθυνση του λογισμικού που δημιουργούνε αλλά και πως θα χρησιμοποιηθεί μετά βοηθώντας τους να ξέρουν καλύτερα πως πρέπει να σχεδιαστεί.

- IX.** Πώς θα περιγράφατε το επίπεδο συνεργασίας και ομαδικής εργασίας στην ομάδα μας στο Scrum;
- Product Owner: Βοήθησε πολύ και ήταν σε πολύ καλύτερο επίπεδο με παρελθοντικά έργα.
 - Scrum Master: Βελτιώθηκε πολύ αλλά σίγουρα ακόμα χρειάζεται δουλειά.

- Dev Leader – Development Member: Η ομάδα των προγραμματιστών πιστεύω ότι τηρησε σχεδόν αψογα το scrum και τις τελετές του, θα ήθελα παραπάνω προσοχή από τους product owners/stakeholders.
- Development Member 1: Πολύ καλό.
- Development Member 2: Ήταν πολύ καλό το επίπεδο και αρκετά ικανοποιητική η συνεργασία, προσέφερε μια ευκολία στην επικοινωνία και την κατανόηση των προβλημάτων που προέκυπταν ή μπορεί να προέκυπταν κατά την ανάπτυξη και εξέλιξη του προϊόντος.

Σχολιασμός: Τα άτομα που ήταν πρώτη φορά που χρησιμοποιούσαν Scrum φαίνεται είναι ευχαριστημένα με την συνεργασία της ομάδας αν και τα 2 πιο έμπειρα άτομα αναφέρουν ότι υπάρχει ακόμα πολύ χώρος για βελτίωση.

- X.** Βρήκατε τη χρήση εργαλείων ή λογισμικού Scrum επωφελής στη διαχείριση του έργου; (Jira)
- Product Owner: Εννοείται χωρίς αυτό δεν θα μπορούσε να γίνει η δουλειά μας.
 - Scrum Master: Εννοείται δεν γίνεται να δουλέψει κάτι τέτοιο χωρίς το κατάλληλο εργαλείο διαχείρισης του έργου.
 - Dev Leader – Development Member: Ναι, με αυτόν τον τρόπο υπάρχει log και ασυγχρόνα μπορεί όλη η ομάδα να ανανεώνει την κατάσταση των issues της
 - Development Member 1: Ίσως το σημαντικό εργαλείο κατά την διάρκεια του πρότζεκτ.
 - Development Member 2: Ναι, εργαλεία όπως το jira αποτελούν πολύ σημαντικό κομμάτι τους scrum και των agile μεθοδολογιών. Η χρήση τους διευκολύνει όλη την ομάδα, ξεκαθαρίζει τα tasks και ποιος τα αναλαμβάνει, δηλώνει χρόνους για την ολοκλήρωση τους, δηλώνει την τρέχουσα κατάσταση του κάθε task καθώς και ολόκληρου του project και πολλά άλλα εργαλεία που βοηθούν στην κατανόηση των προβλημάτων, στην δημιουργία στατιστικών κ.ο.κ.

Σχολιασμός: Απο τις απαντήσεις της ομάδας εδώ καταλαβαίνουμε την σημαντικότητα των εργαλείων διαχείρισης έργων. Όλοι το θεωρούν απο τα σημαντικότερα εφόδια που είχαμε αφού χωρίς αυτό δεν θα μπορούσε να γίνει εύκολα

και σωστά η παρακολούθηση της προόδου του έργου, η διάθεση των καθηκόντων, η πρόβλεψη των χρόνων παράδοσης με την βοήθεια του ιστορικού που υπάρχει μέσα στην εφαρμογή και η εύκολη διαχείριση των backlog με την βοήθεια της απεικόνισης που τους παρέχει.

- XI.** Συνολικά, ποιο είναι το επίπεδο ικανοποίησής σας με την προσέγγιση Scrum για το έργο ανάπτυξης λογισμικού μας;
- Product Owner: Αρκετά καλό.
 - Scrum Master: Πήγε καλά αλλά σίγουρα έχουμε πολλά να βελτιώσουμε στο επόμενο και να φτάσουμε στο επίπεδο που θέλουμε να δουλεύουμε σαν ομάδα.
 - Dev Leader – Development Member: Θα πω ότι είναι σε ένα 7/10 , πιστεύω ότι έχουμε περιθώρια βελτίωσης για να μπορέσουμε να το αξιοποιήσουμε πιο σωστά.
 - Development Member 1: Έμεινα αρκετά ικανοποιημένος, καθώς πριν την εφαρμογή του, ένιωθα πως δεν ξέρω τι πρέπει να κάνω και από που να αρχίσω.
 - Development Member 2: Το επίπεδο ικανοποίησης είναι υψηλό. Η χρήση του scrum βοήθησε σε πολλά μέρη της ανάπτυξης, της επικοινωνίας, της τελειοποίησης καθώς και της εξέλιξης του project μετά την ολοκλήρωσή του.

Σχολιασμός: Τα μέλη της ομάδας σε γενικές γραμμές είναι πολύ ευχαριστημένα με το επίπεδο που δούλεψε το έργο και την χρήση του Scrum λόγω της βελτίωσης που υπήρχε στην επικοινωνία και την εξέλιξη του έργου γενικά αλλά σίγουρα χρειάζονται και άλλες βελτιώσεις όσο προχωράει η ομάδα και θα χρησιμοποιεί την μεθοδολογία Scrum.

7 ΣΥΜΠΕΡΑΣΜΑΤΑ

Στόχος της παρούσας διπλωματικής εργασίας ήταν να ο αναγνώστης να καταλάβει την χρησιμότητα των ευέλικτων μεθοδολογιών στην ανάπτυξη λογισμικού και ακόμα πιο σημαντικό ότι το κάθε έργο και ομάδα μπορεί να της ταιριάζει συγκεκριμένη μεθοδολογία και τρόπος λειτουργίας ώστε να είναι αποτελεσματική, δεν υπάρχει πάντα στάνταρ μεθοδολογία που θα είναι η πιο αποτελεσματική. Οπότε αρχικά μέσω

μιας βιβλιογραφικής ανάλυσης της agile μεθοδολογίας και αργότερα των υπο μεθοδολογιών που περιλαμβάνει βοήθησε να υπάρχει συγκεντρωμένη όλη η πληροφόρηση που χρειάζεται κάποιος για να το καταλάβει. Στην συνέχεια μέσω της μελέτης περίπτωσης και συνέντευξης των μελών που συμμετείχαν ο στόχος ήταν να δείξουμε τα αποτελέσματα που έχει μια τέτοια μεθοδολογία στο κατάλληλο έργο και πως βιώνουν την αλλαγή άτομα που την χρησιμοποιούν για πρώτη φορά στον εργασιακό τους χώρο.

Μέσω της βιβλιογραφικής ανασκόπησης που έγινε πετύχαμε τον στόχο της πληροφόρησης του αναγνώστη με τις περισσότερες και δημοφιλέστερες μεθοδολογίες agile ώστε να μπορεί να επιλέξει με την ομάδα του την καλύτερη για αυτή και μέσω της μελέτης περίπτωσης και της συνέντευξης μπορεί να καταλάβει τα θετικά αποτελέσματα και στην αποτελεσματικότητα του έργου αλλά και στην ψυχολογία των μελών της ομάδας ώστε να τους είναι πιο εύκολη η δουλειά ψυχολογικά νιώθοντας πιο οργανωμένοι και ξέροντας τι έχουν να κάνουν.

Από τότε που η τεχνολογία αναπτύσσεται τόσο γρήγορα και πολλά έργα χρειάζονται συνεχώς να γίνονται αλλαγές εν μέσω της ανάπτυξης τους οι agile μεθοδολογίες είναι μονόδρομος για όλα τα έργα με αυτά τα χαρακτηριστικά. Η προσαρμοστικότητα τους και η δυνατότητα να «φτιαχτεί» όπως θα την ήθελε η κάθε ομάδα για να είναι αποτελεσματική είναι τεράστιο πλεονέκτημα εναντίων της παλαιάς παραδοσιακής μεθοδολογίας του καταρράκτη.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Καπόπουλος, Κομισόπουλος, Παλόγλου (2020) Εθνική Σχολή Δημόσιας Διοίκησης & Αυτοδιοίκησης Εκπαιδευτικό Υλικό Διαχείριση Έργων
2. Agile Practice Guide (2017) Project Management Institute PMI
3. Project Management Institute. (2017). A guide to the Project Management Body of Knowledge (PMBOK guide) (6th ed.)
4. How Agile Methods Inspire Project Management - The Half Double Initiative (2016) Heeager, Svejvig, and Schlichter
5. digital.ai. (2022). 16th Annual State of Agile Report. In <https://digital.ai/resource-center/analyst-reports/state-of-agile-report/>
6. Manifesto for Agile Software Development (2001) K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J.

- Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. Martin, S. Mellor, K. Schwaber, J. Sutherland, and D. Thomas
7. <https://www.atlassian.com/> (Many information's about every methodology)
 8. <http://www.extremeprogramming.org/> (2013)
 9. <https://agilemanifesto.org/> (2001)
 10. <https://www.scrum.org/resources/scrum-guide> (2020)
 11. The Scrum Guide (2020) Ken Schwaber, Jeff Sutherland
 12. <https://www.scrumalliance.org/> (2024)
 13. Linear models vs Agile models: Making the right model decision (2009) Aline U. Hakizabera*, Koichi Yamada
 14. Adaptive Software Development (1999) James A. Highsmith
 15. Raut, Laukik & Wakode, Rajat & Talmale, Pravin. (2015). Overview on Kanban Methodology and its Implementation. International Journal for Scientific Research & Development. 03. 2518-2521.
 16. Extreme Programming Explained: Embrace Change, 2nd Edition (The XP Series) (2004) Kent Beck, Cynthia Andres
 17. DSDM: The Method in Practice. Addison-Wesley Longman Publishing Co., Inc., USA. (1997) Jennifer Stapleton.
 18. Steve R. Palmer and Mac Felsing. 2001. A Practical Guide to Feature-Driven Development (1st. ed.). Pearson Education.
 19. Java Modeling In Color With UML: Enterprise Components and Process (1999) Peter Coad, Jeff de Luca, Eric Lefebvre.
 20. Agile Software Development (2000) Alistair Cockburn
 21. Anwer, Faiza & Aftab, Shabib & Waheed, Usman & Muhammad, Syed. (2017). Agile Software Development Models TDD, FDD, DSDM, and Crystal Methods: A Survey. INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY SCIENCES AND ENGINEERING. 8. 1-10.
 22. Than, Myint. (2022). An Analysis on Adaptive Software Development (ASD) Framework. 10.13140/RG.2.2.16905.11364.
 23. Johann Strasser, (2021) Comparing Project Management Methods: Agile, Traditional or Hybrid