



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ Μ/Υ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΤΜΗΜΑ ΒΙΟΜΗΧΑΝΙΚΗΣ ΔΙΟΙΚΗΣΗΣ & ΤΕΧΝΟΛΟΓΙΑΣ

ΔΙΑΠΑΝΕΠΙΣΤΗΜΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

«ΤΕΧΝΟ-ΟΙΚΟΝΟΜΙΚΑ ΣΥΣΤΗΜΑΤΑ»



Διοίκηση έργων μέσω σύγχρονων μεθόδων: Μελέτη περίπτωσης το εργαλείο Microsoft Azure DevOps

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

ΧΡΥΣΟΥΛΑΣ ΖΑΧΑΡΗ

Επιβλέπων : Δημήτριος Ασκούνης
Καθηγητής Ε.Μ.Π.

Αθήνα, Μάιος 2022



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ Μ/Υ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΤΜΗΜΑ ΒΙΟΜΗΧΑΝΙΚΗΣ ΔΙΟΙΚΗΣΗΣ & ΤΕΧΝΟΛΟΓΙΑΣ

ΔΙΑΠΑΝΕΠΙΣΤΗΜΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

«ΤΕΧΝΟ-ΟΙΚΟΝΟΜΙΚΑ ΣΥΣΤΗΜΑΤΑ»



Διοίκηση έργων μέσω σύγχρονων μεθόδων: Μελέτη περίπτωσης το εργαλείο Microsoft Azure DevOps

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

ΧΡΥΣΟΥΛΑΣ ΖΑΧΑΡΗ

Επιβλέπων : Δημήτριος Ασκούνης
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή Μαΐου 2022.

(Υπογραφή)

.....
Δημήτριος Ασκούνης
Καθηγητής Ε.Μ.Π.

(Υπογραφή)

.....
Χρυσόστομος Δούκας
Αναπληρωτής Καθηγητής Ε.Μ.Π.

(Υπογραφή)

.....
Ιωάννης Ψαρράς
Καθηγητής Ε.Μ.Π.

Αθήνα, Μάιος 2022

(Υπογραφή)

.....

ΧΡΥΣΟΥΛΑ ΖΑΧΑΡΗ

Πτυχιούχος Βιομηχανικής Διοίκησης και Τεχνολογίας ΠΑ.ΠΕΙ.

Copyright © 2019 – Με επιφύλαξη παντός δικαιώματος, All rights reserved

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό.

Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα.

Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς το συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται στο παρόν έγγραφο εκφράζουν το συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Η παρούσα διπλωματική εργασία εκπονήθηκε στο πλαίσιο του Διατμηματικού Προγράμματος Μεταπτυχιακών Σπουδών «Τεχνο-οικονομικά Συστήματα» που προσφέρεται από το τμήμα των Ηλεκτρολόγων Μηχανικών και Μηχανικών Ηλεκτρονικών Υπολογιστών του Εθνικού Μετσόβιου Πολυτεχνείου και του Τμήματος Βιομηχανικής Διοίκησης και Τεχνολογίας του Πανεπιστημίου Πειραιά. Πραγματεύεται την αποτελεσματική διαχείριση και διοίκηση έργων με την χρήση σύγχρονων τεχνικών και μεθόδων που έχουν αναπτυχθεί. Στόχο της εν λόγω εργασίας αποτελεί η παρουσίαση της προστιθέμενης αξίας που προσδίδουν στις επιχειρήσεις τα εργαλεία DevOps, καθώς και οι πλατφόρμες μέσα από τις οποίες λαμβάνει χώρα. Έμφαση δίνεται στην πλατφόρμα της Microsoft Azure DevOps που χρησιμοποιείται από την Uni Systems για την διεκπεραίωση των European Projects που αναλαμβάνει.

Στο πλαίσιο της εν λόγω διπλωματικής εργασίας μελετήθηκε η γενικότερη έννοια των εργαλείων και μεθόδων που εφαρμόζονται και η χρησιμότητά τους στην διοίκηση έργων. Ιδιαίτερη έμφαση δόθηκε στη μέθοδο DevOps και τα πλεονεκτήματα που τη συνοδεύουν, μιας και διαφοροποιούμενο από τα μέχρι πρότινος συνήθη Agile Methods καταφέρνει να συνδυάζει τις διαδικασίες και τις ροές όλων των διαθέσιμων καναλιών προκειμένου να επιτύχει το καλύτερο δυνατό αποτέλεσμα τόσο για την αποτελεσματική διαχείριση του έργου, όσο και για την επιχείρηση που το εφαρμόζει.

Προκειμένου να παρουσιαστούν όλες οι δυνατές λύσεις που προσφέρονται στη σημερινή αγορά, συμπεριλαμβανομένων των ιδιοτήτων αλλά και των χαρακτηριστικών που η κάθε μια εξ αυτών φέρει, διενεργήθηκε ανασκόπηση η οποία αποτελεί και βασικό κεφάλαιο της παρούσας εργασίας. Τέλος, για λόγους πληρότητας και για να καταστεί πλήρως σαφές προς τον δυνητικό αναγνώστη το σύνολο των δυνατοτήτων που προσφέρει μια τέτοια πλατφόρμα αλλά και η ευκολία υλοποίησης ενός έργου μέσα από αυτή, το τελευταίο κεφάλαιο περιλαμβάνει μια ολοκληρωμένη μελέτη περίπτωσης χρήσει του Microsoft Azure DevOps. Όπου

παρουσιάζεται εκτενώς η σειρά των οθονών και η λογική αλληλουχία των βημάτων για την διαχείριση του έργου.

Λέξεις Κλειδιά: DevOps, Microsoft Azure DevOps, Agile Methods, Διοίκηση έργων, ECDC, EpiPulse

Abstract

This diploma thesis was developed within the framework of the master program "Techno-economic Systems", offered by the school of Electrical and Computer Engineering of the National Technical University of Athens and the school of Industrial Management and Technology of the University of Piraeus. It deals with the effective business management of projects using modern techniques and methods that have been developed. This work aims to present the added value that DevOps tools give to businesses, as well as the platforms through which it takes place. Emphasis is placed on the Microsoft Azure DevOps platform used by Uni Systems to handle the European Projects it undertakes.

In the context of this dissertation, the general concept of the tools and methods applied and their usefulness in project management was studied. Particular emphasis was placed on the DevOps method and the advantages that accompany it, as different from the previously usual Agile Methods, it manages to combine the processes and flows of all available channels to achieve the best possible result for both effective project management as well as for the company that implements it.

To present all the possible solutions offered in the current market, including the particularities but also the characteristics that each of them has, a review was carried out which is a key chapter of this work. Finally, for the sake of completeness and to make it fully clear to the potential reader all the possibilities offered by such a platform and the ease of implementing a project through it, the last chapter includes a comprehensive case study using Microsoft Azure DevOps. Where the sequence of screens and the logical sequence of steps for project management are presented in detail.

Keywords: DevOps, Microsoft Azure DevOps, Agile Methods, Project Management, ECDC, EpiPulse

Ευχαριστίες

Με την παρούσα διπλωματική εργασία ολοκληρώνεται ο κύκλος σπουδών μου στο Διατμηματικό Πρόγραμμα Μεταπτυχιακών Σπουδών «Τεχνο-Οικονομικά Συστήματα» και θα ήθελα να ευχαριστήσω όλους όσους με βοήθησαν και συνέβαλαν στην προσπάθεια μου αυτή.

Πρώτα από όλα, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου, τον Καθηγητή κ. Δημήτριο Ασκούνη, για τη βοήθεια και τη στήριξη του, για τις γνώσεις και τα ερεθίσματα που μου μετέδωσε κατά τη διάρκεια των σπουδών μου και ειδικότερα κατά την εκπόνηση της διπλωματικής μου εργασίας, όπου η πόρτα του ήταν πάντα ανοιχτή για μένα για τυχόν απορίες και διευκρινήσεις.

Επίσης θα ήθελα να ευχαριστήσω όλους τους καθηγητές μου, οι οποίοι μου μετέδωσαν τις γνώσεις τους και με επηρέασαν, ο κάθε ένας με τον τρόπο του, αλλά και τους συμφοιτητές και φίλους μου, οι οποίοι με βοήθησαν αυτά τα χρόνια.

Τέλος, θα ήθελα να ευχαριστήσω θερμά την οικογένειά μου για τη συνεχή υποστήριξη, οικονομική αλλά κυρίως ηθική, στην οποία χρωστάω μεγάλο μέρος της σημερινής μου σταδιοδρομίας.

Πίνακας περιεχομένων

1	Εισαγωγή.....	1
1.1	Διοίκηση και διαχείριση έργων.....	1
1.2	Η εξέλιξη των εργαλείων για την διαχείριση έργων.....	5
1.3	Αντικείμενο διπλωματικής.....	6
1.4	Οργάνωση κειμένου.....	6
2	Μεθοδολογία διαχείρισης έργων	7
2.1	Μεθοδολογία Waterfall.....	7
2.2	Μεθοδολογία PM ²	11
2.3	Μεθοδολογία Lean.....	13
2.4	Μεθοδολογία Critical path.....	16
2.5	Μεθοδολογία PRINCE2	19
2.6	Μεθοδολογία Agile.....	22
2.6.1	<i>Scrum</i>	25
2.6.2	<i>Kanban</i>	29
2.6.3	<i>RAD</i>	31
2.7	DevOps Extension of Agile	34
3	Εργαλεία και Πλατφόρμες DevOps.....	37
3.1	Οφέλη του DevOps	37
3.2	Πλατφόρμες DevOps	42
3.2.1	<i>AWS Cloud Computing and Storage in DevOps</i>	43
3.2.2	<i>Azure Cloud Computing and Storage in DevOps</i>	49
3.2.3	<i>Version Control Tool: Git (GitLab, GitHub, Bitbucket)</i>	50
3.2.4	<i>Continuous Integration Tool: Jenkins</i>	52
3.2.5	<i>Container Platforms: Docker</i>	55
3.2.6	<i>Communication and Collaboration: Slack</i>	56
3.2.7	<i>Testing Tool: Selenium</i>	60
3.2.8	<i>Issue and project tracking platform: Jira</i>	65

3.2.9	<i>Error monitoring and crash reporting platform: Raygun</i>	68
3.2.10	<i>Automated deployment and release management platform: Octopus Deploy...</i>	70
3.3	Συγκριτική ανάλυση των πλατφορμών DevOps	73
3.3.1	<i>DevOps Market.....</i>	74
4	Πλατφόρμα Microsoft Azure DevOps.....	76
4.1	Αρχιτεκτονική του Azure DevOps.....	76
4.2	Υπηρεσίες του Azure DevOps	77
4.2.1	<i>Azure Repos</i>	78
4.2.2	<i>Azure Pipelines</i>	80
4.2.3	<i>Azure Boards</i>	81
4.2.4	<i>Azure Test Plans</i>	83
4.2.5	<i>Azure Artifacts</i>	85
5	Διαχείριση έργου για το Ευρωπαϊκό Κέντρο Πρόληψης και Ελέγχου Νόσων (ECDC) μέσω Azure DevOps	88
5.1	EpiPulse Platform	90
5.2	Διαχείριση του έργου στο Azure DevOps	91
6	Συμπεράσματα	101
7	Βιβλιογραφία	103

1

Εισαγωγή

1.1 Διοίκηση και διαχείριση έργων

Η διαχείριση έργου ορίζεται ως η διαδικασία καθοδήγησης ενός έργου από την αρχή μέχρι την ολοκλήρωση του κύκλου ζωής του. Ο κύριος στόχος της διαχείρισης έργου είναι η ολοκλήρωση ενός έργου εντός των καθορισμένων στόχων χρόνου, προϋπολογισμού και ποιότητας. Τα έργα έχουν κύκλους ζωής αφού δεν προορίζονται να διαρκέσουν για πάντα. Ένας κύκλος ζωής διαχείρισης έργου ξεκινά με την έναρξη του έργου και τελειώνει όταν το έργο είτε ολοκληρωθεί είτε τερματιστεί. Τα στάδια διαχείρισης έργου είναι:

- Έναρξη: Ο στόχος αυτής της φάσης είναι να καθοριστεί το έργο.
- Σχεδιασμός: Αυτή η φάση περιλαμβάνει την ανάπτυξη ενός οδικού χάρτη που πρέπει να ακολουθήσουν όλοι.
- Εκτέλεση & Παρακολούθηση: Σε αυτό το στάδιο, δημιουργείται η ομάδα έργου και δημιουργούνται παραδοτέα. Οι διαχειριστές έργων θα παρακολουθούν και θα μετρούν την απόδοση του έργου για να διασφαλίσουν ότι παραμένει σε καλό δρόμο.
- Κλείσιμο: Το έργο ολοκληρώνεται, πραγματοποιείται νεκροψία και το έργο μεταφέρεται σε άλλη ομάδα που θα το συντηρήσει.

(kissflow.com/project/project-management-basics/)

Ο ρόλος του διαχειριστή έργου (project manager) είναι να επιτελεί ταυτόχρονα πολλές εργασίες. Πρέπει να λαμβάνονται συνεχώς αποφάσεις σε όλα τα επίπεδα σχετικά με τη χρήση πόρων, τις προσαρμογές του

χρονοδιαγράμματος, θέματα προϋπολογισμού, διαχείριση ανθρώπινων σχέσεων, επικοινωνία και τεχνικά προβλήματα. Συνεπώς υπάρχει ανάγκη να προσδιοριστούν τα κύρια θέματα, στρατηγικά, τακτικά ή λειτουργικά, ώστε να οριστούν οι προτεραιότητες και να μπορέσει ο project manager να εστιάσει στα κρίσιμα θέματα, τα οποία εναλλάσσονται ανάλογα με τη φάση στην οποία βρίσκεται το έργο.

Σύμφωνα με τον Balachandra οι δέκα παράγοντες επιτυχίας ενός έργου είναι οι ακόλουθοι:

- Στόχος έργου: Ο ορισμός ξεκάθαρων στόχων αποτελεί κλειδί για τον προγραμματισμό και την εκτέλεση ενός έργου. Η κατανόηση των μέτρων απόδοσης και αξιολόγησης είναι σημαντική ώστε να γίνεται καλός συντονισμός. Συνεπώς, όλοι οι εμπλεκόμενοι πρέπει να είναι εξ αρχής ενήμεροι για τους στόχους του έργου.
- Υποστήριξη από την πλευρά της διοίκησης: Ο ανταγωνισμός για πόρους σε συνδυασμό με το μεγάλο βαθμό αβεβαιότητας που διέπουν ένα έργο συχνά οδηγούν σε σύγκρουση και κρίση. Η συνεχής παρουσία της διοίκησης σε ολόκληρο το κύκλο ζωής του έργου βοηθά στην κατανόηση του στόχου του έργου και της σημασίας του. Αυτή η συνειδητοποίηση οδηγεί σε υποστήριξη η οποία μπορεί να αποδειχθεί ανεκτίμητη για την επίλυση προβλημάτων σε περιπτώσεις σύγκρουσης ή κρίσης ή όταν προκύπτει αβεβαιότητα. Συνεπώς, η ξεκάθαρη και συχνή επικοινωνία μεταξύ του project manager και της διοίκησης δρα καταλυτικά για την επιτυχία ενός έργου.
- Προγραμματισμός έργου. Η μετατροπή του στόχου και των μέτρων απόδοσης σε ένα εφικτό πλάνο είναι ο σύνδεσμος μεταξύ της φάσης θεωρητικού σχεδιασμού (conceptual design) και της φάσης παραγωγής. Ένα λεπτομερειακό πλάνο που να καλύπτει θέματα τεχνικά, οικονομικά, οργανωτικά, επικοινωνίας, ελέγχου και χρονοδιαγράμματος είναι η βάση για την υλοποίηση. Ο προγραμματισμός δεν τελειώνει όταν ξεκινά η εκτέλεση καθώς οι ανάγκη για αλλαγές ή μετατροπές είναι συνεχής. Ο προγραμματισμός είναι συνεπώς δυναμικός και διαρκής και συνδέει τους

εναλλασσόμενους στόχους και την απόδοση με τα τελικά αποτελέσματα.

- Συνεργασία με τον Πελάτη. Ο τελικός χρήστης του έργου είναι και ο τελικός κριτής της επιτυχίας του. Ένα έργο το οποίο τελείωσε εγκαίρως, σύμφωνα με τα επιθυμητά τεχνικά χαρακτηριστικά και εντός του προϋπολογισμού, αλλά δε χρησιμοποιήθηκε ποτέ ή σπανίως μπορεί με βεβαιότητα να θεωρηθεί αποτυχία. Στη φάση θεωρητικού σχεδιασμού είναι πολύ σημαντικό να έχουμε καλή επικοινωνία με τον πελάτη (εσωτερικό ή εξωτερικό), ώστε οι στόχοι που θα τεθούν να είναι πλήρως ευθυγραμμισμένοι με τις ανάγκες του πελάτη. Στις επόμενες φάσεις είναι απαραίτητη η συνεχής συνεργασία με τον πελάτη ώστε να διορθώνονται πιθανά λάθη κατά τη μετατροπή των στόχων σε μέτρα απόδοσης. Ωστόσο, λόγω εναλλασσόμενων αναγκών και συνθηκών, παρόλο που στις πρώτες φάσεις είναι χρήσιμο να υπάρχει μη δήλωση των ακριβών αναγκών του πελάτη, αυτό πιθανά παύει να ισχύει στη φάση προγραμματισμού ή υλοποίησης. Το σύστημα configuration management αποτελεί σύνδεσμο μεταξύ υπαρχόντων πλάνων και change requests του πελάτη και της ομάδας του έργου.
- Θέματα προσωπικού. Η ικανοποιητική επίτευξη των τεχνικών στόχων χωρίς να παραβιαστεί το χρονοδιάγραμμα ή ο προϋπολογισμός δε σημαίνει ότι το έργο είναι απολύτως επιτυχημένο, ακόμη και αν ο πελάτης είναι ευχαριστημένος. Εάν οι διάφοροι εμπλεκόμενοι στο έργο δε διατηρούν καλές σχέσεις, η επιτυχία του έργου είναι αμφισβητούμενη, καθώς η καλή συνεργασία και η αφοσίωση στο έργο είναι απαραίτητες για την επιτυχία.
- Τεχνικά θέματα. Η τεχνική κατάρτιση του προσωπικού και η εκπλήρωση των τεχνικών προδιαγραφών πρέπει να είναι από τις πρώτες έννοιες ενός project manager καθώς χωρίς αυτές ένα έργο δεν μπορεί να ολοκληρωθεί.
- Αποδοχή από τον πελάτη. Η συνεχής συνεργασία (consultation) με τον πελάτη καθ' όλο τον κύκλο ζωής ενός έργου αυξάνει την πιθανότητα επιτυχίας, όσον αφορά την αποδοχή χρήστη. Στα τελικά στάδια της υλοποίησης, ο πελάτης πρέπει να κρίνει το έργο που προκύπτει και να

αποφασίζει εάν είναι αποδεκτό ή όχι. Σε περίπτωση που το έργο δε γίνει αποδεκτό σε αυτό το στάδιο, τότε είναι αποτυχημένο.

- Έλεγχος έργου. Η συνεχής ροή πληροφοριών σχετικά με την πραγματική πρόοδο και ο μηχανισμός ανατροφοδότησης επιτρέπουν στον project manager να αντιμετωπίζει την αβεβαιότητα. Συγκρίνοντας την πραγματική πρόοδο με τα τρέχοντα πλάνα, ο project manager μπορεί να διακρίνει παρεκκλίσεις, να αναμένει προβλήματα και να ξεκινά διορθωτικές κινήσεις. Οι όποιες παρεκκλίσεις από τα αρχικά πλάνα μπορούν να διορθωθούν όταν γίνουν αντιληπτές εγκαίρως.
- Επικοινωνία. Η επιτυχής μετάβαση ανάμεσα στις φάσεις του κύκλου ζωής ενός έργου και ο καλός συντονισμός ανάμεσα στους συμμετέχοντες σε κάθε φάση απαιτεί μία συνεχή ανταλλαγή πληροφοριών. Γενικά, η επικοινωνία διευκολύνεται εάν η γραμμή της εξουσίας είναι ξεκάθαρη. Η οργανωτική δομή του έργου πρέπει να περιγράφει τα κανάλια επικοινωνίας και το είδος της πληροφορίας που πρέπει να διέρχεται από αυτά. Επιπλέον, πρέπει να υπάρχει ξεκάθαρη οδηγία σχετικά με το πόσο συχνά πρέπει αυτές οι πληροφορίες να παράγονται και να μεταδίδονται. Οι επίσημες γραμμές επικοινωνίας, καθώς και η ανεπίσημη ροή ανάμεσα στα μέλη της ομάδας συνηγορούν στην επιτυχία του έργου. Η επικοινωνία πρέπει να ορίζεται ξεκάθαρα σε ένα πλάνο επικοινωνίας. Το πλάνο επικοινωνίας είναι απαραίτητο προκειμένου να δημιουργηθεί μια σωστή ομάδα η οποία θα πρέπει να επικοινωνεί. .
- Επίλυση προβλημάτων. Το σύστημα ελέγχου είναι σχεδιασμένο να μπορεί να βρίσκει τις προβληματικές περιοχές και, εάν είναι εφικτό, να βρίσκει την πηγή τους. Επειδή η αβεβαιότητα είναι συχνή πληγή για την ολοκλήρωση των έργων, η ανάπτυξη ενός εναλλακτικού σχεδίου (contingency plan), είναι καλό προληπτικό μέτρο. Η διαθεσιμότητα προετοιμασμένων πλάνων και διαδικασιών για τη διαχείριση προβλημάτων μπορεί να μειώσει τον κόπο που μπορεί να χρειαστεί ώστε να επιλυθούν και το έργο να ολοκληρωθεί ως εάν δεν υπήρχαν αυτά.(el.wikipedia.org)

1.2 Η εξέλιξη των εργαλείων για την διαχείριση έργων

Μόλις τη δεκαετία του 1900 άρχισε να διαμορφώνεται η διαχείριση έργων, καθώς τα έργα βιομηχανοποιήθηκαν, η διαδικασία διαχείρισής τους γνώρισε επίσης επανάσταση. Στη δεκαετία του 1910, ο Henry Gantt εφηύρε το διάγραμμα Gantt, το οποίο σε εκείνο το σημείο κατασκευαζόταν στο χέρι σε έντυπα αντίγραφα και χρησιμοποιήθηκε για τη διαχείριση της κατασκευής του φράγματος Hoover και του συστήματος διακρατικών αυτοκινητοδρόμων. Την ίδια περίοδο το 1911, ο Frederic Taylor δημοσίευσε ένα βιβλίο για τις τακτικές διαχείρισης έργων σχετικά με την αποτελεσματικότητα των ειδικευμένων εργατών. Ένα σύνολο τακτικών που εξακολουθούν να χρησιμοποιούνται σήμερα από επιτυχημένους οργανισμούς. Η σύγχρονη μορφή διαχείρισης έργου, όπως χαρακτηρίζεται από τα εργαλεία που χρησιμοποιούνται, τις τεχνικές που εφαρμόζονται και τη γλώσσα που χρησιμοποιείται εμφανίστηκε για πρώτη φορά στις αρχές της δεκαετίας του 1950, μετά την οποία σημειώθηκε σημαντική ανάπτυξη για τη βελτίωση της γνώσης, της απόδοσης και διαχείριση έργων. Η σύγχρονη διαχείριση έργων ξεκίνησε με την ανάπτυξη της μεθόδου κρίσιμης διαδρομής (CPM) που αναπτύχθηκε το 1957 από την DuPont Corporation, η Τεχνική Αναθεώρησης Αξιολόγησης Προγράμματος (PERT) που αναπτύχθηκε το 1958 από το Ναυτικό των ΗΠΑ. Από το 1980 μέχρι τη δεκαετία του 2000, αναπτύχθηκαν εκδόσεις των μεθόδων που βασίζονται σε υπολογιστή με την εξέλιξη των υπολογιστών. Όταν η εποχή του Διαδικτύου απέκτησε δημοτικότητα στις αρχές της δεκαετίας του 2000, ξεκίνησε η ανάπτυξη εργαλείων που βασίζονται στο διαδίκτυο και αύξησε την αποτελεσματικότητα τόσο των έργων όσο και της διαχείρισης έργων. Ως μέρος του IoT, προέκυψε η εποχή της συνεργασίας, αλλάζοντας τους τρόπους επικοινωνίας και ενίσχυσε τις μικρομεσαίες επιχειρήσεις. Επιπλέον, η εσωτερική επικοινωνία άλλαξε από μονόδρομη σε αμφίδρομη, δίνοντας στις ομάδες έργου την ευκαιρία να επηρεάσουν την κατεύθυνση του έργου. Τον Φεβρουάριο του 2001, 17 προγραμματιστές λογισμικού συναντήθηκαν για να συζητήσουν μεθόδους ανάπτυξης λογισμικού. Δημοσίευσαν το Manifesto for Agile Software Development για να ορίσουν την προσέγγιση που είναι γνωστή με το ίδιο όνομα. Μερικοί από τους συντάκτες του μανιφέστου σχημάτισαν την

Agile Alliance, έναν μη κερδοσκοπικό οργανισμό που προωθεί την ανάπτυξη λογισμικού σύμφωνα με τις 12 βασικές αρχές του μανιφέστου. Αυτό ήταν ένα βήμα για την προώθηση της χρήσης των υπολογιστών στην πρακτική διαχείρισης έργων. (www.researchgate.net, wiki.doing-projects.org)

1.3 Αντικείμενο διπλωματικής

Η παρούσα διπλωματική εργασία πραγματεύεται την αποτελεσματική διαχείριση και διοίκηση έργων με την χρήση σύγχρονων τεχνικών και μεθόδων που έχουν αναπτυχθεί. Στόχο της εν λόγω εργασίας αποτελεί η παρουσίαση της προστιθέμενης αξίας που προσδίδουν στις επιχειρήσεις τα εργαλεία DevOps, καθώς και οι πλατφόρμες μέσα από τις οποίες λαμβάνει χώρα. Ιδιαίτερη έμφαση δίνεται στην πλατφόρμα Microsoft Azure DevOps, η οποία χρησιμοποιείται στην εταιρία Uni Systems MAE για την διεκπεραίωση των European Projects που αναλαμβάνει.

1.4 Οργάνωση κειμένου

Η παρούσα διπλωματική εργασία δομείται σε επτά κεφάλαια. Πιο συγκεκριμένα:

Κεφάλαιο 1 – Εισαγωγή

Κεφάλαιο 2 – Μεθοδολογία Διαχείρισης Έργων

Κεφάλαιο 3 – Εργαλεία και Πλατφόρμες DevOps

Κεφάλαιο 4 – Πλατφόρμα Microsoft Azure DevOps

Κεφάλαιο 5 – Μελέτη Περίπτωσης Uni Systems MAE

Κεφάλαιο 6 – Συμπεράσματα

Κεφάλαιο 7 – Βιβλιογραφία

2

Μεθοδολογία

διαχείρισης έργων

Οι μεθοδολογίες διαχείρισης έργων είναι ένα σύνολο κατευθυντήριων αρχών και διαδικασιών που χρησιμοποιούνται για το σχεδιασμό, τη διαχείριση και την εκτέλεση έργων. Η μεθοδολογία διαχείρισης έργου που επιλέγεται καθορίζει τον τρόπο ιεράρχησης και ολοκλήρωσης της εργασίας. Με τις ρίζες του σε έργα μηχανικής, κατασκευών και στρατιωτικής άμυνας, οι δραστηριότητες διαχείρισης έργων πραγματοποιούνται σε ad-hoc ή άτυπη βάση εδώ και χιλιάδες χρόνια. Ωστόσο, μόλις τη δεκαετία του 1950 άρχισαν να ορίζονται και να χρησιμοποιούνται ευρέως μεταξύ των οργανισμών επίσημες, πειθαρχημένες μεθοδολογίες διαχείρισης έργων. Αυτές οι προσεγγίσεις για τη διαχείριση της εργασίας έχουν να κάνουν με τον καθορισμό του καλύτερου τρόπου έναρξης, σχεδιασμού και εκτέλεσης έργων. Με την άνοδο τόσων πολλών διαφορετικών τύπων διαδικασιών διαχείρισης, γίνεται προφανές ότι λίγοι μπορούν να συμφωνήσουν σχετικά με το ποια μεθοδολογία είναι η καλύτερη. Συχνά, αυτό εξαρτάται από τον ίδιο τον τύπο του οργανισμού. Στο παρόν κεφάλαιο παρουσιάζονται ορισμένες μέθοδοι διαχείρισης έργων που είναι δημοφιλείς σήμερα.(www.workfront.com)

2.1 Μεθοδολογία Waterfall

Το μοντέλο καταρράκτη είναι μια κατανομή των δραστηριοτήτων του έργου σε γραμμικές διαδοχικές φάσεις, όπου κάθε φάση εξαρτάται από τα παραδοτέα

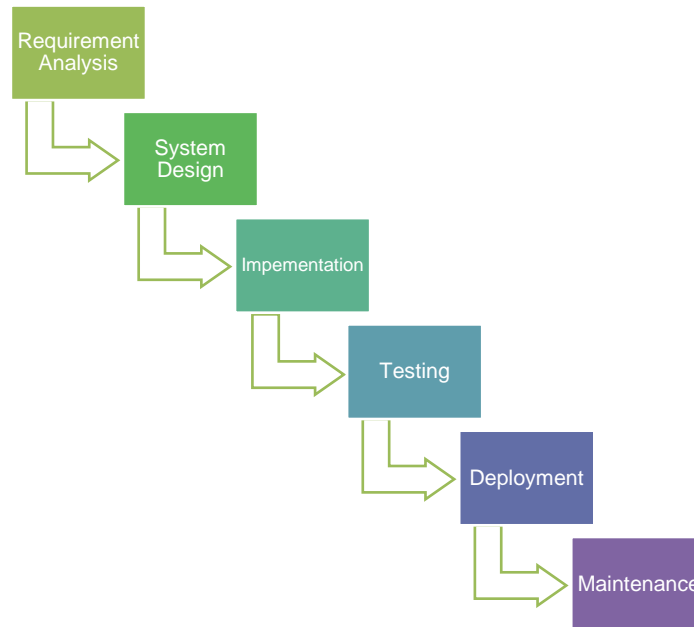
της προηγούμενης και αντιστοιχεί σε μια εξειδίκευση εργασιών. Η μεθοδολογία του καταρράκτη χρησιμοποιεί μια διαδοχική ή γραμμική προσέγγιση για την ανάπτυξη λογισμικού. Το έργο αναλύεται σε μια ακολουθία εργασιών. Μια πραγματική προσέγγιση καταρράκτη απαιτεί φάσεις που ολοκληρώνονται διαδοχικά και έχουν επίσημα κριτήρια εξόδου. Η προσέγγιση Waterfall καθιερώθηκε το 1970 από τον Winston W. Royce. Περιλαμβάνει έξι φάσεις διαχείρισης, όπου το καθένα απαιτεί ένα παραδοτέο από την προηγούμενη φάση για να προχωρήσει. Η μεθοδολογία Waterfall είναι ιδανική για έργα όπως η ανάπτυξη λογισμικού, όπου το τελικό αποτέλεσμα καθορίζεται ξεκάθαρα πριν από την έναρξη και είναι το καλύτερο για έργα που απαιτούν μεγάλη προβλεψιμότητα. Η μεθοδολογία του καταρράκτη απεικονίζεται συχνά με τη μορφή ενός διαγράμματος ροής ή ενός διαγράμματος Gantt. Οι έξι φάσεις της μεθοδολογίας Waterfall:

- **Προδιαγραφές Απαιτήσεων:** Είναι η αρχική διαδικασία σχεδιασμού κατά την οποία η ομάδα συλλέγει όσο το δυνατόν περισσότερες πληροφορίες για να εξασφαλίσει ένα επιτυχημένο έργο. Επειδή οι εργασίες στη μέθοδο καταρράκτη εξαρτώνται από προηγούμενα βήματα, απαιτεί πολλή προνοητικότητα. Αυτή η διαδικασία σχεδιασμού είναι ένα κρίσιμο μέρος του μοντέλου του καταρράκτη και εξαιτίας αυτού, το μεγαλύτερο μέρος του χρονοδιαγράμματος του έργου συχνά δαπανάται στον προγραμματισμό. Για να λειτουργήσει αυτή η μέθοδος συντάσσεται ένα λεπτομερές σχέδιο έργου που εξηγεί κάθε φάση του έργου. Αυτό περιλαμβάνει τα πάντα, από τους πόρους που χρειάζονται και τα συγκεκριμένα μέλη της ομάδας που εργάζονται για το έργο. Αυτό το έγγραφο αναφέρεται συνήθως ως έγγραφο απαιτήσεων έργου. Μέχρι το τέλος της φάσης των απαιτήσεων, θα πρέπει να υπάρχει ένα πολύ σαφές περίγραμμα του έργου από την αρχή μέχρι το τέλος, συμπεριλαμβανομένων:

- Κάθε στάδιο της διαδικασίας
- Ποιος εργάζεται σε κάθε στάδιο
- Βασικές εξαρτήσεις
- Απαιτούμενοι πόροι
- Χρονοδιάγραμμα για το πόσο χρόνο θα διαρκέσει κάθε στάδιο.

- **Σχεδιασμός συστήματος:** Σε μια διαδικασία ανάπτυξης λογισμικού, στη φάση σχεδιασμού η ομάδα έργου προσδιορίζει το υλικό που θα χρησιμοποιεί και άλλες λεπτομερείς πληροφορίες, όπως γλώσσες προγραμματισμού και διεπαφή χρήστη. Υπάρχουν δύο στάδια της φάσης σχεδιασμού του συστήματος: η φάση σχεδιασμού υψηλού επιπέδου και η φάση σχεδιασμού χαμηλού επιπέδου. Στη φάση σχεδιασμού υψηλού επιπέδου, η ομάδα δημιουργεί τον σκελετό του τρόπου λειτουργίας του λογισμικού και του τρόπου πρόσβασης στις πληροφορίες. Κατά τη φάση σχεδιασμού χαμηλού επιπέδου, η ομάδα κατασκευάζει τα πιο συγκεκριμένα μέρη του λογισμικού. Εάν η φάση σχεδιασμού υψηλού επιπέδου είναι ο σκελετός, η φάση σχεδιασμού χαμηλού επιπέδου είναι τα όργανα του έργου. Τα μέλη θα πρέπει να τεκμηριώσουν κάθε βήμα, ώστε η ομάδα να μπορεί να ανατρέξει σε αυτό που έγινε καθώς προχωρά το έργο.
- **Υλοποίηση:** Είναι το στάδιο όπου υλοποιούνται οι προδιαγραφές. Με βάση το έγγραφο απαιτήσεων στο πρώτο βήμα και τη διαδικασία σχεδιασμού συστήματος στο δεύτερο βήμα, η ομάδα ξεκινά την πλήρη διαδικασία ανάπτυξης για την κατασκευή του λογισμικού όπως περιγράφεται τόσο από τη φάση των απαιτήσεων όσο και από τη φάση σχεδιασμού συστήματος.
- **Δοκιμές:** Αυτό είναι το στάδιο στο οποίο η ομάδα ανάπτυξης παραδίδει το έργο στην ομάδα δοκιμών διασφάλισης ποιότητας. Οι QA testers αναζητούν τυχόν bugs ή σφάλματα που πρέπει να διορθωθούν πριν από την ανάπτυξη του έργου. Οι υπεύθυνοι δοκιμών πρέπει να τεκμηριώνουν με σαφήνεια όλα τα ζητήματα που βρίσκουν κατά τους ελέγχους. Σε περίπτωση που κάποιος άλλος προγραμματιστής αντιμετωπίσει παρόμοιο σφάλμα, μπορεί να ανατρέξει στην προηγούμενη τεκμηρίωση για να βοηθήσει στην επίλυση του προβλήματος.
- **Ανάπτυξη:** Για έργα ανάπτυξης, αυτό είναι το στάδιο στο οποίο το λογισμικό αναπτύσσεται στον τελικό χρήστη. Για άλλους κλάδους, αυτό είναι το τελικό παραδοτέο που λανσάρεται και παραδίδεται στους τελικούς πελάτες.

- **Συντήρηση:** Μόλις αναπτυχθεί ένα έργο, μπορεί να υπάρξουν περιπτώσεις όπου προκύπτει ένα νέο σφάλμα ή απαιτείται ενημέρωση λογισμικού. Η φάση συντήρησης αποτελεί μέρος του κύκλου διαδικασιών.



Σχήμα 1- waterfall methodology

Ένα από τα πλεονεκτήματα της προσέγγισης του καταρράκτη είναι ότι επιτρέπει μια σαφή διαδρομή από το σημείο Α στο σημείο Β. Τα έργα με έναν εύκολα καθορισμένο στόχο είναι κατάλληλα για τη μέθοδο του καταρράκτη, επειδή οι διαχειριστές έργων μπορούν να εργαστούν αντίστροφα από τον στόχο για να δημιουργήσουν μια σαφή και λεπτομερή διαδρομή με όλες τις απαραίτητες απαιτήσεις. Εάν το έργο δεν έχει περιορισμούς σε προϋπολογισμό ή χρόνο, τα μέλη της ομάδας μπορούν να αφιερώσουν όσο το δυνατόν περισσότερο χρόνο στη φάση των απαιτήσεων και του σχεδιασμού του συστήματος. Μπορούν να τροποποιήσουν και να προσαρμόσουν τις ανάγκες του έργου όσο θέλουν μέχρι να καταλήξουν σε ένα καλά μελετημένο και καθορισμένο σχέδιο έργου. Η δημιουργία επαναλαμβανόμενων διαδικασιών καθιστά επίσης εύκολη την εκπαίδευση νέων μελών της ομάδας σχετικά με το τι ακριβώς πρέπει να γίνει σε παρόμοια έργα. Αυτό καθιστά τη διαδικασία καταρράκτη μια αποτελεσματική προσέγγιση στη διαχείριση έργων για την τυποποίηση των διαδικασιών.

Από την άλλη, η μεθοδολογία του καταρράκτη είναι γραμμική από τη φύση της, επομένως εάν μια εργασία καθυστερήσει, ολόκληρο το χρονοδιάγραμμα μετατοπίζεται. Μία από τις μεγαλύτερες προκλήσεις είναι ότι είναι δύσκολο να επανέλθει σε μια φάση αφού έχει ήδη ολοκληρωθεί. Εάν γίνει ένα λάθος νωρίς στη διαδικασία, μπορεί να είναι δύσκολο να επιστρέψει η ομάδα και να το διορθώσει καθώς δεν αφήνει χώρο για επανάληψη ή αναζήτηση της καλύτερης λύσης. (asana.com/resources/waterfall-project-management-methodology)

2.2 Μεθοδολογία PM²

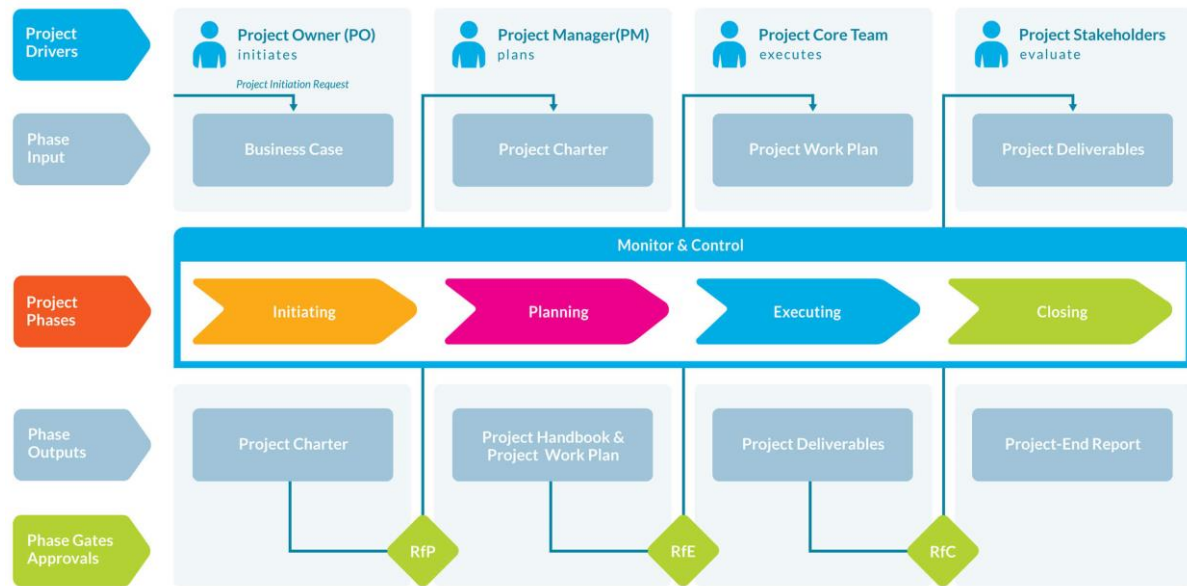


Το PM² είναι μια εύκολη στην εφαρμογή μεθοδολογία που οι ομάδες έργου μπορούν να προσαρμόσουν στις συγκεκριμένες ανάγκες τους. Όλα τα σχετικά υλικά (οδηγός μεθοδολογίας, πρότυπα έργων, εκπαιδευτικό υλικό) είναι ανοιχτού κώδικα. Το PM² χρησιμοποιείται με επιτυχία σε ολόκληρη την ΕΕ σε διάφορα έργα και επιτρέπει στις ομάδες του έργου να επικεντρωθούν στην παροχή αξίας και θετικού αντίκτυπου στους δικαιούχους και τους πελάτες τους. Στόχος είναι να παρέχουν ανοιχτή πρόσβαση στη Μεθοδολογία Διαχείρισης Έργων PM² και στη συνολική Προσφορά PM² σε όλα τα θεσμικά όργανα της Ευρωπαϊκής Ένωσης, τους Ευρωπαϊκούς Φορείς, τους Παρόχους Υπηρεσιών, τις Δημόσιες Διοικήσεις στα κράτη μέλη και τον ευρωπαϊκό ιδιωτικό τομέα επίσης. Αυτό θα επιτρέψει αυξημένη αποτελεσματικότητα στη διαχείριση και την επικοινωνία των εργασιών του έργου που εξυπηρετούν τους στόχους της Ευρωπαϊκής Ένωσης και τις ανάγκες των κρατών μελών και των πολιτών της ΕΕ. Η Μεθοδολογία PM² βασίζεται στις βέλτιστες πρακτικές Διαχείρισης Έργου και υποστηρίζεται από τέσσερις πυλώνες:

- Ένα μοντέλο διακυβέρνησης έργου (δηλαδή Ρόλοι & Αρμοδιότητες)
- Κύκλος ζωής έργου (δηλαδή Φάσεις Έργου)
- Ένα σύνολο διαδικασιών (δηλαδή δραστηριότητες διαχείρισης έργου)
- Ένα σύνολο artifacts (δηλαδή πρότυπα και οδηγίες).

Κάθε έργο έχει μια αρχή και ένα τέλος: ο κύκλος ζωής του έχει αναγνωρίσιμα σημεία έναρξης και λήξης, τα οποία μπορεί να είναι συνδεδεμένα με μια χρονική

κλίμακα. Ο κύκλος ζωής του έργου περιλαμβάνει όλες τις δραστηριότητες του έργου από το σημείο έναρξης μέχρι την τελική ολοκλήρωση του έργου. Ο κύκλος ζωής του έργου PM² έχει τέσσερις φάσεις. Κάθε φάση αντιπροσωπεύει μια χρονική περίοδο στη ζωή του έργου κατά τη διάρκεια των οποίων εκτελούνται παρόμοιες δραστηριότητες (π.χ. δραστηριότητες τύπου προγραμματισμού «αιχμής» όσον αφορά την προσπάθεια κατά τη φάση σχεδιασμού, κ.λπ.). Οι διεπαφές μεταξύ των φάσεων δεν είναι σχεδόν ποτέ ξεκάθαρες, διαχωρίζονται καθώς οι δραστηριότητες που σχετίζονται με μια συγκεκριμένη φάση (π.χ. δραστηριότητες προγραμματισμού) συνεχίζουν να εκτελούνται κατά τη διάρκεια της επόμενης φάσης (π.χ. φάση εκτέλεσης). Επομένως, οι φάσεις ορίζονται συμβατικά. Στην αρχή, τα έργα επικεντρώνονται στην έναρξη και τον προγραμματισμό δραστηριοτήτων, στη μέση στην υλοποίηση, στην παρακολούθηση και στον έλεγχο των δραστηριοτήτων και, στο τέλος, στις δραστηριότητες αποδοχής, μετάβασης και κλεισίματος. Οι οδηγοί έργου διαφέρουν από φάση σε φάση σε ένα έργο PM². Ο Ιδιοκτήτης Έργου (PO) είναι ο κύριος οδηγός κατά την έναρξη του έργου (εκκινεί το έργο και λογοδοτεί για όλα τα έγγραφα), ενώ ο Διαχειριστής Έργου (PM) καθοδηγεί τη Φάση Σχεδιασμού (είναι υπεύθυνος για τον συντονισμό της παράδοσης όλων σχεδία έργων). Η Βασική Ομάδα Έργου (PCT) καθοδηγεί την εκτέλεση του έργου και τα παραδοτέα του έργου, ενώ τα ενδιαφερόμενα μέρη του έργου είναι οι κύριοι μοχλοί της Φάσης Κλεισίματος αξιολογούν τα παραδοτέα του έργου και τη συνολική απόδοση. Το παρακάτω σχήμα δείχνει τους οδηγούς PM² και τα βασικά artifacts ως εισόδους και εξόδους των φάσεων του έργου. (europa.eu/pm2/home_en)



Σχήμα 2-PM2 methodology (europa.eu/pm2/home_en)

2.3 Μεθοδολογία Lean

Η Lean είναι μία από τις σημαντικότερες μεθοδολογίες διαχείρισης έργων για περισσότερα από εβδομήντα χρόνια. Χρονολογείται στα τέλη της δεκαετίας του 1940, όταν το Σύστημα Παραγωγής της Toyota (TPS) έθεσε τα θεμέλια της λιτής κατασκευής. Στόχος της εταιρείας ήταν η μείωση των διεργασιών που δεν συνέβαλαν στο τελικό προϊόν ως προς την αξία του. Η αναφερόμενη μέθοδος βελτίωσης του προϊόντος αναφέρεται τώρα ως η εξάλειψη των απορριμμάτων και αντιπροσωπεύει μία από τις βασικές αρχές της μεθοδολογίας Lean. Με την εφαρμογή του, η TPS κατάφερε να επιτύχει αξιοσημείωτες βελτιώσεις στην αποτελεσματικότητα, την παραγωγικότητα, καθώς και την αξία των υπηρεσιών της. Παρόλο που η εφαρμογή της μεθοδολογίας Lean μπορεί να ανιχνευθεί στα τέλη της δεκαετίας του 1940, η Lean προσέγγιση στη διαχείριση έργων δεν είχε σωστή ονομασία μέχρι τα τέλη της δεκαετίας του 1980. Ήταν ο John Krafcik, ο σημερινός Διευθύνων Σύμβουλος του έργου Waymo, που επινόησε τον όρο Lean το 1988. Ασχολήθηκε με το θέμα στο άρθρο του με τίτλο «Triumph of the Lean Production System». Μόλις τα οφέλη του έγιναν εμφανή, το μοντέλο Lean άρχισε να επεκτείνεται. Η ανάπτυξη λογισμικού ήταν ο πρώτος τομέας εκτός της κατασκευής στον οποίο εφαρμόστηκε το μοντέλο Lean. Το βιβλίο με τίτλο

«Lean Software Development: An Agile Toolkit», που γράφτηκε από τους Mary και Tom Poppendieck και δημοσιεύτηκε το 2003, περιγράφει την εφαρμογή των αρχών Lean μεθοδολογίας στην ανάπτυξη λογισμικού. Με τα χρόνια, η ανάπτυξη λογισμικού Lean έχει γίνει μια από τις πιο ευνοημένες μεθόδους ανάπτυξης λογισμικού. Βασίζεται στις ακόλουθες επτά αρχές :

- **Eliminate Waste**, στην ανάπτυξη εφαρμογών, waste είναι οτιδήποτε δεν προσδίδει επιχειρηματική αξία στον πελάτη και δεν βελτιώνει την ποιότητα του προϊόντος που αναπτύσσεται ούτε επιταχύνει τον χρόνο κυκλοφορίας του έργου.
- **Amplify Learning**, προκειμένου η ομάδα να αναπτύξει ένα σύστημα που θα φέρει επιχειρηματική αξία στον πελάτη, πρέπει να έχει ένα ευρύ φάσμα δεξιοτήτων. Η ομάδα πρέπει να συσσωρεύσει γνώση και να τη μοιραστεί, για παράδειγμα, με τη μορφή ανασκόπησης στο τέλος της επανάληψης.
- **Decide as Late as Possible**, η κύρια ιδέα εδώ είναι η ομάδα να περιμένει μέχρι την τελευταία στιγμή πριν λάβει οποιαδήποτε απόφαση, ειδικά μη αναστρέψιμη. Αυτό ισχύει ιδιαίτερα για αποφάσεις που μπορούν να έχουν σημαντικό αντίκτυπο στην επιτυχία της ανάπτυξης. Όλες οι αποφάσεις πρέπει να υποστηρίζονται από αναλυτικά δεδομένα και αποτελέσματα παρακολούθησης της διαδικασίας, διαφορετικά η ομάδα κινδυνεύει να απορροφηθεί από πάρα πολλές αλλαγές και μπορεί να ξεχάσει τον κύριο σκοπό του έργου. Όσο αργότερα ληφθεί μια απόφαση, τόσο περισσότερες δεξιότητες και κατανόηση έχει η ομάδα και τόσο λιγότερα θα χρειαστεί να επαναλάβει αργότερα.
- **Empower the Team**, προκειμένου οι άνθρωποι να αναλάβουν την ευθύνη, να έχουν κίνητρα και να εργαστούν ως μια σταθερή ομάδα, θα πρέπει να γνωρίζουν τη συμβολή τους στο προϊόν που αναπτύσσεται. Είναι απαραίτητο να δημιουργηθούν συνθήκες στις οποίες κάθε άτομο μπορεί να επικεντρωθεί στην τρέχουσα επιχειρηματική αποστολή. Ο ανθρώπινος παράγοντας είναι ένα από τα πιο σημαντικά στοιχεία στην επιτυχημένη ανάπτυξη.

- **Build Integrity In**, σύμφωνα με τις αρχές της lean μεθοδολογίας, ένα πρόβλημα μπορεί είτε να βρεθεί μετά την εμφάνισή του είτε μπορεί η ομάδα να εξαλείψει τις αιτίες που οδηγούν σε πρόβλημα εκ των προτέρων. Η ομάδα θα πρέπει να κινείται με μικρά βήματα και να ελέγχει την ποιότητα μετά από κάθε βήμα
- **Deliver as Fast as Possible**, αυτή είναι η βάση της επαναληπτικής ανάπτυξης. Όσο πιο γρήγορα η ομάδα δείξει τη βάση στον πελάτη, τόσο πιο γρήγορα θα λάβει τα σχόλιά του, επομένως, θα λάβει το προϊόν με τις απαραίτητες βελτιώσεις πολύ νωρίτερα. Μια ενδιαφέρουσα επιχειρηματική λειτουργία που τέθηκε σε παραγωγή μέσα σε λίγους μήνες μπορεί τελικά να αποδειχθεί εντελώς άχρηστη. Αλλά αν κυκλοφορούσε μέσα σε δύο εβδομάδες, θα μπορούσε να αποδειχθεί επωφελής για τον πελάτη.
- **Optimize the Whole**, ο κύριος τρόπος επίλυσης προβλημάτων είναι η διάσπασή τους σε μικρότερα ζητήματα και η συνεχής εξάλειψη των αιτιών εμφάνισής τους. Η ομάδα θα πρέπει να έχει μια καλή συνολική κατανόηση της τρέχουσας διαδικασίας ανάπτυξης, της ιδέας και της στρατηγικής του προϊόντος που αναπτύσσεται.
(startinfinity.com/project-management-methodologies/lean),
(lvivivity.com/lean-development-key-principles)



Σχήμα 3- Lean methodology

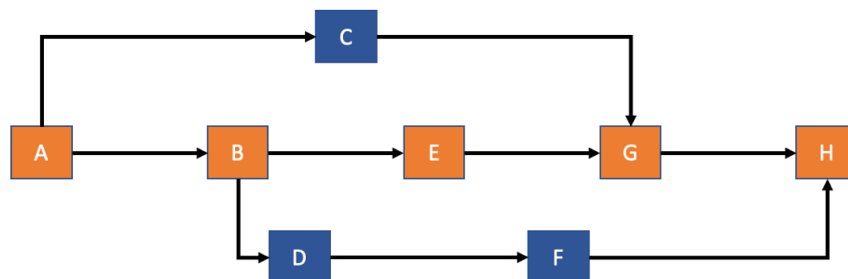
2.4 Μεθοδολογία *Critical path*

Η μέθοδος κρίσιμης διαδρομής (CPM) είναι μια τεχνική που χρησιμοποιείται από τους υπεύθυνους του έργου για τη δημιουργία ενός χρονοδιαγράμματος και την εκτίμηση της συνολικής διάρκειας του. Η μέθοδος της κρίσιμης διαδρομής αναπτύχθηκε στα τέλη της δεκαετίας του 1950 από τους Morgan R. Walker και James E. Kelley. Οι απαρχές της μεθόδου της κρίσιμης διαδρομής συνδέονται στενά με την Τεχνική Αξιολόγησης και Αναθεώρησης Προγράμματος (PERT), μια παρόμοια μέθοδος που χρησιμοποιείται συνήθως σε συνδυασμό με το CPM. Η μέθοδος CPM, γνωστή και ως ανάλυση κρίσιμης διαδρομής (CPA), συνίσταται στη χρήση ενός διαγράμματος δικτύου για την οπτική αναπαράσταση των ακολουθιών εργασιών που απαιτούνται για την ολοκλήρωση ενός έργου. Μόλις καθοριστούν αυτές οι ακολουθίες εργασιών ή διαδρομές, η διάρκειά τους υπολογίζεται για να προσδιοριστεί η κρίσιμη διαδρομή, η οποία καθορίζει τη συνολική διάρκεια του έργου. Στη διαχείριση έργου, η κρίσιμη διαδρομή είναι η μεγαλύτερη ακολουθία εργασιών που πρέπει να πραγματοποιηθεί για να ολοκληρωθεί ένα έργο. Οι εργασίες στο κρίσιμο μονοπάτι ονομάζονται κρίσιμες δραστηριότητες γιατί εάν καθυστερήσουν, θα καθυστερήσει η ολοκλήρωση του έργου. Η εύρεση της

κρίσιμης διαδρομής είναι πολύ σημαντική για τους διαχειριστές έργων επειδή τους επιτρέπει:

- Να υπολογίζουν με ακρίβεια τη συνολική διάρκεια του έργου
- Να προσδιορίζουν τις εξαρτήσεις των εργασιών, τους περιορισμούς πόρων και τους κινδύνους του έργου
- Να δώσουν προτεραιότητα στις εργασίες και να δημιουργήσουν ρεαλιστικά χρονοδιαγράμματα έργων
- Να παρακολουθήσουν την πρόοδο του έργου και να μετρήσουν τη διακύμανση του χρονοδιαγράμματος
- Να χρησιμοποιήσουν τεχνικές συμπίεσης χρονοδιαγράμματος, όπως η διάρκεια σύγκρουσης ή η γρήγορη παρακολούθηση

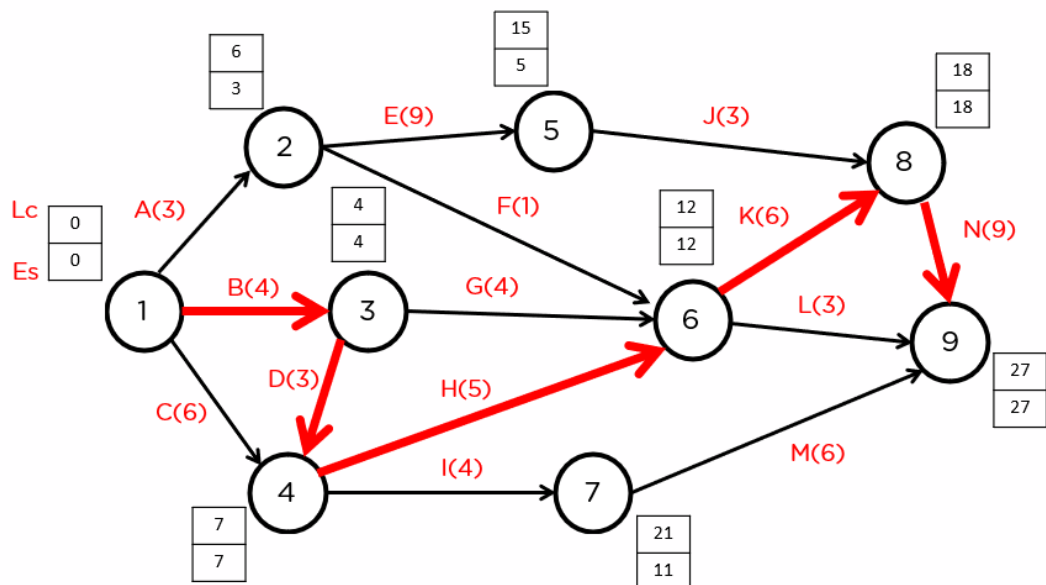
Για να βρουν την κρίσιμη διαδρομή, οι διαχειριστές έργων χρησιμοποιούν τον αλγόριθμο της μεθόδου κρίσιμης διαδρομής (CPM) για να ορίσουν τον ελάχιστο χρόνο που απαιτείται για την ολοκλήρωση κάθε εργασίας με τη μικρότερη καθυστέρηση.



Σχήμα 4- Critical path methodology (www.fabianjost.com)

Σήμερα η κρίσιμη διαδρομή μπορεί να υπολογιστεί αυτόματα με λογισμικό προγραμματισμού έργου εξοπλισμένο με διαγράμματα Gantt, γεγονός που καθιστά την όλη μέθοδο CPM πολύ πιο εύκολη. Ορισμένες βασικές έννοιες CPM:

- Πρώιμος χρόνος έναρξης (ES): Αυτή είναι απλώς η πρώτη φορά που μπορεί να ξεκινήσει μια εργασία στο έργο. Δεν μπορεί να προσδιοριστεί χωρίς να είναι γνωστή η εξάρτηση εργασιών.
- Τελευταία ώρα έναρξης (LS): Αυτή είναι η τελευταία στιγμή κατά την οποία μπορεί να ξεκινήσει μια εργασία προτού οδηγήσει στην καθυστέρηση του χρονοδιαγράμματος του έργου.
- Πρώτος χρόνος τερματισμού (EF): Το νωρίτερο που μπορεί να ολοκληρωθεί μια δραστηριότητα, με βάση τη διάρκειά της και την πρώιμη ώρα έναρξης της.
- Τελευταίος χρόνος τερματισμού (LF): Το τελευταίο που μπορεί να ολοκληρωθεί μια δραστηριότητα, με βάση τη διάρκειά της και την τελευταία ώρα έναρξης.
- Float: Γνωστό και ως slack, το float είναι ένας όρος που περιγράφει πόσο χρόνο μπορεί να καθυστερήσει μια εργασία προτού επηρεάσει τη σειρά εργασιών της και το χρονοδιάγραμμα του έργου. Οι εργασίες στο κρίσιμο μονοπάτι έχουν μηδενικό float, επειδή δεν μπορούν να καθυστερήσουν. (www.projectmanager.com/critical-path-method)



Σχήμα 5- critical path with values (www.simplilearn.com)

2.5 Μεθοδολογία PRINCE2



Το PRINCE2 είναι μια μεθοδολογία διαχείρισης έργου που χρησιμοποιείται κυρίως στο Ηνωμένο Βασίλειο, την Αυστραλία και τις ευρωπαϊκές χώρες. Το "PRINCE2" σημαίνει Projects IN Controlled Environments, επειδή αυτή η μέθοδος εστιάζει στη διαχείριση πόρων και κινδύνων διαιρώντας τα έργα σε μικρότερα στάδια, ορίζοντας σαφείς ρόλους και ευθύνες. Οργανώνοντας το έργο σε λογικά βήματα, το PRINCE2 απαιτεί ένα πλαίσιο διαχείρισης έργου που έχει ένα οργανωμένο και ελεγχόμενο σχέδιο έργου πριν από την έναρξη, που διατηρεί την οργάνωσή του κατά τη διάρκεια του κύκλου ζωής του έργου. Ο έλεγχος επιτυγχάνεται με επτά διαδικασίες PRINCE2. Αυτές οι διαδικασίες διαχείρισης έργου καλύπτουν όλες τις δραστηριότητες που συνθέτουν ένα έργο, από την αρχή μέχρι το τέλος. Καθιερώθηκε για πρώτη φορά το 1989 από την Κεντρική Υπηρεσία Υπολογιστών και Τηλεπικοινωνιών (CCTA), η μέθοδος PRINCE2 βασίστηκε αρχικά στο PROMPT, ένα άλλο ακρωνύμιο που σημαίνει Project Resource Organization Management Planning Technique, το οποίο δημιουργήθηκε το 1975 και χρησιμοποιήθηκε από την κυβέρνηση του Ηνωμένου Βασιλείου για έργα πληροφοριακών συστημάτων. Το PRINCE2 δημοσιεύτηκε τελικά ως μεθοδολογία διαχείρισης έργου το 1996 και είναι δημόσια.

Υπάρχουν επτά αρχές που αποτελούν τη βάση για τη μεθοδολογία PRINCE2. Πρέπει να τηρούνται όλα για να θεωρηθεί η μεθοδολογία διαχείρισης έργου PRINCE2. Αυτές οι επτά αρχές αποτελούν ένα πλαίσιο για την νοοτροπία PRINCE2 και είναι οι εξής:

- **Συνεχιζόμενη επιχειρησιακή αιτιολόγηση**, το πιο σημαντικό έγγραφο στο PRINCE2 είναι η επιχειρηματική υπόθεση, η οποία εξετάζεται και αναθεωρείται καθ' όλη τη διάρκεια του κύκλου ζωής του έργου. Αυτό γίνεται για να διασφαλιστεί ότι το έργο παραμένει βιώσιμο. Αν όχι, το έργο τελειώνει.
- **Συνεχής Μάθηση**, υπάρχει ένα αρχείο καταγραφής μαθημάτων που επισυνάπτεται σε κάθε έργο που λειτουργεί ως αποθήκη για τα

διδάγματα που αντλήθηκαν από προηγούμενα έργα, ώστε να αποφευχθεί η επανάληψη λαθών του παρελθόντος. Το αρχείο καταγραφής μαθήματος αναφέρεται σε όλο το έργο.

- **Σαφής διάκριση ρόλων και ευθυνών**, σε κάθε στάδιο του έργου, ο διαχειριστής του έργου θα πρέπει να ορίζει με σαφήνεια τους ρόλους και τις ευθύνες για τα άτομα και τις ομάδες. Έτσι όλοι είναι ξεκάθαροι για το τι είναι υπεύθυνοι και τι αναμένεται από αυτούς και παρέχεται μια σαφή μέτρηση απόδοσης για όλα τα μέλη της ομάδας.
- **Προσέγγιση κατά φάση**, στη μεθοδολογία PRINCE2, κάθε έργο χωρίζεται σε διακριτές φάσεις διαχείρισης έργου και με την ολοκλήρωση κάθε φάσης σημειώνονται τα αποτελέσματα της φάσης και τα διδάγματα που αντλήθηκαν. Αυτό βοηθά στην παρακολούθηση της κατεύθυνσης προς την οποία οδεύει το έργο
- **Διαχείριση κατ' εξαίρεση**, υπάρχουν ανοχές για κάθε στόχο του έργου. Αυτό δημιουργεί όρια και καθορίζει τον τρόπο με τον οποίο εκχωρείται η εξουσία. Εάν γίνει υπέρβαση της ανοχής, η ομάδα διαχείρισης πρέπει να αποφασίσει εάν αυτή η ανοχή θα επαναπροσδιοριστεί.
- **Εστίαση στην ποιότητα των προϊόντων**, οι ομάδες πρέπει να διασφαλίζουν ότι δεν υπάρχει κενό μεταξύ των παραδοτέων του έργου και των απαιτήσεων του έργου. Για αυτό, η ποιότητα των παραδοτέων πρέπει να ελέγχεται και να μετράται συνεχώς από τις ομάδες.
- **Προσαρμοσμένη προσέγγιση για έργα**, με τη μεθοδολογία PRINCE2, η προσέγγιση για διαφορετικά έργα μπορεί να προσαρμοστεί ανάλογα με τους διαθέσιμους πόρους και το χρόνο. Μπορούν να προσαρμοστούν διαφορετικές παράμετροι, όπως ο αριθμός των μελών της ομάδας, το μέγεθος της επίβλεψης και του σχεδιασμού για κάθε φάση του έργου.

Η μεθοδολογία διαχείρισης έργου PRINCE2 χρησιμοποιεί επτά διαδικασίες για τη διαχείριση έργων. Καθώς το PRINCE2 είναι μια μέθοδος ελεγχόμενου περιβάλλοντος, ο ρόλος του διαχειριστή έργου, του συμβουλίου του έργου και του πελάτη καθορίζονται έτσι ώστε όλοι να βρίσκονται στην ίδια σελίδα.

- **Starting up a project**, εδώ καθορίζεται εάν το έργο είναι βιώσιμο. Για να γίνει αυτό, πρέπει να δημιουργηθεί μια σύνοψη έργου, μια επιχειρηματική υπόθεση και ένα λεπτομερές σχέδιο σταδίου. Οι δραστηριότητες αυτής της διαδικασίας περιλαμβάνουν το έναυσμα, το οποίο είναι ένα έγγραφο υψηλού επιπέδου που δηλώνει την εντολή του έργου.
- **Directing a project**, εδώ αναλύεται η βιωσιμότητα του έργου και το έργο ανατίθεται σε διαχειριστή έργου
- **Initiating a project**, κατά τη διάρκεια αυτής της διαδικασίας, ορίζονται οι πτυχές του έργου δηλαδή το εύρος, το κόστος, τα χρονοδιαγράμματα, ο κίνδυνος, η ποιότητα και τα οφέλη.
- **Stage control**, το έργο υποδιαιρείται σε στάδια και ορίζονται ομάδες για κάθε στάδιο. Ο διαχειριστής έργου παρακολουθεί την απόδοση και τα ζητήματα κάθε σταδίου, αναθέτει καθήκοντα, παρακολουθεί την εργασία, ασχολείται με οποιαδήποτε ζητήματα προκύπτουν και αναφέρει την πρόοδό του στο συμβούλιο του έργου. Οι δραστηριότητες σε αυτήν τη διαδικασία περιλαμβάνουν την εξουσιοδότηση ενός πακέτου εργασίας με την ομάδα, τον έλεγχο της κατάστασης και της προόδου του και τον έλεγχο της ποιότητάς του όταν ολοκληρωθεί. Πρέπει επίσης να επανεξετάσει και να συγκρίνει την πρόοδο με το σχέδιο του έργου, να συλλάβει τυχόν ζητήματα και κινδύνους και να ενεργήσει για την επίλυσή τους.
- **Product Delivery Management**, Αυτή η διαδικασία διαχειρίζεται την παράδοση του προϊόντος του έργου. Ο διαχειριστής του έργου διασφαλίζει ότι δεν υπάρχει κενό μεταξύ των παραδοτέων και των προσδοκιών ποιότητας. Το συμβούλιο του έργου κάνει μια τελική αξιολόγηση και είτε εγκρίνει το προϊόν είτε ζητά αναθεωρήσεις
- **Managing Stage Boundaries**, υπάρχουν δύο μέρη σε αυτή τη διαδικασία: πρώτον, ο διαχειριστής έργου παρέχει στον πίνακα του έργου μια επισκόπηση της απόδοσης, ενημερώνει το σχέδιο του έργου και την επιχειρηματική υπόθεση και δημιουργεί ένα σχέδιο για το επόμενο στάδιο. Δεύτερον, οι πληροφορίες που παρέχονται από τον διαχειριστή του έργου θα βοηθήσουν το συμβούλιο του έργου να

επανεξετάσει το τρέχον στάδιο, να εγκρίνει το επόμενο και να επανεξετάσει το ενημερωμένο σχέδιο. Οι δραστηριότητες περιλαμβάνουν σχεδιασμό του επόμενου σταδίου και αναφορά για το τέλος της φάσης.

- **Closing a project**, Αυτή η διαδικασία αφορά τη διασφάλιση ότι το έργο πέτυχε τους στόχους και τους στόχους του εντός της προθεσμίας. Μερικές φορές οι διαχειριστές έργων προετοιμάζουν το προγραμματισμένο κλείσιμο και το πρόωρο κλείσιμο, αλλά αυτό δεν απαιτείται. Οι υποχρεωτικές δραστηριότητες περιλαμβάνουν την παράδοση του προϊόντος, την αξιολόγηση του έργου και τη σύσταση για το κλείσιμό του στο συμβούλιο του έργου για να κλείσει επίσημα. (www.projectmanager.com/blog/prince2-methodology), (kissflow.com/project/prince2-project-methodology))



Σχήμα 6- Prince2 method (www.best.cheaponline2022.ru)

2.6 Μεθοδολογία Agile

Οι μεθοδολογίες Agile είναι προσεγγίσεις στην ανάπτυξη προϊόντων που ευθυγραμμίζονται με τις αξίες και τις αρχές που περιγράφονται στο Agile Manifesto για την ανάπτυξη λογισμικού. Οι μεθοδολογίες Agile στοχεύουν στην παράδοση του σωστού προϊόντος, με σταδιακή και συχνή παράδοση μικρών τμημάτων λειτουργικότητας, μέσω μικρών διαλειτουργικών ομάδων αυτοοργάνωσης, επιτρέποντας τη συχνή ανατροφοδότηση των πελατών και τη διόρθωση πορείας, όπως απαιτείται. Με αυτόν τον τρόπο, η Agile στοχεύει να διορθώσει τις προκλήσεις που αντιμετωπίζουν οι παραδοσιακές προσεγγίσεις "καταρράκτη" της παράδοσης μεγάλων προϊόντων σε μεγάλες

χρονικές περιόδους, κατά τις οποίες οι απαιτήσεις των πελατών αλλάζουν συχνά, με αποτέλεσμα να παραδίδονται λάθος προϊόντα. Στο μεγαλύτερο μέρος της σύντομης ιστορίας του (από το 1999-2000), το "Agile" ήταν κυρίως μια προσέγγιση για την ανάπτυξη λογισμικού και τα έργα ανάπτυξης εφαρμογών πληροφορικής. Έκτοτε, ωστόσο, επεκτείνεται πλέον και σε άλλους τομείς, ιδίως στους κλάδους της γνώσης και των υπηρεσιών. Είτε πρόκειται για IT είτε για ανάπτυξη λογισμικού ή για οποιοδήποτε άλλο τομέα όπου υπάρχει ροή εργασιών και παράδοση προϊόντων εργασίας, εφαρμόζονται οι μέθοδοι Agile. Οι ευέλικτες μέθοδοι προσπαθούν να μεγιστοποιήσουν την παράδοση αξίας στον πελάτη και να ελαχιστοποιήσουν τον κίνδυνο κατασκευής προϊόντων που δεν ανταποκρίνονται –ή δεν ανταποκρίνονται πλέον– στις ανάγκες της αγοράς ή των πελατών. Το κάνουν αυτό διασπώντας τον παραδοσιακά μακρύ κύκλο παράδοσης (συνήθως των παλαιών "μεθόδων καταρράκτη") σε μικρότερες περιόδους, που ονομάζονται σπριντ ή επαναλήψεις. Η επανάληψη παρέχει τον ρυθμό για την παράδοση ενός λειτουργικού προϊόντος στον πελάτη, τη λήψη σχολίων και την πραγματοποίηση αλλαγών με βάση την ανατροφοδότηση. Έτσι, οι μέθοδοι Agile προσπάθησαν να μειώσουν τους χρόνους παράδοσης (παράδοση νωρίς, παράδοση συχνά) για να διασφαλίσουν ότι μικρότερα κάθετα κομμάτια του προϊόντος φτάνουν στην αγορά, επιτρέποντας στους πελάτες να παρέχουν έγκαιρα σχόλια και να διασφαλίσουν ότι το προϊόν που θα παραλάβουν τελικά ανταποκρίνεται στις ανάγκες τους.

Το Agile έχει γίνει ένας όρος-ομπρέλα για μια ποικιλία σχεδιασμού, διαχείρισης και τεχνικών μεθόδων και διαδικασιών για τη διαχείριση έργων, την ανάπτυξη λογισμικού και άλλων προϊόντων και υπηρεσιών με επαναληπτικό τρόπο. Αυτές οι μέθοδοι περιλαμβάνουν το Scrum, μακράν την πιο διαδεδομένη και δημοφιλή μέθοδο για λογισμικό, τα XP (eXtreme Programming ή Paired Programming) και πιο πρόσφατα το Kanban. Οι ευέλικτες μέθοδοι περιλαμβάνουν επίσης τεχνικές πρακτικές – οι περισσότερες από τις οποίες εμπίπτουν στον γενικό όρο DevOps – που επιτρέπουν τη Δοκιμαστική Αυτοματοποίηση, τη Συνεχή Ενσωμάτωση/ Συνεχή Παράδοση/ Ανάπτυξη (CI/CD) και συνολικά, έναν συνεχώς συρρικνωμένο κύκλο παράδοσης για λογισμικό και άλλα προϊόντα και

υπηρεσίες. Η χρήση του Agile ως προσέγγιση στη διαχείριση έργων έχει αυξηθεί δραματικά τα τελευταία χρόνια. Η Gartner προβλέπει ότι οι ευέλικτες μέθοδοι ανάπτυξης θα χρησιμοποιηθούν σύντομα στο 80% όλων των έργων ανάπτυξης λογισμικού. Το Agile Manifesto θέσπισε επίσης 12 αρχές σε σχέση με την ανάπτυξη λογισμικού και αργότερα αναδιαμορφώθηκε ώστε να ταιριάζει σε μια ευρύτερη οπτική γωνία των χρηστών:

- Ικανοποίηση των πελατών
- Έγκαιρη και συνεχής παράδοση
- Αποδοχή της αλλαγής
- Συχνή παράδοση
- Συνεργασία επιχειρήσεων και προγραμματιστών
- Άτομα με κίνητρα
- Συνομιλία πρόσωπο με πρόσωπο
- Λειτουργικά προϊόντα
- Τεχνική αριστεία
- Απλότητα
- Αυτοοργανωμένες ομάδες
- Ρύθμιση, προβληματισμός και προσαρμογή

(www.digite.com/agile/agile-methodology/,

www.businessnewsdaily.com/4987-what-is-agile-scrum-methodology.html)



Σχήμα 7- Agile (www.researchgate.net)

2.6.1 Scrum

Το Scrum είναι μια ευέλικτη μεθοδολογία ανάπτυξης που χρησιμοποιείται στην ανάπτυξη Λογισμικού που βασίζεται σε επαναληπτικές και αυξητικές διαδικασίες. Το Scrum είναι προσαρμόσιμο, γρήγορο, ευέλικτο και αποτελεσματικό πλαίσιο που έχει σχεδιαστεί για να προσφέρει αξία στον πελάτη σε όλη την ανάπτυξη του έργου. Πρωταρχικός στόχος του Scrum είναι να ικανοποιήσει τις ανάγκες του πελάτη μέσα από ένα περιβάλλον διαφάνειας στην επικοινωνία, συλλογικής ευθύνης και συνεχούς προόδου. Η ανάπτυξη ξεκινά από μια γενική ιδέα για το τι πρέπει να κατασκευαστεί, εκπονώντας μια λίστα χαρακτηριστικών κατά που θέλει να αποκτήσει ο ιδιοκτήτης του προϊόντος. Η ιστορία του Scrum μπορεί να ανιχνευθεί πίσω στο 1986 στο άρθρο του Harvard Business Review (HBR) με τίτλο «The New Product Development Game» από τους Hirotaka Takeuchi & Ikujiro Nonaka. Αυτό το άρθρο περιγράφει πώς εταιρείες όπως η Honda, η Canon και η Fuji-Xerox παράγουν νέα προϊόντα παγκοσμίως χρησιμοποιώντας μια επεκτάσιμη και βασισμένη σε ομάδες προσέγγιση για την ανάπτυξη προϊόντων. Αυτή η προσέγγιση τονίζει τη σημασία της ενδυνάμωσης των αυτοοργανωμένων ομάδων. Το άρθρο ήταν μια επιρροή για την ανάπτυξη πολλών από τις έννοιες που γέννησαν αυτό που τώρα ονομάζουμε Scrum. Το Scrum είναι ένας όρος που προέρχεται από το Rugby, ο οποίος αναφέρεται στο πώς το παιχνίδι ξεκινά ξανά μετά από ένα φάουλ ή όταν η μπάλα έχει φύγει από το παιχνίδι. Το 1993, ο Jeff Sutherland και η ομάδα του στην Easel Corporation δημιούργησαν τη διαδικασία Scrum για χρήση σε διαδικασίες ανάπτυξης λογισμικού συνδυάζοντας τις έννοιες του άρθρου του 1986 με τις έννοιες της αντικειμενοστραφούς ανάπτυξης, του εμπειρικού ελέγχου διεργασιών, της επαναληπτικής ανάπτυξης και των επαυξητικών, διεργασιών λογισμικού. και τη βελτίωση της παραγωγικότητας, καθώς και την ανάπτυξη πολύπλοκων και δυναμικών συστημάτων.

Το Scrum είναι μια εξέλιξη του Agile Management. Η μεθοδολογία Scrum βασίζεται σε ένα σύνολο πολύ καθορισμένων πρακτικών και ρόλων που πρέπει να εμπλέκονται κατά τη διαδικασία ανάπτυξης λογισμικού. Πρόκειται

για μια ευέλικτη μεθοδολογία που ανταμείβει την εφαρμογή των 12 ευέλικτων αρχών σε ένα πλαίσιο που συμφωνείται από όλα τα μέλη της ομάδας του προϊόντος. Το Scrum εκτελείται σε προσωρινά μπλοκ που είναι σύντομα και περιοδικά, που ονομάζονται Sprints, τα οποία συνήθως κυμαίνονται από 2 έως 4 εβδομάδες, που είναι ο όρος για ανατροφοδότηση και προβληματισμό. Κάθε Sprint είναι μια οντότητα από μόνη της, παρέχει δηλαδή ένα ολοκληρωμένο αποτέλεσμα, μια παραλλαγή του τελικού προϊόντος που πρέπει να μπορεί να παραδοθεί στον πελάτη με τη μικρότερη δυνατή προσπάθεια όταν ζητηθεί. Η διαδικασία έχει ως σημείο εκκίνησης, μια λίστα στόχων/απαιτήσεων που συνθέτουν το σχέδιο του έργου. Ο πελάτης του έργου είναι αυτός που δίνει προτεραιότητα σε αυτούς τους στόχους λαμβάνοντας υπόψη την ισορροπία της αξίας και του κόστους τους, με αυτόν τον τρόπο καθορίζονται οι επαναλήψεις και οι επακόλουθες παραδόσεις. Αφενός, η αγορά απαιτεί ποιότητα, γρήγορη παράδοση με χαμηλότερο κόστος, για τα οποία μια εταιρεία πρέπει να είναι πολύ ευέλικτη, για να επιτύχει σύντομους κύκλους ανάπτυξης που μπορούν να ικανοποιήσουν τη ζήτηση των πελατών χωρίς να υπονομεύσουν την ποιότητα του αποτελέσματος. Είναι μια πολύ εύκολη μεθοδολογία στην εφαρμογή και πολύ δημοφιλής για τα γρήγορα αποτελέσματα που παίρνει. Η μεθοδολογία Scrum χρησιμοποιείται κυρίως για την ανάπτυξη λογισμικού, αλλά και άλλοι τομείς εκμεταλλεύονται τα πλεονεκτήματά της εφαρμόζοντας αυτήν τη μεθοδολογία στα οργανωτικά τους μοντέλα, όπως ομάδες πωλήσεων, μάρκετινγκ και ανθρώπινου δυναμικού κ.λπ.

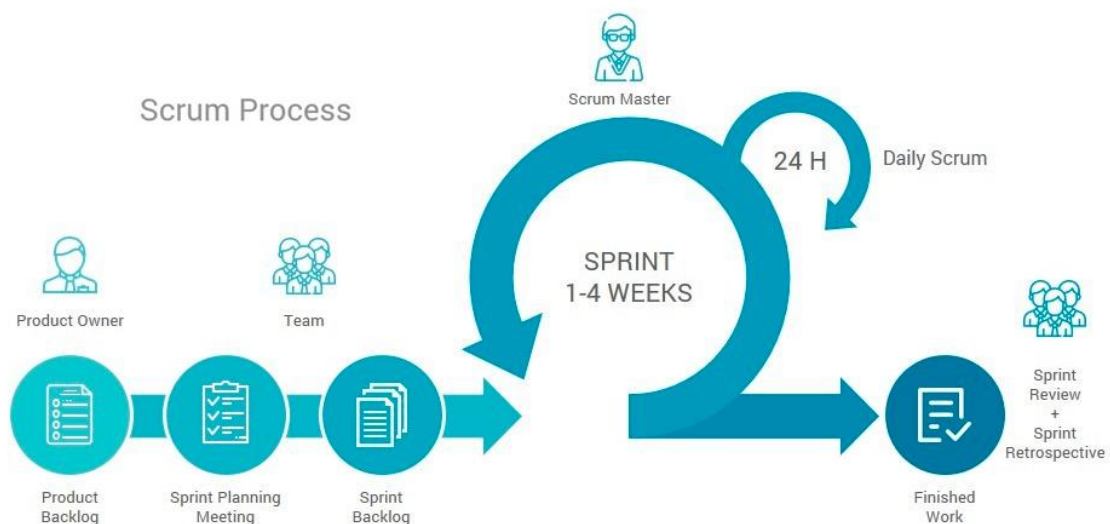
Στο Scrum, η ομάδα εστιάζει στη δημιουργία ποιοτικού λογισμικού. Ο ιδιοκτήτης ενός έργου Scrum εστιάζει στον καθορισμό των χαρακτηριστικών που πρέπει να έχει το προϊόν για να δημιουργήσει (τι να κατασκευάσει, τι όχι και με ποια σειρά) και να ξεπεράσει κάθε εμπόδιο που θα μπορούσε να εμποδίσει το έργο της ομάδας ανάπτυξης. Η ομάδα Scrum αποτελείται από τους ακόλουθους ρόλους:

- **Scrum Master:** Το άτομο που οδηγεί την ομάδα και την καθοδηγεί να συμμορφωθεί με τους κανόνες και τις διαδικασίες της μεθοδολογίας. Ο Scrum master διαχειρίζεται τη μείωση των εμποδίων του έργου και συνεργάζεται με τον κάτοχο του προϊόντος για να μεγιστοποιήσει την

απόδοση επένδυσης (ROI). Ο Scrum Master είναι υπεύθυνος για την ενημέρωση του Scrum, την παροχή καθοδήγησης, καθοδήγησης και εκπαίδευσης στις ομάδες σε περίπτωση που το χρειαστεί.

- **Product Owner:** Είναι ο εκπρόσωπος των ενδιαφερόμενων μερών και των πελατών που χρησιμοποιούν το λογισμικό. Επικεντρώνονται στο επιχειρηματικό κομμάτι και είναι υπεύθυνοι για την απόδοση επένδυσης (ROI) του έργου. Μεταφράζουν το όραμα του έργου στην ομάδα, επικυρώνουν τα οφέλη σε ιστορίες που πρέπει να ενσωματωθούν στο Product Backlog και δίνουν προτεραιότητα σε αυτά σε τακτική βάση.

Ομάδα Scrum: Μια ομάδα επαγγελματιών με τις απαραίτητες τεχνικές γνώσεις που αναπτύσσουν το έργο πραγματοποιώντας από κοινού τις ιστορίες για τις οποίες δεσμεύονται στην αρχή κάθε σπριντ.



Σχήμα 8- Scrum method (www.brainhub.eu)

Το Scrum έχει πολλά πλεονεκτήματα σε σχέση με άλλες ευέλικτες μεθοδολογίες ανάπτυξης. Αυτή τη στιγμή είναι το πιο χρησιμοποιούμενο και αξιόπιστο πλαίσιο αναφοράς στη βιομηχανία λογισμικού. Μερικά από τα γνωστά οφέλη του Scrum:

- **Εύκολη κλιμάκωση:** Οι διαδικασίες Scrum είναι επαναληπτικές και αντιμετωπίζονται εντός συγκεκριμένων περιόδων εργασίας, γεγονός που διευκολύνει την εστίαση της ομάδας σε συγκεκριμένες λειτουργίες

για κάθε περίοδο. Αυτό όχι μόνο έχει το πλεονέκτημα της επίτευξης καλύτερων παραδοτέων σύμφωνα με τις ανάγκες του χρήστη, αλλά δίνει επίσης τη δυνατότητα στις ομάδες να κλιμακώσουν τις ενότητες ως προς τη λειτουργικότητα, το σχεδιασμό, το εύρος και τα χαρακτηριστικά με τακτικό, διαφανή και απλό τρόπο.

- **Συμμόρφωση των προσδοκιών:** Ο πελάτης καθορίζει τις προσδοκίες του υποδεικνύοντας την αξία που αποφέρει κάθε απαίτηση/ιστορικό του έργου, η ομάδα τις εκτιμά και με αυτές τις πληροφορίες ο Κάτοχος Προϊόντος καθορίζει την προτεραιότητά του. Σε τακτική βάση, στις επιδείξεις σπριντ, ο Κάτοχος Προϊόντος επαληθεύει ότι πληρούνται οι απαιτήσεις και μεταδίδει σχόλια στην ομάδα.
- **Ευέλικτο στις αλλαγές:** Γρήγορη αντίδραση σε αλλαγές στις απαιτήσεις που δημιουργούνται από τις ανάγκες των πελατών ή τις εξελίξεις της αγοράς. Η μεθοδολογία έχει σχεδιαστεί για να προσαρμόζεται στις μεταβαλλόμενες απαιτήσεις που συνεπάγονται σύνθετα έργα.
- **Μείωση του χρόνου διάθεσης στην αγορά:** Ο πελάτης μπορεί να αρχίσει να χρησιμοποιεί τις πιο σημαντικές λειτουργίες του έργου πριν το προϊόν είναι εντελώς έτοιμο.
- **Υψηλότερη ποιότητα λογισμικού:** Η μέθοδος εργασίας και η ανάγκη λήψης μιας λειτουργικής έκδοσης μετά από κάθε επανάληψη, βοηθά στην απόκτηση λογισμικού υψηλότερης ποιότητας.
- **Έγκαιρη Πρόβλεψη:** Χρησιμοποιώντας αυτήν τη μεθοδολογία, γνωρίζουμε τη μέση ταχύτητα της ομάδας ανά σπριντ (σημεία ιστορίας), με την οποία, κατά συνέπεια, είναι δυνατό να εκτιμηθεί πότε θα είναι διαθέσιμη μια συγκεκριμένη λειτουργικότητα που εξακολουθεί να βρίσκεται στο ανεκτέλεστο.
- **Μείωση των κινδύνων:** Το γεγονός της εκτέλεσης των πιο πολύτιμων λειτουργιών στην πρώτη θέση και της γνώσης της ταχύτητας με την οποία η ομάδα προχωρά στο έργο, επιτρέπει την αποτελεσματική εκκαθάριση των κινδύνων εκ των προτέρων.

(www.digite.com/agile/scrum-methodology/)

2.6.2 Kanban

Το Kanban είναι ένα οπτικό σύστημα για τη διαχείριση της εργασίας καθώς κινείται μέσα από μια διαδικασία. Το Kanban οπτικοποιεί τόσο τη διαδικασία (τη ροή εργασίας) όσο και την πραγματική εργασία που διέρχεται από αυτήν τη διαδικασία. Ο στόχος του Kanban είναι να εντοπίσει πιθανά σημεία συμφόρησης στη διαδικασία και να τα διορθώσει, ώστε η εργασία να ρέει μέσω αυτής οικονομικά, αποδοτικά με τη βέλτιστη ταχύτητα ή απόδοση. Το πρώτο σύστημα Kanban αναπτύχθηκε από τον Taiichi Ohno (Βιομηχανικός Μηχανικός και Επιχειρηματίας) για την αυτοκινητοβιομηχανία Toyota στην Ιαπωνία. Δημιουργήθηκε ως ένα απλό σύστημα προγραμματισμού, στόχος του οποίου ήταν ο βέλτιστος έλεγχος και διαχείριση της εργασίας και του αποθέματος σε κάθε στάδιο της παραγωγής. Ένας βασικός λόγος για την ανάπτυξη του Kanban ήταν η ανεπαρκής παραγωγικότητα και αποδοτικότητα της Toyota σε σύγκριση με τους Αμερικανούς αντιπάλους της στον τομέα της αυτοκινητοβιομηχανίας. Με το Kanban, η Toyota πέτυχε ένα ευέλικτο και αποτελεσματικό σύστημα ελέγχου παραγωγής έγκαιρα, το οποίο αύξησε την παραγωγικότητα, ενώ μείωσε τα δαπανηρά απόθεμα πρώτων υλών, ημικατεργασμένων υλικών και τελικών προϊόντων. Ένα σύστημα Kanban ελέγχει ιδανικά ολόκληρη την αλυσίδα αξίας από τον προμηθευτή έως τον τελικό καταναλωτή. Με αυτόν τον τρόπο, βοηθά στην αποφυγή της διακοπής του εφοδιασμού και της υπεραπόθεσης αγαθών σε διάφορα στάδια της διαδικασίας παραγωγής. Το Kanban απαιτεί συνεχή παρακολούθηση της διαδικασίας. Ιδιαίτερη προσοχή πρέπει να δοθεί για να αποφευχθούν τα σημεία συμφόρησης που θα μπορούσαν να επιβραδύνουν τη διαδικασία παραγωγής. Ο στόχος είναι να επιτευχθεί υψηλότερη απόδοση με χαμηλότερους χρόνους παράδοσης. Με την πάροδο του χρόνου, το Kanban έχει γίνει ένας αποτελεσματικός τρόπος σε μια ποικιλία συστημάτων παραγωγής.

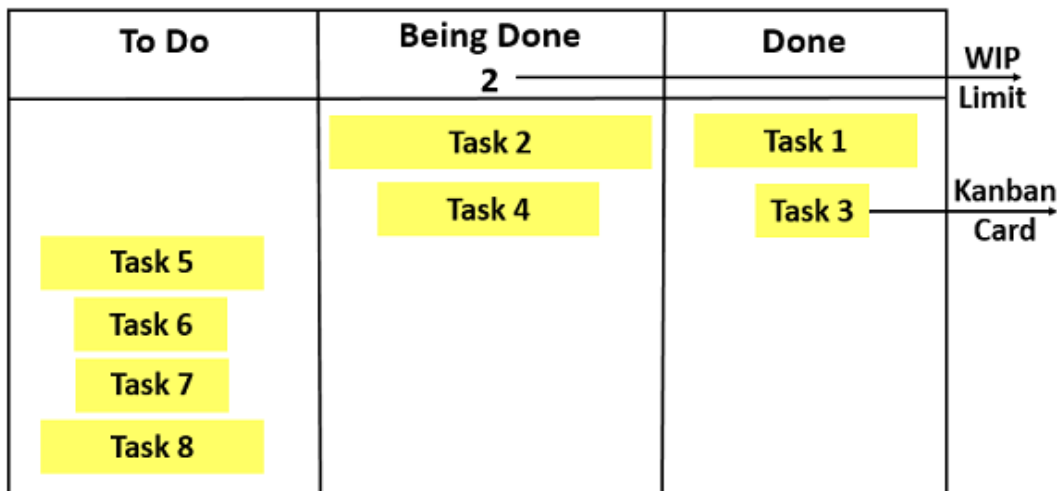
Η μέθοδος Kanban είναι μια διαδικασία για τη σταδιακή βελτίωση είτε πρόκειται για ανάπτυξη λογισμικού, IT/ Ops, Staffing, Recruitment, Marketing , Sales, Procurement κ.λπ. Στην πραγματικότητα, σχεδόν κάθε επιχειρηματική λειτουργία μπορεί να επωφεληθεί από την εφαρμογή των αρχών της Kanban Μεθοδολογίας. Η Μέθοδος Kanban ακολουθεί ένα σύνολο αρχών και

πρακτικών για τη διαχείριση και τη βελτίωση της ροής της εργασίας. Είναι μια εξελικτική, μη διασπαστική μέθοδος που προωθεί σταδιακές βελτιώσεις στις διαδικασίες ενός οργανισμού. Οι θεμελιώδεις αρχές που την διέπουν:

- Start with what you are doing now, το Kanban δεν απαιτεί συγκεκριμένη ρύθμιση και μπορεί να εφαρμοστεί απευθείας στην τρέχουσα ροή εργασίας. Αυτό καθιστά εύκολη την εφαρμογή, καθώς δεν χρειάζεται να αλλάξουν οι υπάρχουσες διαδικασίες. Τα οφέλη του Kanban είναι σταδιακά και οποιαδήποτε βελτίωση της διαδικασίας υιοθετείται με την πάροδο του χρόνου.
- Agree to pursue incremental, evolutionary change, οι σαρωτικές αλλαγές μπορούν να αναστατώσουν τις ομάδες, να διαταράξουν τη ροή και να βλάψουν την απόδοση. Το Kanban έχει σχεδιαστεί για να παρουσιάζει ελάχιστη αντίσταση ενθαρρύνοντας συνεχείς, σταδιακές και εξελικτικές αλλαγές.
- Respect the current process, roles, and responsibilities, δεν πρέπει να υπάρξουν οργανωτικές αλλαγές στην αρχή. Η μεθοδολογία αναγνωρίζει ότι οι υπάρχουσες διαδικασίες, οι ρόλοι και οι ευθύνες μπορεί να έχουν αξία και αξίζει να διατηρηθούν. Όμως ενθαρρύνεται η σταδιακή αλλαγή για να αποφευχθεί η συναισθηματική αντίσταση.
- Encourage acts of leadership at all levels, το Kanban προωθεί την ηγεσία και τη λήψη αποφάσεων μεταξύ όλων των μελών. Όλοι θα πρέπει να καλλιεργούν μια νοοτροπία συνεχούς βελτίωσης (Kaizen) προκειμένου οι εργαζόμενοι να έχουν τη βέλτιστη απόδοση.

(getnave.com/blog/what-is-the-kanban-method/)

Kanban Board



Σχήμα 9- Kanban schema (www.digite.com)

Το κύριο σημείο του πλαισίου Kanban είναι ότι βελτιώνει σημαντικά τον τρόπο με τον οποίο χειρίζονται τα έργα χωρίς να αλλάξει η οργανωτική δομή. Μερικά από τα οφέλη που προσφέρει η μεθοδολογία Kanban είναι:

- Ενισχυμένη ευελιξία
- Συνεχής βελτίωση
- Αυξημένη συνεργασία
- Ενδυνάμωση των εργαζομένων
- Ομαλή ροή εργασίας
- Καλύτερη διαχείριση αποθεμάτων
- Βελτιωμένος ποιοτικός έλεγχος

(www.digite.com/kanban/what-is-kanban/,

kissflow.com/project/agile/kanban-methodology/)

2.6.3 RAD

Το Rapid Application Development είναι ένα ευέλικτο πλαίσιο που επικεντρώνεται κυρίως στην ταχεία δημιουργία πρωτοτύπων προϊόντων λογισμικού, που επαναλαμβάνεται συχνά με βάση σχόλια και κυκλοφορεί συνεχώς ενημερωμένες εκδόσεις αυτών των προϊόντων στην αγορά. Η μέθοδος Rapid Application Development σχεδιάστηκε ως άμεση απάντηση στην τότε κυρίαρχη προσέγγιση καταρράκτη στην ανάπτυξη λογισμικού. Η

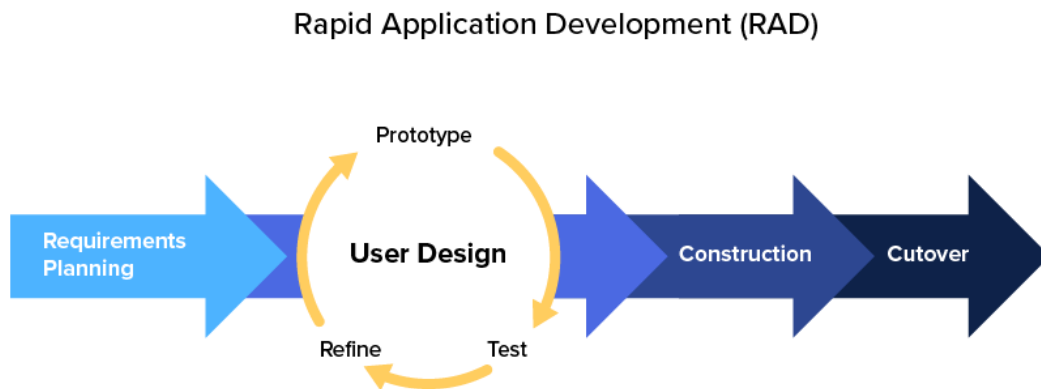
μεθοδολογία του καταρράκτη βασίστηκε σε διαδικασίες σχεδιασμού και διαδοχικού σχεδιασμού. Η ιδέα RAD εισήχθη επίσημα στο κοινό το 1991 με το βιβλίο Rapid Application Development του James Martin.

Η προσέγγιση RAD προσφέρει ισχυρά οφέλη σε μια ομάδα που είναι εξοικειωμένη με την ευέλικτη φιλοσοφία, έχει ένα σχετικά μικρό έργο να αναπτύξει και έχει πελάτες ή χρήστες πρόθυμους να δεσμευτούν ότι θα είναι μέρος ολόκληρου του έργου ανάπτυξης. Ωστόσο, υπάρχουν αρκετές περιπτώσεις στις οποίες το RAD μπορεί να μην είναι το καλύτερο πλαίσιο. Για παράδειγμα, εάν η εταιρεία χρειάζεται να αναπτύξει ένα έργο ανάπτυξης λογισμικού μεγάλης κλίμακας, εάν δεν υπάρχουν αρκετοί εξειδικευμένοι προγραμματιστές και σχεδιαστές στο προσωπικό ή εάν δεν μπορεί να εξασφαλιστεί μια δέσμευση για ενεργή συμμετοχή από τους χρήστες για την επαναληπτική διαδικασία.

Το μοντέλο RAD αποτελείται από τέσσερις φάσεις:

- **Φάση 1: Requirements Planning**, στην πρώτη φάση, όλα τα σχετικά μέλη (διευθυντές, προσωπικό πληροφορικής, χρήστες κ.λπ.) σχεδιάζουν και συμφωνούν για τις ανάγκες, το πεδίο εφαρμογής, τις προκλήσεις και τις απαιτήσεις του έργου. Αυτό που κάνει τη RAD να διαφέρει από άλλα μοντέλα από αυτή την άποψη είναι ότι θέτει ευρείες απαιτήσεις για να παραμείνει ευέλικτη με την πάροδο του χρόνου
- **Φάση 2: User Design**, σε αυτή τη φάση οι χρήστες συνεργάζονται με προγραμματιστές για να δημιουργήσουν και να αναπτύξουν ένα ή περισσότερα πρωτότυπα που καλύπτουν τις περιγραφόμενες απαιτήσεις συστήματος. Αυτή είναι μια συνεχής φάση όπου οι χρήστες αλληλοεπιδρούν με το πρωτότυπο και παρέχουν ανατροφοδότηση μέχρι να εγκριθεί ένα πραγματικό τελικό προϊόν.
- **Φάση 3: Construction**, αυτό το βήμα εστιάζει στην εφαρμογή των σχολίων που παρέχονται από τους χρήστες μέσω κωδικοποίησης, δοκιμής και οποιωνδήποτε άλλων εφαρμοζόμενων εργασιών ανάπτυξης. Η 2η και η 3η φάση τροφοδοτούνται μεταξύ τους έως ότου οι χρήστες εγκρίνουν το προϊόν.
- **Φάση 4: Cutover**, μόλις εγκριθεί τελικά το προϊόν, οι προγραμματιστές βάζουν κάποιες τελευταίες πινελιές με τη μορφή δοκιμής, μετατροπής,

διεπαφής ή εκπαίδευσης χρηστών. Μόλις το προϊόν αξιολογηθεί σωστά για παράγοντες όπως η σταθερότητα και η μακροζωία, είναι έτοιμο για παράδοση.



Σχήμα 10- RAD schema (www.kissflow.com)

Λόγω της ευελιξίας και της προσαρμοστικότητάς της σε νέες εισροές, μια προσέγγιση RAD ενέχει πολύ λιγότερο κίνδυνο από μια βασική μέθοδο που βασίζεται σε σχέδιο. Με ένα πρώιμο πρωτότυπο, είναι αρκετά εύκολο να εντοπιστούν τυχόν βασικές προκλήσεις που σχετίζονται με το έργο. Ως εκ τούτου, η RAD εξαλείφει τυχόν πιθανά προβλήματα νωρίς στον κύκλο ζωής, καθιστώντας φθηνότερη και ευκολότερη την αντιμετώπισή της κατά την ανάπτυξη. Ως άμεσο αποτέλεσμα, τα έργα RAD χρειάζονται συνήθως λιγότερο χρόνο για να ολοκληρωθούν. Ομοίως, η χρήση και η αξιολόγηση ενός πρωτοτύπου κατά τη διαδικασία ανάπτυξης επιτρέπει στους χρήστες να δίνουν σχόλια και να εντοπίζουν πιθανές αλλαγές πιο αποτελεσματικά. Αντί να προγραμματίζουν εκ των προτέρων το τελικό αποτέλεσμα (όπως με ένα βασικό μοντέλο Waterfall), οι χρήστες μπορούν να τροποποιήσουν και να προσαρμόσουν το πρωτότυπο για να αντιμετωπίσουν τυχόν σχόλια και παρατηρήσεις. Σε κάποιο βαθμό, αυτό καθιστά το RAD μια κυκλική προσέγγιση, με την εξέλιξη του προϊόντος να συμβαδίζει με την εμπειρία χρήστη. Με μια συνεχή ροή σχολίων και αλληλεπίδρασης με τους χρήστες, ένα έργο που αναπτύχθηκε με ένα μοντέλο RAD μπορεί να

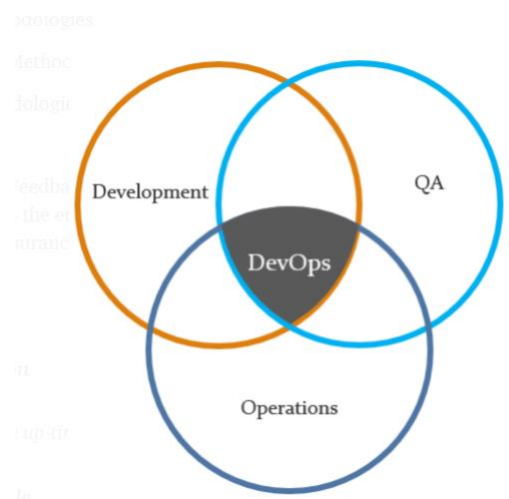
είναι πιο εφαρμόσιμο και ευκολότερο στην εφαρμογή σε ένα επιχειρηματικό περιβάλλον. Δεδομένου ότι το λογισμικό αλλάζει ανάλογα με τους χρήστες, το προϊόν που προκύπτει είναι πιο πιθανό να εκτιμηθεί από τους τελικούς χρήστες και να προσφέρει λειτουργικότητα φιλική προς το χρήστη. Τέλος, οι προσεγγίσεις RAD διευκολύνουν την αντιμετώπιση τυχόν δημοσιονομικών μειονεκτημάτων. Λόγω της ευελιξίας και της σταδιακής φύσης της, μια μέθοδος RAD επιτρέπει στους προγραμματιστές να εντοπίζουν και να αντιμετωπίζουν τεχνικά ζητήματα πιο γρήγορα και να αντιδρούν ανάλογα. Σε σύγκριση με μια προσέγγιση Waterfall, ο κίνδυνος οποιωνδήποτε αστοχιών μεγάλης κλίμακας είναι δραστικά μικρότερος.

www.creatio.com/page/rapid-application-development,

www.productplan.com/glossary/rapid-application-development/

2.7 DevOps Extension of Agile

Το DevOps είναι μια μεθοδολογία ανάπτυξης λογισμικού που συνδυάζει την ανάπτυξη λογισμικού (Dev) με τις λειτουργίες τεχνολογίας πληροφοριών (Ops) που συμμετέχουν μαζί σε ολόκληρο τον κύκλο ζωής της υπηρεσίας, από το



Σχήμα 11- DevOps schema (www.sitepoint.com)

σχεδιασμό μέσω της διαδικασίας ανάπτυξης ως την υποστήριξη παραγωγής. Με την

ευρύτερη έννοια του, το DevOps είναι μια φιλοσοφία που προωθεί την καλύτερη επικοινωνία και τη συνεργασία μεταξύ αυτών των ομάδων -- και άλλων -- σε έναν οργανισμό. Στην πιο στενή ερμηνεία του, το DevOps περιγράφει την υιοθέτηση επαναληπτικής ανάπτυξης λογισμικού, αυτοματισμού και προγραμματιζόμενης ανάπτυξης και συντήρησης υποδομής. Ο όρος καλύπτει επίσης αλλαγές κουλτούρας, όπως η οικοδόμηση εμπιστοσύνης και συνοχής μεταξύ προγραμματιστών και διαχειριστών συστημάτων και ευθυγράμμιση τεχνολογικών έργων με τις επιχειρηματικές απαιτήσεις. Το DevOps μπορεί να αλλάξει την αλυσίδα παράδοσης

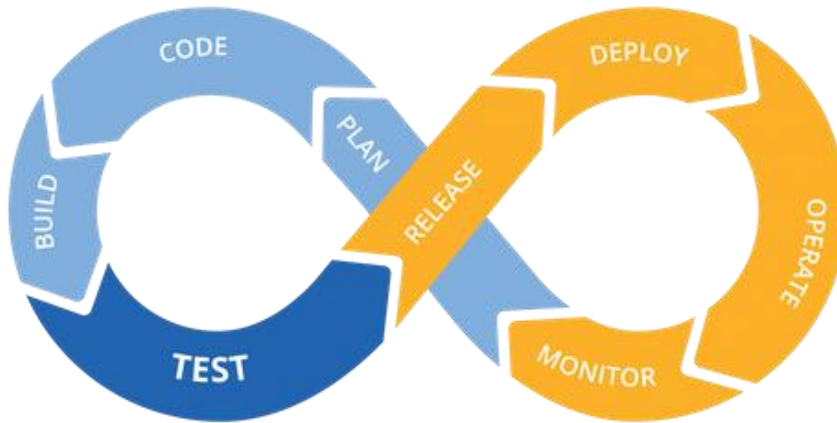
λογισμικού, τις υπηρεσίες, τους ρόλους εργασίας, τα εργαλεία πληροφορικής και τις βέλτιστες πρακτικές. Ενώ το DevOps δεν είναι τεχνολογία, τα περιβάλλοντα DevOps εφαρμόζουν γενικά κοινές μεθοδολογίες. Αυτά περιλαμβάνουν τα ακόλουθα:

- Εργαλεία συνεχούς ενοποίησης και συνεχούς παράδοσης ή συνεχούς ανάπτυξης (CI/CD), με έμφαση στην αυτοματοποίηση εργασιών.
- Συστήματα και εργαλεία που υποστηρίζουν την υιοθέτηση DevOps, συμπεριλαμβανομένης της παρακολούθησης σε πραγματικό χρόνο, της διαχείρισης συμβάντων, της διαχείρισης διαμορφώσεων και των πλατφορμών συνεργασίας.
- Cloud computing, microservices και Containers που υλοποιούνται ταυτόχρονα με τις μεθοδολογίες DevOps.

Η προσέγγιση DevOps είναι μία από τις πολλές τεχνικές που χρησιμοποιούν εταιρίες πληροφορικής για την εκτέλεση έργων που ανταποκρίνονται στις επιχειρηματικές ανάγκες. Τα DevOps μπορούν να συνυπάρξουν με την ανάπτυξη λογισμικού Agile, οδηγίες διαχείρισης έργων, όπως το Lean και το Six Sigma και άλλες στρατηγικές. Ορισμένοι επαγγελματίες πληροφορικής πιστεύουν ότι ο απλός συνδυασμός Dev και Ops δεν είναι αρκετός και ο όρος DevOps θα πρέπει να περιλαμβάνει ρητά τις επιχειρήσεις (BizDevOps), την ασφάλεια (DevSecOps) ή άλλους τομείς.

Το DevOps βασίζεται σε ευέλικτες μεθοδολογίες όπως το Scrum, το Kanban ή το XP για τη βελτίωση της ανάπτυξης και ανάπτυξης κώδικα. Οι ευέλικτες μέθοδοι επιτρέπουν γρήγορη δημιουργία πρωτοτύπων, συχνές εκδόσεις και σύντομους κύκλους ανάδρασης. Αυτά τα χαρακτηριστικά το καθιστούν ιδανικό για ταχέως εξελισσόμενα επιχειρηματικά περιβάλλοντα. Το DevOps φέρνει μια ολιστική προσέγγιση σε ολόκληρη τη γραμμή παράδοσης λογισμικού. Τα κοινά στάδια του DevOps είναι η συνεργασία, ο αυτοματισμός, η συνεχής ενοποίηση, η συνεχής παράδοση, η συνεχής δοκιμή, η συνεχής παρακολούθηση και η ταχεία αποκατάσταση. Το DevOps αξιοποιεί ομάδες πολλαπλών λειτουργιών. Αυτές οι ομάδες αποτελούνται από προγραμματιστές, σχεδιαστές, επιχειρήσεις, δοκιμαστές και ομάδες υποστήριξης. Πριν ενεργήσει σε μια διαδικασία ανάπτυξης έργου λογισμικού:

- Η ομάδα σχεδιάζει την παράδοση του λογισμικού από τη σχεδίαση έως την ανάπτυξη.
- Η ομάδα μηχανικών DevOps αναπτύσσει νέο κώδικα καθημερινά.
- Ο κωδικός ελέγχεται αυτόματα.



Σχήμα 12 – DevOps methodology (intercept.cloud/en/news/what-is-azure-devops/)

- Μετά από επιτυχή δοκιμή, ο κώδικας αναπτύσσεται για μια μικρή ομάδα χρηστών (Περιβάλλον Staging ή Testing).
- Εάν ο κώδικας είναι σταθερός, αναπτύσσεται για τους υπόλοιπους χρήστες. Εάν εντοπιστούν σφάλματα, διορθώνονται γρήγορα.

medium.com/@raycad.seedotech/devops-methodology-and-process-dde388eb65bd,

searchitoperations.techtarget.com/definition/DevOps,

crashtest-security.com/devops-methodology-guide/)

3

Εργαλεία και

Πλατφόρμες DevOps

Αφού έχουν αναλυθεί οι μεθοδολογίες διαχείρισης έργων και έχει παρουσιαστεί η χρησιμότητα του DevOps, κρίνεται σκόπιμο τόσο για λόγους πληρότητας όσο και για να γίνουν περισσότερο κατανοητές οι διαφορετικές μορφές και λειτουργίες του να γίνει παρουσίαση των πλατφορμών που το προσφέρουν. Στο παρόν κεφάλαιο παρουσιάζονται λύσεις και πλατφόρμες που έχουν ξεχωρίσει σε παγκόσμιο επίπεδο.

3.1 Οφέλη του DevOps

Με την αυξανόμενη έμφαση στις βελτιωμένες ικανότητες για αποτελεσματική ανάπτυξη και παράδοση λογισμικού, οι επιχειρήσεις συνειδητοποιούν την αξία του DevOps. Τα οφέλη είναι ιδανικά για όλους τους τύπους επιχειρήσεων, συμπεριλαμβανομένων κερδοσκοπικών και μη κερδοσκοπικών οργανισμών, καθώς και κυβερνητικών και εκπαιδευτικών ιδρυμάτων. Η έκθεση έρευνας με τίτλο «The State of IT Modernization» από την IDG δείχνει ότι μόνο το 26% περίπου των οργανισμών μπόρεσαν να περάσουν τα αρχικά στάδια του μετασχηματισμού της πληροφορικής. Επίσης οι επιχειρήσεις μπόρεσαν να συνειδητοποιήσουν τα οφέλη της εξοικονόμησης κόστους, του υψηλότερου χρόνου λειτουργίας, της καλύτερης ποιότητας υπηρεσιών και της βελτιωμένης εμπειρίας πελατών. Ωστόσο, πολλές επιχειρήσεις δεν είναι ξεκάθαροι σχετικά με τις ακριβείς δυνατότητες του DevOps για επιχειρηματικό μετασχηματισμό.

Ακολουθώς παρουσιάζονται τα οφέλη που φέρουν επανάσταση στον ψηφιακό μετασχηματισμό:

- **Μεγαλύτερη ευελιξία στην ανάπτυξη λογισμικού**, οι προγραμματιστές λογισμικού μπορούν να επιτύχουν ευελιξία μέσω του DevOps με σχόλια σε πραγματικό χρόνο από τους χρήστες, βελτιώνοντας έτσι τις δεξιότητές τους για την παροχή βελτιωμένου κώδικα με χαμηλότερο χρόνο διεκπεραίωσης. Αν και τα αποτελέσματα δεν είναι εμφανή αμέσως μέχρι την ανάπτυξή τους από την ομάδα, το DevOps παρέχει την ευελιξία για την ομάδα να συλλέγει σχόλια και να μοιράζεται με την ομάδα ανάπτυξης άμεσα για να διασφαλίζει γρήγορες αλλαγές. Τα οφέλη ευελιξίας με το DevOps βασίζονται σε μεγάλο βαθμό στη συμβιωτική σχέση μεταξύ των ομάδων ανάπτυξης λογισμικού και λειτουργιών πληροφορικής. Η έκταση της ευελιξίας με την εφαρμογή του DevOps αντανακλά από την αποτελεσματικότητα και των δύο ομάδων στην εκτέλεση των καθηκόντων τους και στην επικοινωνία μεταξύ τους. Καθώς τόσο οι ομάδες ανάπτυξης λογισμικού όσο και οι ομάδες λειτουργιών πληροφορικής χρησιμοποιούν παρόμοιες πλατφόρμες και εργαλεία, μπορούν να βρουν μείωση σε πολλά εμπόδια.
- **Βελτιωμένη ταχύτητα ανάπτυξης**, το DevOps συνεισφέρει σημαντική επιχειρηματική αξία παρέχοντας υψηλότερη ταχύτητα για την ανάπτυξη λογισμικού. Το DevOps έχει τη δυνατότητα να ενθαρρύνει την ταχύτητα συλλογής και υλοποίησης σχολίων, οδηγώντας έτσι σε ταχύτερες βελτιώσεις κώδικα. Στη συνέχεια, οι ταχύτερες βελτιώσεις στην ποιότητα του κώδικα μπορούν να μειώσουν το χρόνο μέχρι την παραγωγή. Ως αποτέλεσμα, οι επιχειρήσεις θα μπορούσαν να επιτύχουν ταχύτερο χρόνο διάθεσης στην αγορά για διαφορετικές λύσεις λογισμικού. Ως εκ τούτου, οι επιχειρήσεις μπορούν να επιτύχουν ένα πλεονέκτημα έναντι των ανταγωνιστών στην παροχή λύσεων λογισμικού, ενώ προσφέρουν αξία για τους πελάτες τους. Επιπλέον, οι επιχειρήσεις θα μπορούσαν να επιτύχουν καλύτερη ανθεκτικότητα στις διακυμάνσεις των συνθηκών της αγοράς και στα προηγούμενα συμπεριφορές των πελατών. Η ταχύτητα ανάπτυξης λογισμικού είναι

ένα από τα κορυφαία πλεονεκτήματα του DevOps καθώς δεν εστιάζει μόνο στην ταχύτερη επεξεργασία. Η ταχύτερη ανάπτυξη λογισμικού με το DevOps διευκολύνει τις ευκαιρίες για λειτουργική σταθερότητα, συνεχείς βελτιώσεις ποιότητας και αποτροπή διακοπών λειτουργίας.

- **Συντονισμός Ομάδων Ανάπτυξης και Λειτουργίας με άλλα Τμήματα**, το DevOps στοχεύει στη διασφάλιση του συντονισμού μεταξύ της ανάπτυξης λογισμικού, των λειτουργιών πληροφορικής και της υπόλοιπης επιχείρησης στις καθημερινές δραστηριότητες στην επιχείρηση. Ως εκ τούτου, μπορεί να διευκολύνει την ανάπτυξη μιας συνεχούς κοινής δέσμευσης και των τριών ομάδων προς ένα συνδυασμένο σύνολο στόχων. Το DevOps επιτρέπει στους οργανισμούς να δημιουργούν συνεπείς πολιτικές και διαδικασίες, ενώ παράλληλα ενθαρρύνει την κοινή ευθύνη για την επιτάχυνση των διαδικασιών. Το πιο σημαντικό από όλα είναι ότι οι επιχειρήσεις μπορούν να επιτύχουν τα πραγματικά οφέλη των DevOps ξεκινώντας από την ευθυγράμμιση, αντί να την αντιμετωπίζουν ως άσκηση οικοδόμησης ομάδας.
- **Μεγαλύτερη ισορροπία σταθερότητας και νέα χαρακτηριστικά**, το DevOps συνεπάγεται τον συνδυασμό τεχνικών που βασίζονται σε δοκιμές, μιας κοινής βάσης κωδικών, αυτοματοποιημένων αναπτύξεων και συνεχούς ενοποίησης για τον προληπτικό εντοπισμό προβλημάτων στον κώδικα, τη διαμόρφωση ή την υποδομή της εφαρμογής. Τα οφέλη του DevOps στη διατήρηση της σταθερότητας με την εισαγωγή νέων χαρακτηριστικών είναι επίσης εμφανή με τη μειωμένη πολυπλοκότητα του προβλήματος. Οι μικρότερες αλλαγές στο DevOps παρέχουν ευελιξία για βελτιώσεις του κώδικα. Επιπλέον, οι μηχανικοί DevOps θα μπορούσαν να χρησιμοποιήσουν δεδομένα σε πραγματικό χρόνο σχετικά με την απόδοση του συστήματος για την πρόβλεψη των επιπτώσεων των αλλαγών της εφαρμογής. Επιπλέον, οι ομάδες δεν χρειάζεται να περιμένουν άλλες ομάδες για την αντιμετώπιση προβλημάτων για να εξασφαλίσουν ταχύτερη επίλυση προβλημάτων με τον κώδικα.

- **Αξιόπιστη Παράδοση Υπηρεσιών**, η αξιοπιστία είναι ένα από τα κύρια πλεονεκτήματα του DevOps, καθώς είναι ένας κρίσιμος δείκτης αξίας για τους πελάτες. Η ακρίβεια στην παράδοση είναι ένα πολύ σημαντικό στοιχείο στην πτυχή της ικανοποίησης του πελάτη. Με τις μεθοδολογίες DevOps, οι επιχειρήσεις μπορούν να αναλύουν μεγάλα έργα σε μικρότερα στοιχεία με μοναδική λειτουργικότητα. Όλα τα μικρά εξαρτήματα παραδίδονται σταθερά σύμφωνα με μια συγκεκριμένη σειρά. Τα DevOps μπορούν να επιτρέψουν την παράδοση των απαιτήσεων για διαφορετικά στοιχεία ενός μεγαλύτερου έργου στο αρχικό και στα επόμενα στάδια. Όπως όλοι γνωρίζετε, το DevOps μπορεί να ενεργοποιήσει αλλαγές σε οποιαδήποτε στιγμή στη διαδικασία. Ως εκ τούτου, οι επιχειρήσεις έχουν την ευκαιρία να χρησιμοποιήσουν τα οφέλη του DevOps για να επιτύχουν καλύτερη, αξιόπιστη παροχή υπηρεσιών.
- **Μεγαλύτερη ικανοποίηση πελατών**, τα DevOps μπορούν να επιτρέψουν τη συχνή και έγκαιρη παράδοση ενημερώσεων σε διαφορετικούς πελάτες, παρέχοντάς τους ταυτόχρονα τη δυνατότητα αξιολόγησης και δοκιμής. Οι μικρότεροι κύκλοι κυκλοφορίας είναι ένας από τους κύριους λόγους για τα οφέλη ικανοποίησης πελατών από τα DevOps. Επιπλέον, το DevOps επιτρέπει τη δοκιμή των διαφορετικών εκδόσεων του προϊόντος μέσω δοκιμών σε διαφορετικές ομάδες πελατών. Ως αποτέλεσμα, οι επιχειρήσεις θα μπορούσαν να επιτύχουν καλύτερη σύγκριση για διάφορες νέες δυνατότητες που έχει να προσφέρει μια επιχείρηση με νέα προϊόντα λογισμικού ή ενημερώσεις. Οι επιχειρήσεις μπορούν να επιτύχουν σημαντικές βελτιώσεις στη συχνότητα ανάπτυξης και στους χρόνους ανάκτησης με μείωση των ποσοστών αποτυχίας αλλαγών μέσω του DevOps. Το πιο σημαντικό από όλα, η αυτοματοποίηση του αγωγού παράδοσης διασφαλίζει ότι η εφαρμογή είναι διαθέσιμη και σταθερή υπό διαφορετικές συνθήκες χρήσης.
- **Αυτοματοποίηση Συνεχούς Ενοποίησης και Συνεχούς Παράδοσης**, η πιο σημαντική πτυχή που οδηγεί σε διάφορα πλεονεκτήματα των DevOps αναφέρεται στη συνεχή κυκλοφορία και

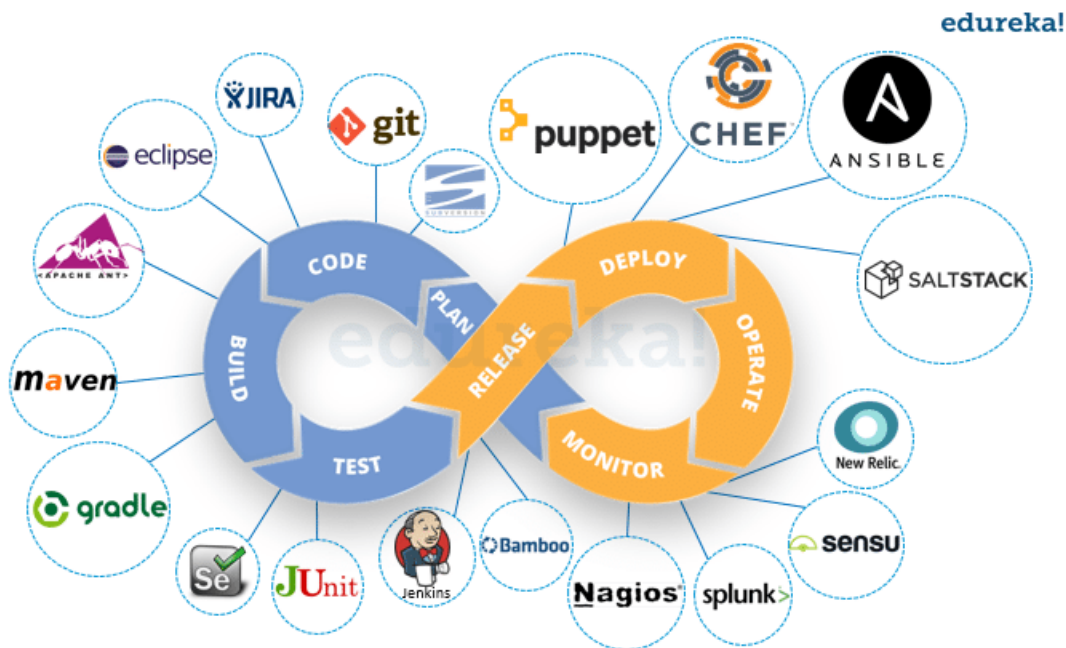
ανάπτυξη. Το DevOps παρέχει αυτοματοποίηση των διαδικασιών για την ενσωμάτωση νέων χαρακτηριστικών σε εφαρμογές και τη διασφάλιση της ανάπτυξής τους. Η αυτοματοποίηση των αγωγών συνεχούς ενοποίησης (CI) και συνεχούς παράδοσης (CD) στα DevOps βοηθούν στην ταχύτερη και στιγμιαία ανάπτυξη και ενσωμάτωση του κώδικα. Επιπλέον, η αυτοματοποίηση της διασφάλισης ποιότητας (QA) και η ενσωμάτωσή της στις διαδικασίες διασφαλίζει την ποιότητα του κώδικα. Η αυτοματοποίηση του CI και του CD όχι μόνο προσφέρει βελτιώσεις στην αποτελεσματικότητα, την ποιότητα και τη συνέχεια των εκδόσεων, αλλά παρέχει επίσης απλότητα στις διαδικασίες ανάπτυξης και ανάπτυξης

- **Μικρότερα σημεία συμφόρησης**, η επόμενη σημαντική αναφορά μεταξύ των πλεονεκτημάτων του DevOps είναι η μείωση των σημείων συμφόρησης στη διαδικασία ανάπτυξης. Το DevOps εστιάζει στη συνεργασία μεταξύ των ομάδων ανάπτυξης και λειτουργίας. Οι επιχειρήσεις που υιοθετούν DevOps πρέπει να επικεντρωθούν στη διασφάλιση της συνεργασίας μέσω cross-training σε όλες τις δύο ομάδες. Η διασταυρούμενη εκπαίδευση για συνεργασία θα μπορούσε να βοηθήσει στη μείωση του χρόνου αναμονής στη διαδικασία.
- **Εισαγωγή ευνοϊκών πολιτισμικών αλλαγών**, οι εργαζόμενοι αρχίζουν να ανακαλύπτουν την ανάγκη να εξαρτώνται ο ένας από τον άλλο για την επίτευξη των ενοποιημένων στόχων και να αντιμετωπίζουν την κοινή ευθύνη. Επιπλέον, η εστίαση στη συνεργασία διασφαλίζει επίσης ότι ο κύκλος ανάπτυξης και παράδοσης στο DevOps είναι ένα γεγονός προσανατολισμένο στην ομάδα. Όλα τα μέλη της ομάδας έχουν μια ακριβή ιδέα για τους στόχους που επιτυγχάνουν από κοινού. Επιπλέον, το DevOps βοηθά τους υπαλλήλους να συνειδητοποιήσουν τη συμβολή τους στην επιχειρηματική αξία μιας επιχείρησης πέρα από τις εξειδικεύσεις τους.
- **Πεδίο καινοτομίας και βελτίωσης**, η τελευταία προσθήκη μεταξύ των πλεονεκτημάτων του DevOps είναι η ευκαιρία για καινοτομία, μαζί με συνεχή μάθηση και βελτίωση. Το DevOps ενθαρρύνει τα μέλη της ομάδας να διασφαλίζουν τη συνεχή βελτίωση της διαδικασίας μέσω της

βιωματικής μάθησης. Επιπλέον, ο μετασχηματισμός του DevOps μπορεί να ενθαρρύνει μια πολιτισμική αλλαγή στον οργανισμό που επικεντρώνεται στην καινοτομία. Στη συνέχεια, το DevOps μπορεί να βοηθήσει τους εργαζόμενους να αισθάνονται ενδυναμωμένοι για την καινοτομία και την αντίληψη των λύσεων και να χρησιμοποιούν περιπτώσεις από διαφορετικές οπτικές γωνίες.
(www.whizlabs.com/blog/top-benefits-of-devops/)

3.2 Πλατφόρμες DevOps

Το DevOps είναι μια κουλτούρα, η πιο αξιοσημείωτη ιδέα που έφερε ποτέ η προσέγγιση DevOps είναι η συνεργασία μεταξύ των ομάδων ανάπτυξης λογισμικού και λειτουργιών. Επίσης η έμφαση στην προσέγγιση DevOps δίνεται στην αυτοματοποίηση των διαδικασιών ανάπτυξης λογισμικού, όπως η κατασκευή, η δοκιμή, η ανίχνευση και απόκριση περιστατικών, η απελευθέρωση και άλλες, για την επίτευξη ταχύτερης κυκλοφορίας, προϊόντα υψηλής ποιότητας και μειωμένες αστοχίες και επαναλήψεις. λειτουργιών λογισμικού/λογισμικού. Ωστόσο, το DevOps είναι σήμερα κάτι περισσότερο από μια συλλογική κουλτούρα και αυτοματισμός ανάπτυξης λογισμικού. Ενσωματώνει αναδυόμενες τεχνολογίες όπως η τεχνητή νοημοσύνη (AI), η μηχανική μάθηση (ML), το διαδίκτυο των πραγμάτων (IoT) και το cloud computing. Έχουν αναπτυχθεί πάρα πολλά εργαλεία DevOps για κατασκευή, έλεγχο έκδοσης, διαχείριση διαμόρφωσης, διαχείριση έργου, διαχείριση περιστατικών και πολλά άλλα. Παρακάτω θα αναλυθούν μερικά κορυφαία εργαλεία σε διάφορες κατηγορίες διαδικασιών DevOps.
(www.simplilearn.com/tutorials/devops-tutorial/devops-tools)



Σχήμα 13- DevOps platforms (www.tutcourse.com)

3.2.1 AWS Cloud Computing and Storage in DevOps



Το AWS διαθέτει την ευρύτερη γκάμα προσφορών υπηρεσιών στις κατηγορίες PaaS, SaaS και IaaS, συμπεριλαμβανομένων υπολογιστών, διαχείρισης ταυτότητας και πρόσβασης (ACM), δικτύωσης και αποθήκευσης. Ενώ το AWS προσφέρει δημόσια, ιδιωτικά και υβριδικά clouds, η εστίασή του είναι περισσότερο στο δημόσιο cloud. Το AWS παρέχει ένα σύνολο ευέλικτων υπηρεσιών που έχουν σχεδιαστεί για να επιτρέπουν στους οργανισμούς να δημιουργούν και να παραδίδουν προϊόντα ταχύτερα και πιο αξιόπιστα εφαρμόζοντας πρακτικές DevOps. Αυτές οι υπηρεσίες απλοποιούν την παροχή και τη διαχείριση της υποδομής, την ανάπτυξη κώδικα εφαρμογής, την αυτοματοποίηση των διαδικασιών έκδοσης λογισμικού και την παρακολούθηση της απόδοσης της εφαρμογής και της υποδομής. Το DevOps στο AWS συμβάλλει στη συγκέντρωση του οργανισμού αξιοποιώντας την υποδομή ως υπηρεσίες κώδικα όπως το AWS CloudFormation και το AWS Cloud Development Kit (CDK). Η συνεχής ανάπτυξη γίνεται με τη βοήθεια υπηρεσιών όπως το AWS CodeBuild, το AWS CodeDeploy, το AWS CodePipeline και το AWS

CodeCommit. Στη συνέχεια, οι εφαρμογές αναπτύσσονται μέσω υπηρεσιών όπως το AWS Elastic Beanstalk, το Amazon Elastic Container Service (Amazon ECS) ή το Amazon Elastic Kubernetes Service (Amazon EKS) και το AWS OpsWorks για την απλοποίηση της διαμόρφωσης κοινών αρχιτεκτονικών. Η χρήση αυτών των υπηρεσιών διευκολύνει επίσης τη συμπερίληψη άλλων σημαντικών υπηρεσιών, όπως η αυτόματη κλιμάκωση και η εξισορρόπηση ελαστικού φορτίου. Τέλος, η παρακολούθηση και η καταγραφή γίνεται μέσω του Amazon CloudWatch και του Cloud Trail. Τα εργαλεία της AWS ανάλογα με τον ρόλο τους μπορούν να κατηγοριοποιηθούν σε διαφορετικές ενότητες:

- **Συνεχής ενοποίηση και συνεχής παράδοση (CI/CD)**, το Continuous Integration (CI) είναι μια πρακτική ανάπτυξης λογισμικού όπου οι προγραμματιστές συγχωνεύουν τακτικά τις αλλαγές κώδικα σε ένα κεντρικό αποθετήριο κώδικα, με το οποίο εκτελούνται αυτοματοποιημένες εκδόσεις και δοκιμές. Το CI βοηθά στην γρηγορότερη εύρεση και αντιμετώπιση σφαλμάτων, στη βελτίωση της ποιότητας του λογισμικού και στη μείωση του χρόνου που απαιτείται για την επικύρωση και την κυκλοφορία νέων ενημερώσεων λογισμικού. Η συνεχής παράδοση είναι μια πρακτική ανάπτυξης λογισμικού όπου οι αλλαγές κώδικα προετοιμάζονται αυτόματα για κυκλοφορία στην παραγωγή. Η συνεχής παράδοση επεκτείνεται με τη συνεχή ενσωμάτωση με την ανάπτυξη όλων των αλλαγών κώδικα σε περιβάλλον δοκιμής ή/και περιβάλλον παραγωγής μετά το στάδιο κατασκευής.
 - Το AWS CodePipeline είναι μια υπηρεσία συνεχούς παράδοσης που δίνει τη δυνατότητα να μοντελοποιηθούν, να οπτικοποιηθούν και να αυτοματοποιηθούν τα βήματα που απαιτούνται για την κυκλοφορία του λογισμικού. Με το AWS CodePipeline, η διαδικασία πλήρους έκδοσης μπορεί να μοντελοποιηθεί για την κατασκευή κώδικα, την ανάπτυξη σε περιβάλλοντα προπαραγωγής, τη δοκιμή της εφαρμογής και την κυκλοφορία της στην παραγωγή. Στη συνέχεια, το AWS CodePipeline δημιουργεί, δοκιμάζει και αναπτύσσει την

εφαρμογή σύμφωνα με την καθορισμένη ροή εργασίας κάθε φορά που υπάρχει αλλαγή κώδικα.

- Το AWS CodeBuild είναι μια πλήρως διαχειριζόμενη υπηρεσία συνεχούς ενοποίησης που μεταγλωττίζει τον πηγαίο κώδικα, εκτελεί δοκιμές και παράγει πακέτα λογισμικού που είναι έτοιμα για ανάπτυξη. Ο προγραμματιστής θα πρέπει να παρέχει, να διαχειρίζεται και να κλιμακώνει τους διακομιστές κατασκευής. Το CodeBuild κλιμακώνεται συνεχώς και μπορεί να επεξεργάζεται πολλαπλές κατασκευές ταυτόχρονα. Το CodeBuild προσφέρει διάφορα προ ρυθμισμένα περιβάλλοντα για διάφορες εκδόσεις των Microsoft Windows και Linux. Οι πελάτες μπορούν επίσης να φέρουν τα προσαρμοσμένα περιβάλλοντα κατασκευής τους ως κοντέινερ Docker.
 - Το AWS CodeDeploy είναι μια πλήρως διαχειριζόμενη υπηρεσία ανάπτυξης που αυτοματοποιεί τις αναπτύξεις λογισμικού σε μια ποικιλία υπολογιστικών υπηρεσιών όπως το Amazon Elastic Compute Cloud (Amazon EC2), το AWS Fargate, το AWS Lambda και οι διακομιστές εντός της εγκατάστασης. Το AWS CodeDeploy διευκολύνει να απελευθερωθούν γρήγορα νέες δυνατότητες, βοηθά να αποφευχθεί ο χρόνος διακοπής λειτουργίας κατά την ανάπτυξη της εφαρμογής και χειρίζεται την πολυπλοκότητα της ενημέρωσης των εφαρμογών.
 - Το AWS CodeStar παρέχει μια εννοποιημένη διεπαφή χρήστη, επιτρέποντάς να διαχειριστούν εύκολα τις δραστηριότητες ανάπτυξης λογισμικού σε ένα μέρος. Με το AWS CodeStar, μπορεί να ρυθμιστεί ολόκληρη η αλυσίδα εργαλείων συνεχούς παράδοσης μέσα σε λίγα λεπτά, επιτρέποντάς να ξεκινήσει η απελευθέρωση κώδικα πιο γρήγορα.
- **Microservices**, Δημιουργείται και αναπτύσσεται μια αρχιτεκτονική microservices χρησιμοποιώντας κοντέινερ ή υπολογιστές χωρίς διακομιστή
- Η υπηρεσία Amazon Elastic Container Service (Amazon ECS) είναι μια πλήρως διαχειριζόμενη υπηρεσία ενορχήστρωσης

Χρυσούλα Ζάχαρη

εμπορευματοκιβωτίων που βοηθάει να αναπτυχθεί, να διαχειριστεί και να κλιμακώσει εύκολα εφαρμογές που έχουν διαμορφωθεί σε εμπορευματοκιβώτια. Ενσωματώνεται σε βάθος με την υπόλοιπη πλατφόρμα AWS για να παρέχει μια ασφαλή και εύχρηστη λύση για την εκτέλεση φόρτου εργασίας κοντέινερ στο cloud.

- ο Το AWS Lambda είναι μια υπολογιστική υπηρεσία χωρίς διακομιστή που επιτρέπει να εκτελεστεί κώδικας χωρίς παροχή ή διαχείριση διακομιστών, δημιουργία λογικής κλιμάκωσης συμπλέγματος με επίγνωση του φόρτου εργασίας, διατήρηση ενσωματώσεων συμβάντων ή διαχείριση χρόνου εκτέλεσης. Με το Lambda, μπορεί να εκτελεστεί κώδικα για σχεδόν κάθε τύπο εφαρμογής ή υπηρεσία υποστήριξης όλα με μηδενική διαχείριση.
- **Υποδομή ως Κώδικας**, παρέχει, διαμορφώνει και διαχειρίζεται τους πόρους της υποδομής AWS χρησιμοποιώντας κώδικα και πρότυπα.



Σχήμα 14 - AWS DevOps (aws.amazon.com)

- **Παρακολούθηση και Καταγραφή**, γίνεται καταγραφή αρχείων και παρακολούθηση της απόδοσης εφαρμογών και υποδομών σε σχεδόν

πραγματικό χρόνο με τη χρήση του Cloud and Network Monitoring και του Distributed Tracing.

- **Πλατφόρμα ως υπηρεσία**, βοηθά στην ανάπτυξη εφαρμογών Ιστού χωρίς να χρειάζεται να παρέχεται και να γίνεται διαχείριση της υποδομής και της στοίβας εφαρμογών.
- **Έλεγχος έκδοσης**, το AWS CodeCommit είναι μια πλήρως διαχειριζόμενη υπηρεσία ελέγχου πηγών που διευκολύνει τις εταιρείες να φιλοξενούν ασφαλή και εξαιρετικά επεκτάσιμα ιδιωτικά αποθετήρια Git.

(<https://www.varonis.com/blog/aws-devops>, <https://aws.amazon.com/devops/>)

Παρακάτω παρουσιάζεται ένας πίνακας σύγκρισης της AWS με το Azure DevOps:

Διοίκηση έργων μέσω σύγχρονων μεθόδων: Μελέτη περίπτωσης το εργαλείο
Microsoft Azure DevOps

Χρυσούλα Ζάχαρη

Parameters	AWS	Azure
Launched Date	Launched in 2006.	Launched in 2010.
Market Share	31% Share of the global computing market	11% Share from the worldwide market.
Availability Zone	61 Availability Zone	140 Availability Zones
Storage services	<ul style="list-style-type: none"> S3 Buckets EBS SDB domains Easy to use SQS CloudFront AWS Import/Export 	<ul style="list-style-type: none"> Blob Storage Containers Azure Drive Table Storage Tables Storage Stats
Databases Services	<ul style="list-style-type: none"> MySQL Oracle DynamoDB 	<ul style="list-style-type: none"> MS SQL SQL Sync
Deployment Services	<ul style="list-style-type: none"> Amazon Web Services Amazon Machine Instance (AMI) Traditional Deployment Models Fine-grained updates Elastic Beanstalk Cloud Formation 	<p>Cspkg (fancy zip file)</p> <p>Upload via portal or API via blob storage</p> <p>Course-grained updates</p> <p>"click to scale."</p> <p>More magic</p>
Networking Services	<ul style="list-style-type: none"> IP/Elastic IP/ELB Virtual Private Cloud Route 53 ELB Firewall heavily configurable 	<ul style="list-style-type: none"> Automatic IP assignment Load-balancing Azure Connect Balancing Endpoints defined in csdef/cscfg
Price	Per hour- rounded up	On-demand reserved spot.
Customers	Adobe, Airbnb, Expedia, Yelp, Nokia, Netflix, Novartis.	Pearson, 3M, Towers Watson, NBC, Essar, Serko, etc.
Type of Cloud	Virtual Private Cloud (VPC)	Virtual Network
Connection type	Direct Connect	ExpressRoute

Σχήμα 15- Σύγκριση AWS με Azure DevOps (www.guru99.com)

Από τον πίνακα συμπεραίνουμε πως η AWS έχει το μεγαλύτερο μερίδιο αγοράς, αλλά όχι σε βαθμό που να υπάρχει πραγματικός ανταγωνισμός μεταξύ των δύο εταιρειών. Επιπλέον, και οι δύο εταιρείες εισάγουν νέα προϊόντα, νέες ενοποιήσεις και νέες δομές που τις εξελίσσουν και βελτιώνουν τις ανάγκες των πελατών τους.

3.2.2 Azure Cloud Computing and Storage in DevOps



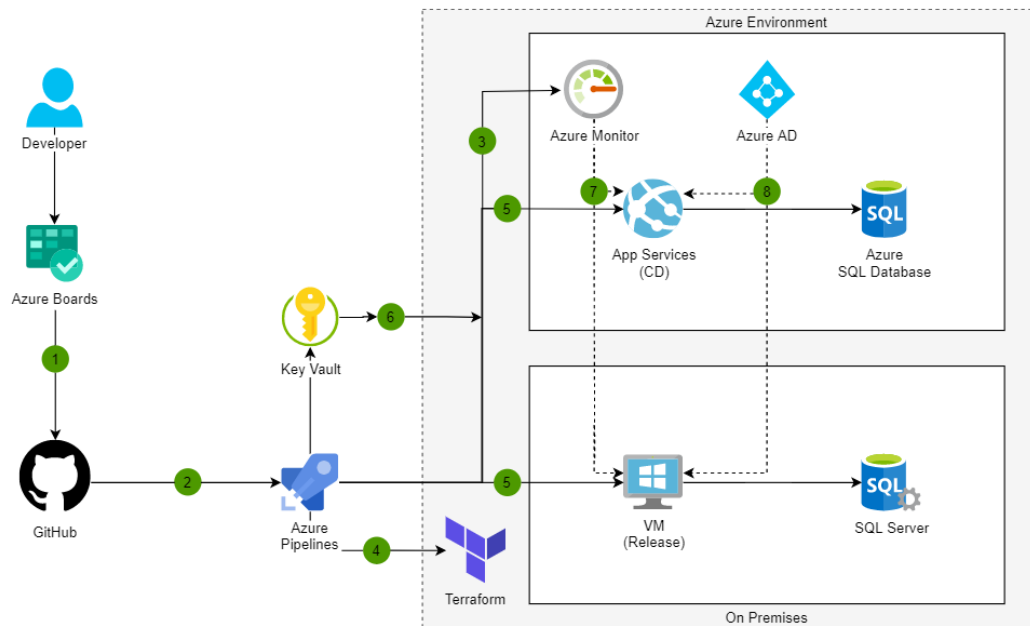
Azure DevOps

Το Azure DevOps είναι μια ολοκληρωμένη πλατφόρμα DevOps που βασίζεται στην πλατφόρμα cloud της Microsoft Azure, με τον έλεγχο έκδοσης να περιλαμβάνεται στην υπηρεσία Azure Repos στη σουίτα DevOps. Η πλατφόρμα Azure DevOps διατίθεται με δυο τρόπους:

- Οι Υπηρεσίες Azure DevOps που υποστηρίζονται από το Azure
- Ο διακομιστής Azure DevOps είναι ο διάδοχος του Microsoft Team Foundation Server (MTFS) και προσφέρει μια επιλογή αυτοδιαχείρισης εκτός σύνδεσης

Η υπηρεσία Azure DevOps που βασίζεται σε cloud προσφέρει την πλήρη πλατφόρμα εξαιρουμένων των σχεδίων δοκιμών εντελώς δωρεάν για έως και πέντε χρήστες. Οι συνδρομές Visual Studio αποκτούν πλήρη πρόσβαση στις υπηρεσίες Azure DevOps. Είναι η καλύτερη επιλογή για την τροφοδοσία αγωγών ανάπτυξης λογισμικού σε οποιαδήποτε κλίμακα, εάν η επιχείρηση είναι στο οικοσύστημα της Microsoft ή βασίζεται στο Azure για την εφαρμογή της. Τα κύρια χαρακτηριστικά είναι:

- Άμεση ενοποίηση με την πλατφόρμα cloud Azure και άλλες προσφορές υπηρεσιών της Microsoft
- Μια πλήρης πλούσια σε χαρακτηριστικά πλατφόρμα για δωρεάν πρόσβαση που περιλαμβάνει απεριόριστους ιδιωτικούς χώρους αποθήκευσης
- Το Azure DevOps Server Express για εφαρμογή του Azure DevOps εκτός σύνδεσης για μεμονωμένους προγραμματιστές ή ομάδες έως 5 χρηστών εντελώς δωρεάν
- Περισσότερες από 1.000 ενσωματώσεις είναι διαθέσιμες μέσω του Extensions Marketplace για συμπλήρωση και επέκταση της λειτουργικότητας DevOps
- Azure Boards, Pipelines, Repos, Test Plans, Artifacts για την τροφοδοσία όλων των σταδίων ενός SDLC
- Υποστήριξη για κάθε πελάτη Git
- Αναζήτηση σημασιολογικού κώδικα



Σχήμα 16- Azure DevOps ecosystem (docs.microsoft.com)

3.2.3 Version Control Tool: Git (GitLab, GitHub, Bitbucket)



Το Git είναι ένα από τα πιο δημοφιλή εργαλεία DevOps, που χρησιμοποιείται ευρέως στη βιομηχανία λογισμικού. Είναι ένα κατακευματισμένο εργαλείο SCM (διαχείριση πηγών κώδικα), το οποίο χρησιμοποιούν κυρίως οι απομακρυσμένες ομάδες και οι συνεισφέροντες ανοιχτού κώδικα. Το Git επιτρέπει την παρακολούθηση της προόδου των εργασιών ανάπτυξης. Οι χρήστες μπορούν να αποθηκεύσουν διαφορετικές εκδόσεις του πηγών κώδικα και να επιστρέψουν σε μια προηγούμενη έκδοση όταν είναι απαραίτητο. Είναι επίσης εξαιρετικό για πειραματισμούς, καθώς μπορούν να δημιουργήσουν ξεχωριστούς κλάδους και να συγχωνεύσουν νέες λειτουργίες μόνο όταν είναι έτοιμες για χρήση. Από προεπιλογή, το Git είναι ένα εργαλείο γραμμής εντολών, αλλά οι χρήστες μπορούν επίσης να κάνουν λήψη του προγράμματος-πελάτη GUI που τους επιτρέπει να διαχειρίζονται τον πηγαίο κώδικα από μια φιλική προς το χρήστη γραφική διεπαφή. Πολλοί επεξεργαστές πηγών κώδικα, όπως το Visual Studio Code, διαθέτουν επίσης ένα ενσωματωμένο εργαλείο επεξεργασίας Git. Για να ενσωματωθεί το Git με τη ροή εργασίας DevOps, πρέπει επίσης να

φιλοξενηθεί ο πηγαίος κώδικας σε αποθετήριο όπου τα μέλη της ομάδας μπορούν να προωθήσουν την εργασία τους. Επί του παρόντος, οι πιο δημοφιλείς διαδικτυακές υπηρεσίες φιλοξενίας repo Git είναι οι GitHub, GitLab και Bitbucket. Μπορεί να φιλοξενήσει δημόσια και ιδιωτικά αποθετήρια σε αυτές τις πλατφόρμες, να παρακολουθούνται και να συζητούνται θέματα και να διαχειρίζονται εκδόσεις. Παρέχουν επίσης πρόσθετες λειτουργίες DevOps, όπως ενσωματωμένες λειτουργίες συνεχούς ανάπτυξης και ασφάλειας, έλεγχο κώδικα, εργαλεία ελέγχου και συνεργασίας. (raygun.com)

Τα οφέλη του Git είναι τα ακόλουθα:

- **Ταυτόχρονη ανάπτυξη**, ο καθένας έχει το δικό του τοπικό αντίγραφο κώδικα και μπορεί να εργάζεται ταυτόχρονα στα δικά του υποκαταστήματα. Το Git λειτουργεί εκτός σύνδεσης αφού σχεδόν κάθε λειτουργία είναι τοπική.
- **Πιο γρήγορες εκδόσεις**, οι κλάδοι επιτρέπουν την ευέλικτη και ταυτόχρονη ανάπτυξη. Ο κύριος κλάδος περιέχει σταθερό, υψηλής ποιότητας κωδικό από τον οποίο αποδεδεσμεύεται. Οι κλάδοι δυνατοτήτων περιέχουν εργασίες σε εξέλιξη, οι οποίες συγχωνεύονται στον κύριο κλάδο μετά την ολοκλήρωση. Διαχωρίζοντας τον κλάδο έκδοσης από την ανάπτυξη σε εξέλιξη, είναι ευκολότερο να γίνει διαχείριση του σταθερού κώδικα και να αποσταλούν ενημερώσεις πιο γρήγορα.
- **Ενσωματωμένο integration**, λόγω της δημοτικότητάς του, το Git είναι ενσωματωμένο στα περισσότερα εργαλεία και προϊόντα. Κάθε σημαντικό IDE έχει ενσωματωμένη υποστήριξη Git και πολλά εργαλεία υποστηρίζουν τη συνεχή ενοποίηση, τη συνεχή ανάπτυξη, τις αυτοματοποιημένες δοκιμές, την παρακολούθηση στοιχείων εργασίας, τις μετρήσεις και την ενοποίηση χαρακτηριστικών αναφορών με το Git. Αυτή η ενοποίηση απλοποιεί την καθημερινή ροή εργασίας.
- **Ισχυρή κοινοτική υποστήριξη**, το Git είναι ανοιχτού κώδικα και έχει γίνει το de facto πρότυπο για τον έλεγχο εκδόσεων, επομένως δεν υπάρχει έλλειψη εργαλείων και πόρων που μπορούν να αξιοποιήσουν οι ομάδες. Ο όγκος της υποστήριξης της κοινότητας για το Git σε

σύγκριση με άλλα συστήματα ελέγχου εκδόσεων διευκολύνει τη λήψη βοήθειας όταν χρειάζεται.

- **Το Git λειτουργεί με οποιαδήποτε ομάδα**, η χρήση του Git με ένα εργαλείο διαχείρισης πηγαίου κώδικα μπορεί να αυξήσει την παραγωγικότητα μιας ομάδας ενθαρρύνοντας τη συνεργασία, επιβάλλοντας πολιτικές, αυτοματοποιώντας τις διαδικασίες και βελτιώνοντας την ορατότητα και την ιχνηλασιμότητα της εργασίας. Η ομάδα μπορεί να εγκατασταθεί σε μεμονωμένα εργαλεία για έλεγχο έκδοσης, παρακολούθηση αντικειμένων εργασίας και συνεχή ενοποίηση και ανάπτυξη. Ή, μπορούν να επιλέξουν μια λύση όπως το GitHub ή το Azure DevOps που υποστηρίζει όλες αυτές τις εργασίες σε ένα μέρος.
- **Αιτήματα έλξης**, χρησιμοποιούνται αιτήματα έλξης για να συζητούνται οι αλλαγές κώδικα με την ομάδα πριν τις συγχώνευση στον κύριο κλάδο. Οι συζητήσεις στα αιτήματα έλξης είναι πολύτιμες για τη διασφάλιση της ποιότητας του κώδικα και την αύξηση της γνώσης σε όλη την ομάδα σας. Πλατφόρμες όπως το GitHub και το Azure DevOps προσφέρουν μια πλούσια εμπειρία αιτήματος έλξης όπου οι προγραμματιστές μπορούν να περιηγηθούν σε αλλαγές αρχείων, να αφήσουν σχόλια, να επιθεωρήσουν δεσμεύσεις, να προβάλουν εκδόσεις και να ψηφίσουν για την έγκριση του κώδικα.
- **Πολιτικές υποκαταστημάτων**, οι ομάδες μπορούν να διαμορφώσουν το GitHub και το Azure DevOps για να επιβάλουν συνεπείς ροές εργασίας και διαδικασίες σε όλη την ομάδα. Μπορούν να δημιουργήσουν πολιτικές υποκαταστημάτων για να διασφαλίσουν ότι τα αιτήματα έλξης πληρούν τις απαιτήσεις πριν από την ολοκλήρωση. Οι πολιτικές υποκαταστημάτων προστατεύουν σημαντικούς κλάδους αποτρέποντας άμεσες πιέσεις, απαιτώντας αναθεωρητές και διασφαλίζοντας καθαρές κατασκευές. (docs.microsoft.com)

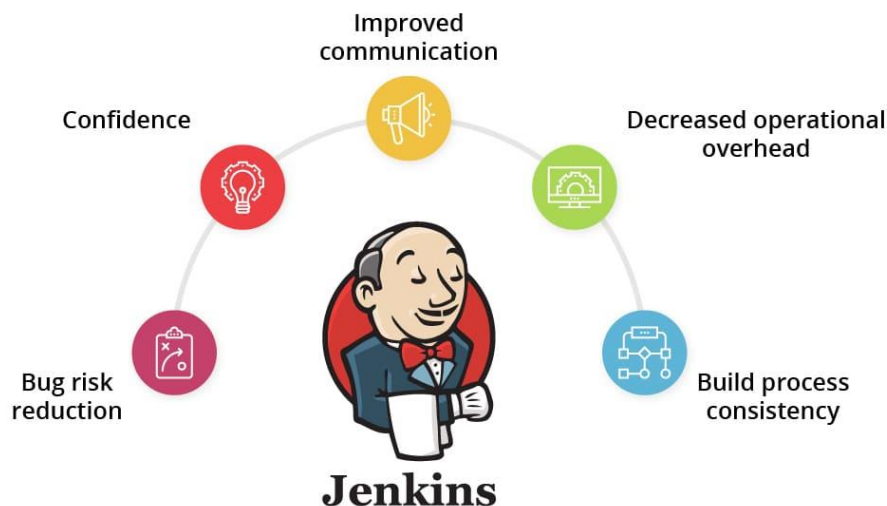
3.2.4 Continuous Integration Tool: Jenkins



Jenkins

Το Jenkins είναι ένα εργαλείο ενοποίησης DevOps. Για συνεχή ενοποίηση (CI), το Jenkins

ξεχωρίζει καθώς έχει σχεδιαστεί τόσο για εσωτερικές επεκτάσεις όσο και για επεκτάσεις πρόσθετων. Το Jenkins είναι ένας διακομιστής CI αυτοματισμού ανοιχτού κώδικα που βασίζεται σε Java και υποστηρίζεται από πολλαπλά λειτουργικά συστήματα, συμπεριλαμβανομένων των Windows, macOS και άλλων λειτουργικών συστημάτων Unix. Το Jenkins μπορεί επίσης να αναπτυχθεί σε πλατφόρμες που βασίζονται σε cloud. Η συνεχής ενσωμάτωση και η συνεχής παράδοση είναι δύο βασικές πρακτικές της μεθοδολογίας DevOps που καθιστούν το Jenkins απαραίτητο εργαλείο DevOps. Το Jenkins είναι συμβατό με τα περισσότερα εργαλεία και υπηρεσίες ενοποίησης CI/CD χάρη στα περισσότερα από 1.500 πρόσθετα που είναι διαθέσιμα για την παροχή σημείων ενοποίησης για την παροχή προσαρμοσμένης λειτουργικότητας κατά την ανάπτυξη λογισμικού. Ένα πολύτιμο εργαλείο αυτοματισμού CI, είναι αρκετά εύκολο στην εγκατάσταση και τη διαμόρφωση. Έχει σχεδιαστεί για να υποστηρίζει κατακευματισμένες ροές εργασίας για επιταχυνόμενες και διαφανείς κατασκευές, δοκιμές και αναπτύξεις σε πλατφόρμες. (www.simplilearn.com/tutorials/devops-tutorial/devops-tools)



Σχήμα 17- Jenkins tools (www.fiverr.com)

Παρακάτω παρουσιάζεται μια συγκριτική μελέτη με το Azure DevOps:

- Ομαδικές Εργασίες: Το Azure επιτρέπει στην ομάδα να εκτελεί μια ακολουθία εργασιών, που έχουν ήδη καθοριστεί σε μια διοχέτευση, σε μια μεμονωμένη εργασία, ενώ το Jenkins γενικά εκτελείται από έναν

μόνο χρήστη που οδηγεί σε προβλήματα παρακολούθησης και λογοδοσίας.

- Διεπαφή YAML: Με το YAML στο Azure Pipelines, μπορεί να διαμορφωθεί η διοχέτευση CI/CD ως κώδικα, ενώ το Jenkins δεν έχει διεπαφή YAML.
- Πλατφόρμα, γλώσσα και cloud: Στο Azure Pipelines, μπορούν να αναπτυχθούν διάφορες εφαρμογές όπως Node.js, android, iOS, java, python και πολλές άλλες και στη συνέχεια οι ομάδες να τις αναπτύξουν είτε σε εσωτερική εγκατάσταση, AWS, Azure ή GCP. Σε ό,τι αφορά το Jenkins, λαμβάνει σενάρια γραμμών που πρέπει να προγραμματιστούν στο Groovy.
- Το Analytics στο Azure Pipelines παρέχεται στο τέλος με δύο παραμέτρους - ρυθμός και διάρκεια εκτέλεσης. Το Jenkins δεν παρέχει αναλυτικά στοιχεία.
- Προσθήκες και Εργασίες: Οι ενσωματωμένες προσθήκες και οι επεκτάσεις μπορούν να ληφθούν από την αγορά του Azure DevOps. Το Jenkins έχει μια μεγάλη γκάμα προσθηκών για να διαλέξει ο χρήστης.
- Η ενσωμάτωση του Azure Pipelines με τη Microsoft είναι εύκολη, αλλά απαιτεί αλλαγές στη διαμόρφωση για να ενσωματωθεί με προϊόντα που δεν ανήκουν στη Microsoft. Το Jenkins, από την άλλη πλευρά, μπορεί εύκολα να τροποποιηθεί και να επεκταθεί.
- Εύκολη υποστήριξη: Δεδομένου ότι το Jenkins είναι ανοιχτού κώδικα, υπάρχει τεράστια υποστήριξη από τις ευέλικτες ομάδες.

Συνοψίζοντας η επιλογή καθορίζεται στην ομάδα ή στη φύση του έργου. Ενώ το Jenkins είναι πιο ευέλικτο στη δημιουργία και την ανάπτυξη περίπλοκων ροών εργασίας, το Azure DevOps προσαρμόζεται πιο γρήγορα. Στις περισσότερες περιπτώσεις, οι οργανισμοί χρησιμοποιούν και τα δύο εργαλεία και σε τέτοιες περιπτώσεις, η Azure Pipelines υποστηρίζει την ενοποίηση με

to Jenkins.(aspresys.com/infrastructure-managed-services/devops/azure-devops-vs-jenkins-who-wins-the-battle/)

3.2.5 Container Platforms: Docker



Το Docker είναι μια πλατφόρμα που έχει σχεδιαστεί για να απλοποιεί τη διαδικασία δημιουργίας, ανάπτυξης, συσκευασίας και αποστολής εφαρμογών μαζί με τα μέρη του, όπως βιβλιοθήκες και άλλα εξαρτήματα. Το Docker είναι ανοιχτού κώδικα και είναι συμβατό με υπηρεσίες cloud όπως το AWS, το GCP και το Azure Cloud. Το Docker τρέχει επίσης σε λειτουργικά συστήματα Windows και Linux. Ένα από τα μεγαλύτερα οφέλη της χρήσης του Docker με το DevOps είναι ότι το χρησιμοποιούν όλοι οι προγραμματιστές, οι δοκιμαστές και οι διαχειριστές συστήματος. Το Docker και το DevOps προωθούν τη συνεργασία μεταξύ διαφόρων ομάδων που εμπλέκονται σε έναν κύκλο ζωής λογισμικού.

Τα κύρια οφέλη από τη χρήση του Docker είναι η συνέπεια και οι δυνατότητες αυτοματοποίησής του. Το Docker λειτουργεί το ίδιο σε πολλές πλατφόρμες, γεγονός που μπορεί να εξαλείψει το σημαντικό άγχος στην αναπτυξιακή πλευρά των πραγμάτων. Εν τω μεταξύ, η ικανότητα του Docker να αυτοματοποιεί τις επαναλαμβανόμενες εργασίες που μαστίζουν τους προγραμματιστές σε καθημερινή βάση είναι επίσης αρκετά ισχυρή. Πιο αναλυτικά:

- Γρήγορη ανάπτυξη, μπορεί να μειώσει την ανάπτυξη σε δευτερόλεπτα. Είναι λόγω του γεγονότος ότι μπορεί να δημιουργήσει ένα κοντέινερ για κάθε διαδικασία και ακόμη και αν δεν εκκινεί ένα λειτουργικό σύστημα.
- Ασφάλεια, το Docker διασφαλίζει ότι οι εφαρμογές που εκτελούνται σε κοντέινερ είναι πλήρως διαχωρισμένες και απομονωμένες μεταξύ τους, από άποψη ασφάλειας, παρέχοντάς τον πλήρη έλεγχο της ροής και της διαχείρισης της κυκλοφορίας.
- Απλότητα και ταχύτερες διαμορφώσεις, ο τρόπος με τον οποίο το Docker απλοποιεί τα θέματα είναι ένα από τα βασικά οφέλη του. Παρέχει ευελιξία στους χρήστες να λαμβάνουν τη δική τους

διαμόρφωση, να την τοποθετούν στον κώδικα και να την αναπτύσσουν περαιτέρω χωρίς προβλήματα.

- Αποδοτικότητα CI, το πλεονέκτημά του είναι η δυνατότητα διαχωρισμού μη εξαρτημένων βημάτων και παράλληλα εκτέλεσης τους. Επιπλέον, η διάρκεια του χρόνου που χρειάζεται από την κατασκευή έως την παραγωγή μπορεί να επιταχυνθεί σημαντικά.
- Συνεχής ενσωμάτωση, ενώ πρόκειται για τη συνεχή ενοποίηση, το Docker λειτουργεί καλά ως μέρος των αγωγών του μαζί με εργαλεία όπως τα Travis, Jenkins και Wercker. Αυτά τα εργαλεία μπορούν να αποθηκεύσουν τη νέα έκδοση ως εικόνα Docker, κάθε φορά που ενημερώνεται ο πηγαίος κώδικας.

(data-flair.training/blogs/advantages-and-disadvantages-of-docker/)

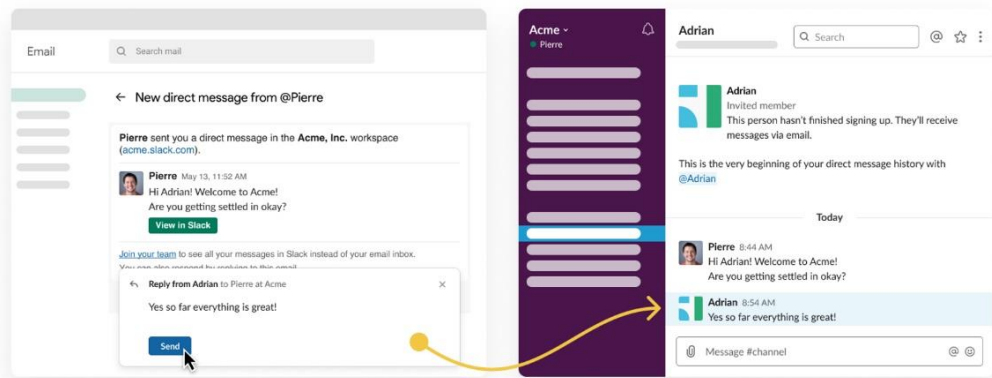
3.2.6 *Communication and Collaboration: Slack*



Ένα από τα πιο δημοφιλή εργαλεία επικοινωνίας και συνεργασίας είναι το Slack. Πρώτα απ' όλα, το Slack προσφέρει δωρεάν, standard και εταιρικές εκδόσεις επί πληρωμή για να καλύψει ένα ευρύ φάσμα πελατών με διαφορετικές ανάγκες. Το Slack είναι ένα αυτόνομο εργαλείο που επιδεικνύει:

- Ισχυρές δυνατότητες αναζήτησης με καλά σχεδιασμένους modifiers αναζήτησης για διευκόλυνση της παρακολούθησης, διαχείρισης και κοινής χρήσης αρχείων.
- Μια φιλική αρχιτεκτονική διαχείρισης έργου ενσωματώνεται με εργαλεία διαχείρισης έργων όπως το Twitter, το Google Hangouts, το Trello και άλλα.
- Ισχυρές δυνατότητες συνεργασίας και επικοινωνίας μέσω κοινόχρηστων καναλιών, απευθείας συνομιλίας, φωνής και τηλεδιάσκεψης.
- Προστέθηκαν λειτουργίες όπως το πρόγραμμα δημιουργίας ροής εργασιών, οι λειτουργίες ειδοποίησης και λήψης σημειώσεων.

Το Slack είναι μια απλοϊκή εφαρμογή με διαισθητική διεπαφή χρήστη και πλήθος προκατασκευασμένων σημείων ενοποίησης που το καθιστούν μια εξαιρετική λύση για τη συμπλήρωση περισσότερων από 900 άλλων επιχειρηματικών εργαλείων. Είναι λειτουργικό από ένα πρόγραμμα περιήγησης ιστού, μπορεί να συγχρονιστεί με τον επιτραπέζιο υπολογιστή και μπορεί να χρησιμοποιηθεί σε κινητές συσκευές όπως tablet και smartphone. Το Slack προσφέρει επίσης εκτεταμένο χώρο αποθήκευσης και μεγάλη γκάμα ενσωματώσεων. (www.simplilearn.com/tutorials/devops-tutorial/devops-tools)



Σχήμα 18- Slack platform (api.slack.com)

Σε αντίθεση με τα email, τα οποία τείνουν να έχουν μια πιο επίσημη αίσθηση και χροιά, το Slack παρέχει μια πιο προσωπική και χαλαρή πλατφόρμα, παρόμοια με το Facebook Messenger ή το WhatsApp. Το Slack παρέχει ενημερώσεις επικοινωνίας σε πραγματικό χρόνο, επομένως οι επείγουσες συνομιλίες ή σημαντικές ειδοποιήσεις μπορούν να πραγματοποιηθούν αμέσως. Αυτό αυξάνει την παραγωγικότητα, καθώς όλοι θα λαμβάνουν ενημερώσεις αυτόματα, αυξάνοντας τον ρυθμό με τον οποίο πραγματοποιούνται οι ενέργειες. Επιπλέον, μέσω της χρήσης του Slack, οι εργαζόμενοι θα αισθάνονται λιγότερη πίεση για να στείλουν μηνύματα που είναι καλά διαμορφωμένα και μελετημένα, καθώς η πλατφόρμα δημιουργεί ένα λιγότερο επίσημο περιβάλλον. Επιπλέον, μπορούν να δημιουργηθούν

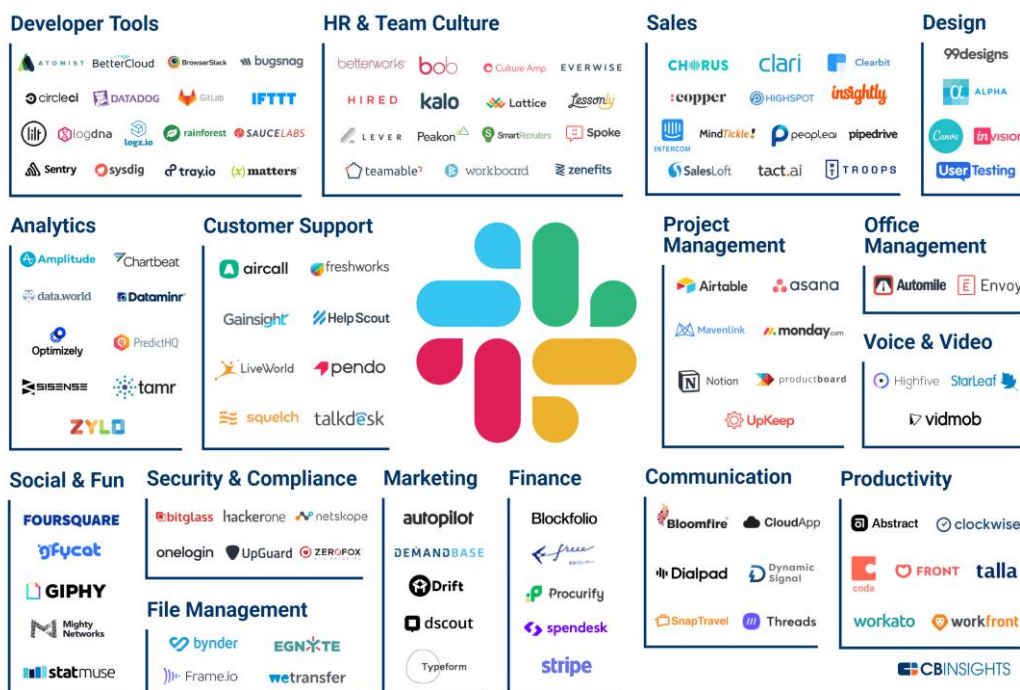
διάφορα κανάλια Slack για διαφορετικές ομάδες ή ομάδες υπαλλήλων για τη δημιουργία συνομιλιών γύρω από ένα κεντρικό θέμα, έργο ή ζήτημα. Το Slack διαθέτει επίσης μια δυνατότητα ChatBot που μπορεί να προστεθεί στα κανάλια Slack. Αυτό το ChatBot μπορεί να διαμορφωθεί ώστε να απαντά εύκολα σε ερωτήσεις και να στέλνει ειδοποιήσεις, σχόλια και αναφορές από ενσωματώσεις με εφαρμογές τρίτων. Δεδομένου ότι όλοι ειδοποιούνται ταυτόχρονα, αυτό μπορεί να βοηθήσει στην έναρξη επιχειρηματικών διαδικασιών με βάση το πλαίσιο. Το Slack είναι μια κεντρική τοποθεσία όπου οι προγραμματιστές και τα μέλη της ομάδας λειτουργιών μπορούν να συναντηθούν και να ανταποκριθούν κατάλληλα στις ειδοποιήσεις που έχουν ληφθεί. Το Slack έχει επίσης τη δυνατότητα να ενσωματωθεί με πολλά από τα πιο δημοφιλή εργαλεία DevOps, όπως τα GitHub, Jenkins CI, Jira και Azure Pipelines. Το Slack είναι σε θέση να ενσωματωθεί απρόσκοπτα στην υπάρχουσα αλυσίδα εργαλείων ανάπτυξης, καθιστώντας ευκολότερο τον εντοπισμό και την επίλυση τυχόν κατασκευών, σφαλμάτων ή γενικών προβλημάτων που ενδέχεται να προκύψουν. Η ενσωμάτωση εφαρμογών τρίτων με το Slack παρέχει συγκεκριμένη λειτουργικότητα στην πλατφόρμα. Για παράδειγμα, η ενσωμάτωση του GitHub και του Slack θα παρέχει αυτόματα ενημερώσεις του GitHub στο Slack. Το πιο συνηθισμένο είναι η ενσωμάτωση του Slack με το Azure Pipelines. Αυτή η ενοποίηση παρέχει στους προγραμματιστές ειδοποίηση καθώς οι εκδόσεις κινούνται σε όλο τον κύκλο ζωής της ανάπτυξης. Οι ειδοποιήσεις μπορούν να ρυθμιστούν για δραστηριότητες όπως ολοκληρωμένες εκδόσεις και εκκρεμείς εγκρίσεις. Αυτό βοηθά στην αύξηση της διαφάνειας στο χώρο εργασίας, αφού όλα τα απαραίτητα μέρη λαμβάνουν ειδοποιήσεις απευθείας στο Slack. Εκτός από τη χρήση του για εσωτερική επικοινωνία και υποθέσεις υποστήριξης πελατών σε έναν οργανισμό, το Slack μπορεί επίσης να χρησιμοποιηθεί ως κοινότητα μεταξύ προγραμματιστών για να παραμένουν ενημερωμένοι για τις πιο πρόσφατες βέλτιστες πρακτικές, ειδήσεις και εκδηλώσεις. Μερικά Developer Slack Groups είναι τα εξής:

- Το **DevOps Chat** διαθέτει κανάλια Slack που καλύπτουν μια ποικιλία θεμάτων, όπως η διαχείριση διαμόρφωσης, η ανάπτυξη και το Docker. Εάν οι προγραμματιστές έχουν μια ερώτηση για ένα θέμα που δεν

μπορεί να απαντηθεί εσωτερικά, αυτά τα φόρουμ χρησιμεύουν ως εργαλείο για τη λήψη απαντήσεων, βέλτιστων πρακτικών ή άλλων τρόπων εξέτασης ενός προβλήματος.

- Το **Kubernetes** είναι ένα ευρέως χρησιμοποιούμενο εργαλείο DevOps και έχει το δικό του κανάλι Slack. Καθώς περισσότερες εταιρείες εισάγουν το Kubernetes στην αλυσίδα εργαλείων τους, περισσότεροι προγραμματιστές είναι πιθανό να έχουν ερωτήσεις σχετικά με το εργαλείο. Το Kubernetes Slack είναι ένας πολύ καλός τρόπος για να λαμβάνονται συμβουλές και απαντήσεις από άλλους στην κοινότητα.
- Το **DevChat Slack** διαθέτει περισσότερα από 50 διαφορετικά κανάλια για να βοηθήσει τους προγραμματιστές λογισμικού να ενωθούν για να λύσουν προβλήματα και να μάθουν ο ένας από τον άλλο. Οι σύνδεσμοι, οι πόροι και οι απαντήσεις κοινοποιούνται σε πραγματικό χρόνο, πράγμα που σημαίνει ότι οι ερωτήσεις μπορούν να απαντηθούν γρήγορα. Αυτό είναι ένα εξαιρετικό Slack για προγραμματιστές που μπορούν να συμμετάσχουν για να μοιραστούν ιδέες με άλλους στην κοινότητα.
- Το **Developers Slack** είναι μια παγκόσμια κοινότητα για

The Slack Integration Ecosystem



Σχήμα 19- Slack ecosystem (www.cbinsights.com)

προγραμματιστές σε όλο τον κόσμο. Τα θέματα του καναλιού περιλαμβάνουν τη δοκιμή, την ανάπτυξη, τον εντοπισμό σφαλμάτων και τα πακέτα. Είναι ένα κανάλι για προγραμματιστές να συνομιλούν με άτομα από όλο τον κόσμο και να μάθουν περισσότερα για τα νέα και τις βέλτιστες πρακτικές που υπάρχουν εκεί έξω.

Καθώς το DevOps έχει να κάνει με την αύξηση της επικοινωνίας μεταξύ Development & Operations, τα κανάλια Slack μπορούν να παρέχουν μια εύκολη κεντρική πλατφόρμα για να συναντηθούν οι ομάδες. Το Slack παρέχει στην ομάδα πρόσβαση στην ίδια παρακολούθηση, ειδοποιήσεις και μετρήσεις. Όπως αναφέρθηκε παραπάνω, πολλές εφαρμογές τρίτων ενσωματώνονται στο Slack. Αυτές οι εφαρμογές μπορούν να παρέχουν ειδοποιήσεις απευθείας στο Slack, διασφαλίζοντας ότι τα ενδιαφερόμενα μέρη και στις δύο πλευρές μπορούν να δουν τις ειδοποιήσεις και να μπορούν να τις συζητήσουν μαζί για να βρουν μια λύση με όλους τους εμπλεκόμενους. Υπάρχουν διάφοροι τρόποι με τους οποίους οι ομάδες ανάπτυξης χρησιμοποιούν ενσωματώσεις Slack και τρίτων για να λαμβάνουν ειδοποιήσεις προκειμένου να αυξήσουν την επικοινωνία μεταξύ ανάπτυξης και λειτουργιών.

- Jira Cloud: Μια ενσωμάτωση μεταξύ του Slack και του Jira θα επιτρέψει την προεπισκόπηση, τη συζήτηση και την έκδοση εισιτηρίων για θέματα του Jira απευθείας στο Slack. Ένα έργο μπορεί να συνδεθεί στο Slack και η Jira θα στείλει αυτόματα ειδοποιήσεις όπως απαιτείται.
- Jenkins CI: Το Jenkins είναι ένα από τα πιο δημοφιλή και πιο χρησιμοποιούμενα εργαλεία στην αγορά. Η ενοποίηση μεταξύ του Slack και του Jenkins θα επιτρέψει τη δημοσίευση ειδοποιήσεων έκδοσης απευθείας σε ένα κανάλι Slack.
- PagerDuty: Ενσωματώνοντας το PagerDuty και το Slack, οι χρήστες μπορούν να λάβουν άμεσα ένα περιστατικό ειδοποίησης που αποστέλλεται απευθείας στο Slack.

(blog.inedo.com/best-practices-slack-devops)

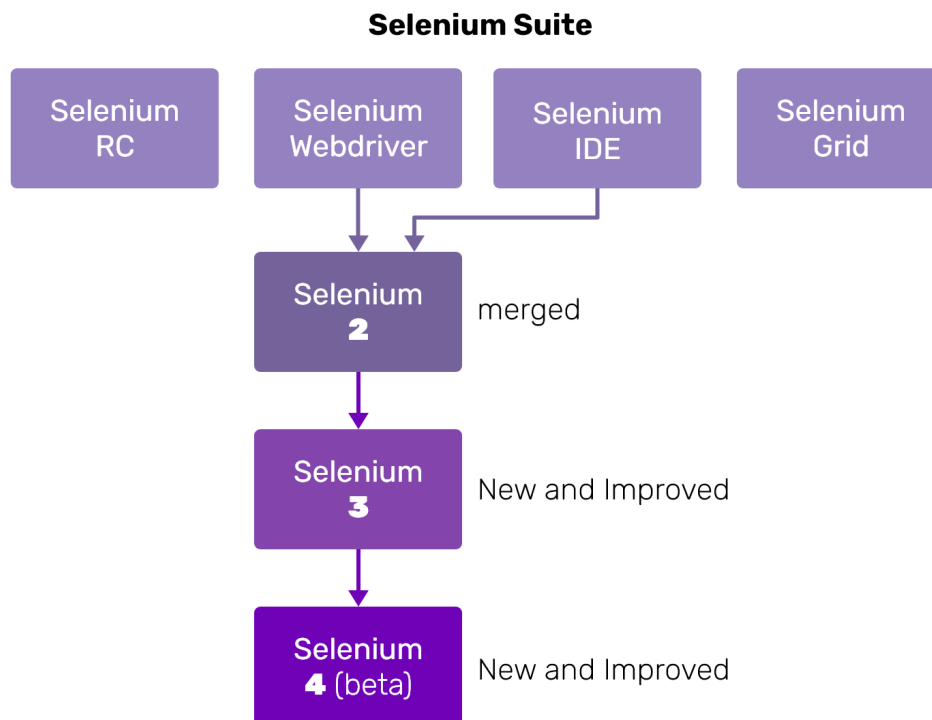
3.2.7 Testing Tool: Selenium



Το σκεπτικό πίσω από την αυτοματοποίηση είναι η εξάλειψη της ανθρώπινης συμμετοχής σε ορισμένες εργασίες χρησιμοποιώντας την τεχνολογία για την αυτόματη εκτέλεση αυτών των εργασιών. Η δοκιμή αυτοματισμού ή η αυτοματοποίηση δοκιμών χρησιμοποιεί εφαρμογές λογισμικού, εκτός από το λογισμικό που αναπτύσσεται, για την αυτόματη εκτέλεση υποθέσεων δοκιμής, τη σύγκριση των αποτελεσμάτων και την αναφορά ελαττωμάτων. Αυτή η διαδικασία εκτελείται ξανά και ξανά για την παροχή λογισμικού υψηλής ποιότητας στους τελικούς χρήστες. Η δοκιμή είναι αναπόσπαστο μέρος της ανάπτυξης λογισμικού και είναι λιγότερο επιρρεπής σε σφάλματα. Το Selenium είναι ένα κορυφαίο πλαίσιο δοκιμών ανοιχτού κώδικα για εφαρμογές Ιστού που υποστηρίζει όλα τα μεγάλα προγράμματα περιήγησης και πλατφόρμες όπως το Linux, τα Windows και το macOS. Η διαφοροποίηση του Selenium είναι ότι ενσωματώνεται με ένα ευρύ φάσμα γλωσσών προγραμματισμού, όπως Python, C#, Ruby, Java, JavaScript, PHP και PERL, και πολλά άλλα πλαίσια δοκιμών αυτοματισμού. Η σουίτα δοκιμών Selenium περιλαμβάνει τα ακόλουθα:

- Selenium IDE (Integrated Development Environment) για δημιουργία και εκτέλεση δοκιμαστικών περιπτώσεων για διερευνητικές δοκιμές και καταγραφή δοκιμαστικών αναπαραγωγών
- Το Selenium client API επιτρέπει στους προγραμματιστές να γράφουν δοκιμαστικά σενάρια απευθείας σε διάφορες γλώσσες προγραμματισμού αντί να πρέπει να τα γράφουν πρώτα σε Selenese γλώσσα προγραμματισμού.
- Το Selenium WebDriver διαθέτει συνδέσμους ειδικά για τη γλώσσα για τη σύνταξη δοκιμαστικών σεναρίων.
- Το Selenium Grid είναι ένας έξυπνος διακομιστής μεσολάβησης που επιτρέπει την εκτέλεση δοκιμών σε πολλά προγράμματα περιήγησης και λειτουργικά συστήματα παράλληλα.
- Ο διακομιστής Selenium Remote Control γραμμένος σε Java επικοινωνεί εντολές Selenium σε προγράμματα περιήγησης για εκτέλεση.

www.simplilearn.com/tutorials/devops-tutorial/devops-tools)



Σχήμα 20- Selenium suite (www.functionize.com/selenium-testing)

Το DevOps έχει σημαντικό αντίκτυπο στις ομάδες δοκιμών. Οι οργανισμοί πρέπει να είναι σχετικοί, απαραίτητοι και να παραμένουν ανταγωνιστικοί. Ο ρόλος του QA αλλάζει μεταξύ των οργανισμών. Ως μέρος της ομάδας δοκιμών διεπαφής χρήστη, οι προγραμματιστές δοκιμών Selenium πρέπει να συγχρονίσουν και να ενορχηστρώσουν τη σχεδίαση και την εκτέλεσή τους σύμφωνα με το χρονοδιάγραμμα και τους κανόνες ενεργοποίησης, που ορίζονται στα εργαλεία ή τις πλατφόρμες συνεχούς ενοποίησης ή συνεχούς παράδοσης. Το Test Design πρέπει να είναι πιο ευέλικτο, αβίαστο και χωρίς σφάλματα. Υπάρχει μια στροφή προς τη βελτίωση των υφιστάμενων ή νέων πλαισίων αυτοματισμού δοκιμών ώστε να ενσωματώνονται απρόσκοπτα με αγωγούς συνεχούς ενοποίησης/συνεχούς παράδοσης. Επιπλέον, οι οργανισμοί αξιοποιούν τη Μηχανική Μάθηση και την Τεχνητή Νοημοσύνη για να αντιμετωπίσουν τις προκλήσεις που αφορούν την πολυπλοκότητα και την κλίμακα στα περιβάλλοντα δοκιμών. Οι επιχειρήσεις εξερευνούν ερευνητικούς τομείς της τεχνητής νοημοσύνης όπως το Computer Vision και η επεξεργασία φυσικής γλώσσας για να αντιμετωπίσουν τις προκλήσεις. Το Selenium είναι

μόνο ένα μικρό μέρος της μεγάλης εικόνας των πρακτικών δοκιμών που υιοθετούνται στο DevOps. Υπάρχουν πολλά περισσότερα εργαλεία όπως το Anthill, το TeamCity, το GitHub Actions και παρόμοιες πλατφόρμες που χρησιμοποιούνται από ομάδες δοκιμών και ανάπτυξης. Ένα πλαίσιο δοκιμής Selenium πρέπει να παρέχει έναν μηχανισμό για την ενεργοποίηση των δοκιμών ή μπορεί να κληθεί κατ' απαίτηση από αυτά τα εργαλεία. Ένα πλαίσιο αυτοματισμού, γενικά, πρέπει να έχει αποτελεσματικούς και έξυπνους τρόπους τεκμηρίωσης των προδιαγραφών και έναν μηχανισμό που να παρέχει ιχνηλασιμότητα μεταξύ των δοκιμών και των προδιαγραφών στις αναφορές. Επομένως, πρέπει να δημιουργούνται εκτελέσιμες προδιαγραφές δοκιμής και να χρησιμοποιούνται εργαλεία κατασκευής όπως τα Gradle, Maven και άλλα παρόμοια εργαλεία. Τέτοια εργαλεία, μαζί με τους πίνακες Kanban και Scrum σε εργαλεία διαχείρισης ευέλικτων δοκιμών, επιτρέπουν να επιτευχθεί υψηλότερη παραγωγικότητα μεταξύ των ομάδων δοκιμών. Η επίτευξη κάποιας ταχύτητας στην παράδοση του λογισμικού είναι επωφελής για τις επιχειρήσεις. (www.softwaretestinghelp.com/devops-in-selenium-testing/)

Το Selenium παραμένει με πολλά εργαλεία δοκιμών που εμφανίζονται στην αγορά κάθε χρόνο και είναι ανταγωνιστικό καθώς:

- Είναι δωρεάν, το Selenium δεν είναι το μόνο εργαλείο δοκιμών αυτοματισμού στην αγορά, αλλά είναι το μόνο δωρεάν που μπορεί να ανταγωνιστεί προϊόντα επί πληρωμή. Αλλά δεν παρέχει τόσο μεγάλη επιλογή γλώσσας, δεν λειτουργεί σε Linux και απλώς δεν έχει fanbase από πίσω. Η κατάσταση ανοιχτού κώδικα του Selenium καθιστά ένα εργαλείο με απότομη καμπύλη εκμάθησης χαμηλό σημείο εισόδου για νεοφυείς επιχειρήσεις και ανεξάρτητους προγραμματιστές. Δεν αποτελεί έκπληξη το γεγονός ότι ακόμη μεγαλύτερες εταιρείες δεν βιάζονται να στραφούν σε επιλογές επί πληρωμή και να προσφέρουν μερικές χιλιάδες δολάρια συν συντήρηση όταν το Selenium εξακολουθεί να αντέχει.
- Είναι ενσωματωμένο με τη ροή εργασιών Agile, DevOps, Continuous Delivery, το Selenium ακολουθούσε την μεθοδολογία Agile και DevOps πριν καν εφαρμοστούν. Η όλη φύση του Selenium εμπίπτει τέλεια στις

βασικές αρχές πίσω από το Agile, το DevOps ή τη Συνεχή Παράδοση. Δεδομένου ότι είναι φορητό σε όλες τις πλατφόρμες και συχνά δεν απαιτεί εκμάθηση νέων γλωσσών, το Selenium επιτρέπει απaráμιλλη ευελιξία. Το Selenium ενσωματώνεται εύκολα με διάφορες πλατφόρμες ανάπτυξης όπως Jenkins, Maven, TestNG, QMetry, SauceLabs κ.λπ. Οι παράλληλες δοκιμές και οι δοκιμές μεταξύ προγραμμάτων περιήγησης με το Selenium Grid και εξωτερικά δίκτυα cloud, όπως το Lambda Test, επιτρέπουν στους προγραμματιστές να λαμβάνουν σχόλια πολύ πιο γρήγορα και να εργάζονται για τις αλλαγές αμέσως.

- Υποστηρίζει δοκιμές για φορητές συσκευές, υπάρχουν δύο κύριες επιλογές το Appium και το Selendroid. Και τα δύο βασίζονται στο Selenium, έτσι ώστε οι προγραμματιστές που είναι ήδη έμπειροι σε αυτό να μπορούν να εφαρμόζουν τις ίδιες αρχές κατά τη δοκιμή εφαρμογών για κινητά. Και τα δύο εργαλεία είναι ανοιχτού κώδικα και έχουν μεγάλη υποστήριξη από την κοινότητα. Η κύρια διαφορά είναι ότι το Appium υποστηρίζει συσκευές iOS, Android και Windows, ενώ το Selendroid επικεντρώθηκε αποκλειστικά στο Android. Το Selendroid συνδυάζεται επίσης με το Appium, έτσι ώστε όταν οι χρήστες δοκιμάζουν τις εκδόσεις Android 2.3 έως 4.3, το πρόγραμμα να μεταβεί αυτόματα στο Selendroid.
- Περιλαμβάνει μεγάλη γκάμα υποστηριζόμενων γλωσσών, πλατφορμών και προγραμμάτων περιήγησης. Η ευελιξία που παρέχει το Selenium είναι σχεδόν απaráμιλλη στον κόσμο του αυτοματισμού δοκιμών. Πρώτον, υπάρχουν δέκα υποστηριζόμενες γλώσσες, μεταξύ των οποίων είναι η Java, η Ruby, η C#, η PHP, η JavaScript και η Python. Είναι επίσης το μόνο βασικό εργαλείο που καλύπτει τη δοκιμή Linux.
- Περιλαμβάνει μεγάλη βιβλιοθήκη με πρόσθετα και επεκτάσεις, το Selenium μπορεί να επεκταθεί πέρα από την τυπική του λειτουργικότητα με ένα ευρύ φάσμα πρόσθετων. Αυτά τα πρόσθετα δεν υπάρχουν μόνο για το WebDriver αλλά και για το Grid και το Selenium IDE. Τα τελευταία είναι ιδιαίτερα εκτεταμένα, ικανά να ενισχύσουν το λειτουργικά αδύναμο εργαλείο και να το κάνουν έτοιμο για παραγωγή.

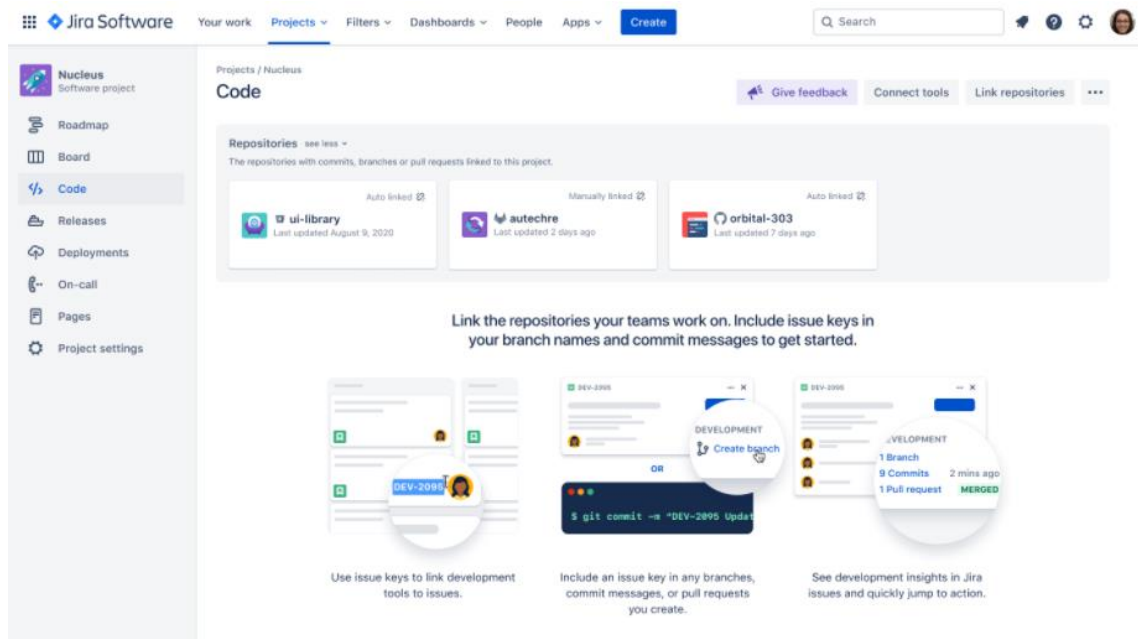
3.2.8 Issue and project tracking platform: Jira



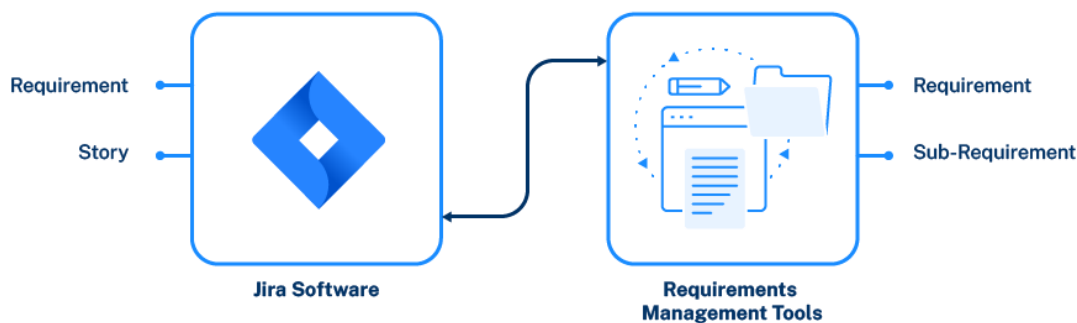
Το Jira είναι μια δημοφιλής πλατφόρμα παρακολούθησης ζητημάτων και διαχείρισης έργων.

Είναι προϊόν της Atlassian, εταιρείας πληροφορικής

που αναπτύσσει πολλά περιζήτητα εργαλεία λογισμικού, όπως το Confluence, το Bitbucket, το Trello, το Statuspage και άλλα. Μπορεί να χρησιμοποιηθεί το Jira είτε ως SaaS είτε ως on-premise. Το Jira ξεκίνησε ως εργαλείο παρακολούθησης σφαλμάτων, έτσι το σκέφτονται ακόμα οι περισσότεροι προγραμματιστές. Η φιλική προς τον χρήστη διεπαφή του επιτρέπει να βλέπει την κατάσταση ανάπτυξης των έργων στο πλαίσιο, να δημιουργεί διακλαδώσεις και αιτήματα έλξης, να προβάλλει δεσμεύσεις, να διαχειρίζεται εκδόσεις, να οπτικοποιεί την πρόοδο και πολλά άλλα. Οι δυνατότητες διαχείρισης έργου προστέθηκαν αργότερα στο Jira και συμπληρώνουν τις δυνατότητες παρακολούθησης σφαλμάτων. Καθώς το Jira εστιάζει στην ανάπτυξη λογισμικού Agile (δηλαδή μεθοδολογίες που επικεντρώνονται στην επαναληπτική ανάπτυξη), συνοδεύεται από πίνακες Scrum και Kanban, χάρτες πορείας, προηγμένα εργαλεία αναφοράς και άλλα. Το Jira διαθέτει επίσης μια προηγμένη μηχανή αυτοματισμού που δίνει τη δυνατότητα στο χρήστη να δημιουργεί κανόνες αυτοματισμού χρησιμοποιώντας μια εύχρηστη διεπαφή μεταφοράς και απόθεσης. Όχι μόνο μπορεί να αυτοματοποιεί τις εργασίες Jira, μπορεί επίσης να συνδεθεί με εργαλεία όπως το Bitbucket, το GitHub ή το Microsoft Teams και να τα συμπεριλάβει στις ροές εργασιών αυτοματισμού. (raygun.com/blog/best-devops-tools/)



Σχήμα 21- Jira platform (docs.convisoappsec.com/integrations/jira/)



Σχήμα 22- Jira software (www.connectall.com/integration/jira-software/)

Κάποια βασικά χαρακτηριστικά της πλατφόρμα είναι τα εξής:

- Simply powerful project configuration, οι ομάδες μπορούν εύκολα να προσαρμόσουν τη ροή εργασίας του έργου τους, τους τύπους θεμάτων και τα πεδία για τον πίνακα που θέλουν και χρειάζονται.
- Built-in roadmaps, παρακολούθηση της προόδου του τρόπου με τον οποίο προχωρούν τα έργα, ώστε να υπάρχει μια σαφή επισκόπηση της εργασίας και των σχεδίων της.
- Customizable dashboard, οι προσαρμοσμένοι πίνακες εργαλείων είναι μια ενιαία υπηρεσία για όλες τις πληροφορίες που χρειάζονται για την οργάνωση έργων και τα επιτεύγματα σε μια ενιαία προβολή.

- Comprehensive Agile reporting, οι ομάδες έχουν πρόσβαση σε περισσότερες από δώδεκα αναφορές με πρακτικές πληροφορίες σε πραγματικό χρόνο σχετικά με τον τρόπο με τον οποίο η ομάδα τους αποδίδει Sprint over Sprint.
- Powerful search, το Jira Software διαθέτει προηγμένες δυνατότητες αναζήτησης που υποστηρίζονται από τη Γλώσσα ερωτημάτων Jira που προσφέρει στις ομάδες λεπτομερείς προβολές της εργασίας τους.
- Native mobile support, με το λογισμικό Jira για iPhone και Android.
- Marketplace with 1,000+ plug-and-play add-ons, πάνω από 1.000 πρόσθετα plug-and-play από το Atlassian Marketplace επεκτείνουν το λογισμικό Jira ώστε να ταιριάζει σε οποιαδήποτε περίπτωση χρήσης ή διαδικασία ανάπτυξης.
- Enterprise-ready, το λογισμικό Jira παρέχει υψηλή διαθεσιμότητα, απόδοση σε κλίμακα και προηγμένα χαρακτηριστικά ασφαλείας, όπως κρυπτογράφηση σε κατάσταση ηρεμίας.

(www.atlassian.com/software/jira/comparison/jira-vs-azure-devops)

	Jira	Azure DevOps
Customizable dashboard	+	+
Backlog Management	3 backlog levels	5 backlog levels
Collaboration	+	+
Kanban	+	+
Sprints	+	+
Portfolio Planning	Paid extension	Free extension
Advanced search capabilities	+	+
Reliability	High	High
Traceability	Not direct	Advanced
Product Roadmapping	+	+
Ease of Deployment	Easy	Easy
Quality of technical support	Good	Satisfactory
Integration with third-party	+	+

tools		
Marketplace with add-ons	+	+
Enterprise-ready	+	+
Ease of integration using Standard APIs and Tools	Easy	Easy, but depends on the tool

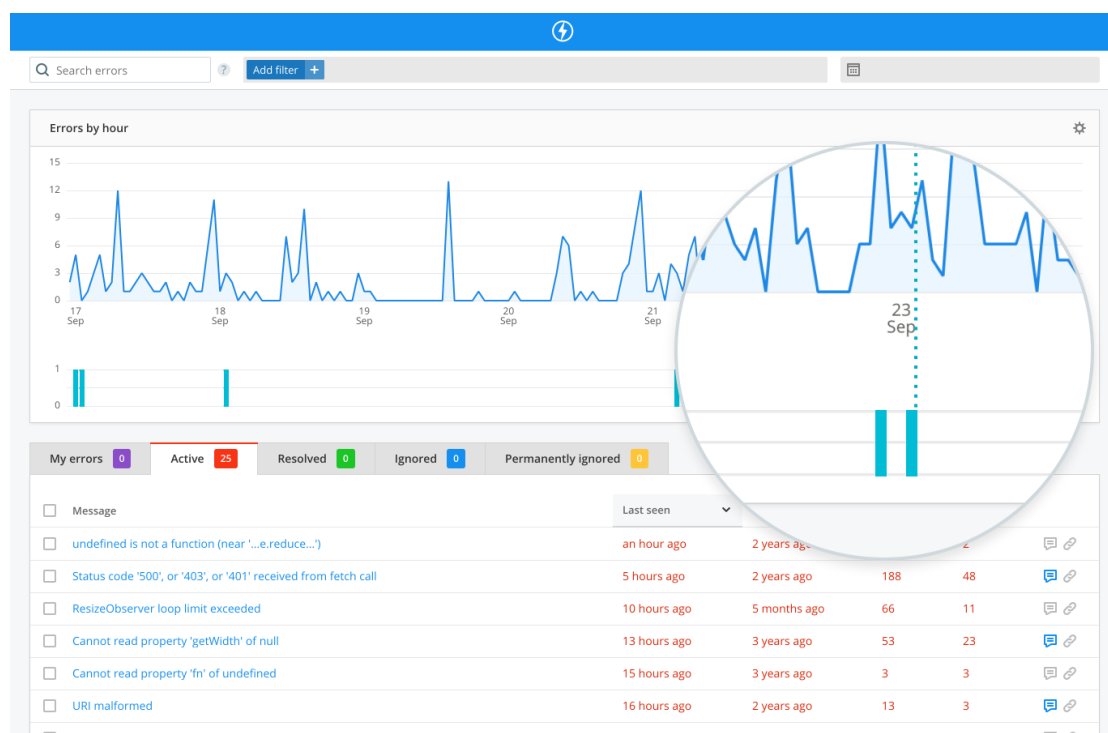
Σχήμα 23 – Features of Azure and Jira (www.atlassian.com)

3.2.9 Error monitoring and crash reporting platform: Raygun



Το Raygun είναι λογισμικό εφαρμογών, το οποίο χρησιμοποιείται για την εύρεση και την επίλυση προβλημάτων απόδοσης που σχετίζονται με τις εφαρμογές. Το Raygun λειτουργεί ως κεντρικό μέρος για την παρακολούθηση ολόκληρης της στοίβας για προβλήματα που επηρεάζουν τους τελικούς χρήστες και επίσης, παρέχει ένα πλήρες σύνολο προβλημάτων καθώς και εργαλεία ροής εργασιών που επιτρέπουν στον χρήστη να διορθώνει τα προβλήματα γρήγορα, τείνει να βελτιώνει τις εμπειρίες του τελικού χρήστη. Το Raygun έχει μια δυνατότητα που ονομάζεται αναφορά σφαλμάτων, η οποία καταγράφει, αναλύει, παρακολουθεί και επιλύει αυτόματα το σφάλμα. Κάνει τους χρήστες αρκετά ικανούς να κατανοήσουν πώς οι τελικοί χρήστες αντιμετωπίζουν τις εφαρμογές τους, κάνοντάς τους να ανακαλύψουν τις βασικές αιτίες της αργής φόρτωσης σελίδων και άλλων προβλημάτων απόδοσης που αντιμετωπίζουν συνήθως οι τελικοί χρήστες. Το Raygun υποστηρίζει διάφορες γλώσσες προγραμματισμού για επιτραπέζιους υπολογιστές, κινητά και Web, όπως Java, Scala, .NET, Python, PHP, iOS, Android και JavaScript. Αυτό το λογισμικό μπορεί να παρακολουθεί τις εφαρμογές και να εξυπηρετεί τις πλήρεις περιόδους λειτουργίας και τα ιστορικά σφαλμάτων κάθε χρήστη. Το Raygun επιτρέπει στους τελικούς χρήστες να παρακολουθούν τις αναπτύξεις και τις εκδόσεις των εφαρμογών τους. Επίσης, οι χρήστες μπορούν να επαληθεύσουν εάν έχουν επιδιορθώσει τα προβλήματα και να εντοπίσουν νέα ζητήματα που εισάγονται σε κάθε

ανάπτυξη και κυκλοφορία. Το Raygun βοηθά την ομάδα ανάπτυξης να ανακαλύψει και να αναδημιουργήσει τα ζητήματα που επηρεάζουν τους τελικούς χρήστες με ακρίβεια και μεγαλύτερη ταχύτητα. Λιγότερο από το 1% των πελατών αναφέρουν ένα σφάλμα λογισμικού που αντιμετωπίσαν και το Raygun βρίσκει αυτά τα λάθη και αποτρέπει μελλοντικά προβλήματα λογισμικού. Το Raygun κατηγοριοποιεί τα σφάλματα που σχηματίζονται σε ομάδες. Στη συνέχεια, αυτές οι ομάδες ταξινομούνται σε ενεργές, υπό επίλυση, επιλυμένες σε μια συγκεκριμένη έκδοση, αγνοημένες και μόνιμα αγνοημένες, έτσι ώστε ο πελάτης να μπορεί να αναλύσει την κατάσταση του σφάλματος και τον τρόπο αντιμετώπισης ή επίλυσης του προβλήματος. Λαμβάνουν προειδοποιήσεις/ειδοποιήσεις ανάλογα με αυτές τις ομάδες σφαλμάτων επίσης.



Σχήμα 24 – Raygun platform (www.raygun.com)

Οι λόγοι που κάνουν το Raygun να ξεχωρίζει είναι οι ακόλουθοι:

- Αναζήτηση πλήρους κειμένου, ομάδες σφαλμάτων και όλα τα δεδομένα που συλλέγονται έχουν δυνατότητα αναζήτησης.
- Ομαδοποίηση σφαλμάτων, κάθε εμφάνιση σφάλματος παρουσιάζεται σε μία ομάδα με πρόσβαση σε μεμονωμένες παρουσίες του, συμπεριλαμβανομένου του ίχνους στοίβας.

- Προβολή δραστηριότητας εφαρμογής, εμφανίζει κάθε ενέργεια σε μια ομάδα σφαλμάτων.
- Επηρεασμένοι χρήστες, ο αριθμός των επηρεαζόμενων χρηστών εμφανίζεται να αντιστοιχεί σε κάθε σφάλμα.
- Εξωτερικές ενσωματώσεις, η εφαρμογή μπορεί να ενσωματωθεί με εξωτερικές ενσωματώσεις όπως GitHub, Bitbucket, Asana, JIRA κ.λπ.
- Οπτικοποιήσεις και επίπεδο λεπτομέρειας, παρέχει βαθύτερο επίπεδο λεπτομέρειας σε σύγκριση με άλλα εργαλεία παρακολούθησης σφαλμάτων και παρέχει καλές απεικονίσεις γραφημάτων.

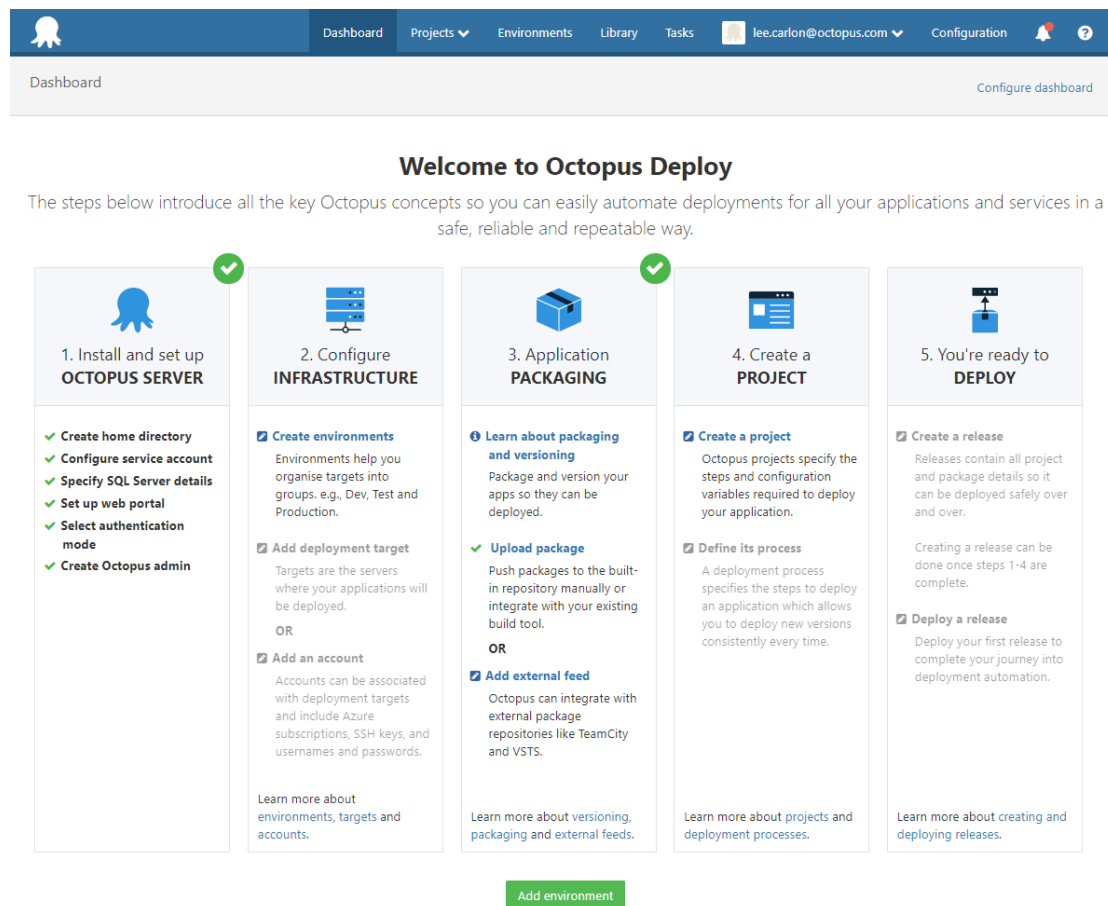
(<https://www.supportpro.com/blog/raygun/>)

3.2.10 Automated deployment and release management platform:

Octopus Deploy



Το Octopus Deploy είναι ένα αυτοματοποιημένο εργαλείο ανάπτυξης που μπορεί να ενσωματωθεί με τις περισσότερες διαδικασίες δημιουργίας κώδικα για την ανάπτυξη και τη διαμόρφωση εφαρμογών. Αυτό βοηθά την ομάδα DevOps να επιτύχει τη μέγιστη απόδοση καταργώντας την επαναλαμβανόμενη χειρωνακτική εργασία και τις συχνές αλλαγές στη διαμόρφωση της εφαρμογής, ενώ προωθεί μια εφαρμογή από το ένα περιβάλλον στο άλλο. Το Octopus Deploy χωρίζεται σε δύο μέρη. Πρώτα είναι ο διακομιστής Octopus όπου προστίθεται και διαμορφώνεται η εφαρμογή ή η υπηρεσία που πρόκειται να αναπτυχθεί. Το δεύτερο είναι το Octopus Tentacles, το οποίο είναι εγκατεστημένο στο σύνολο μηχανημάτων όπου θα πρέπει να αναπτυχθούν οι εφαρμογές που έχουν ρυθμιστεί στον Octopus Server. Τόσο ο διακομιστής Octopus όσο και τα πλοκάμια εγκαθίστανται ως υπηρεσία. Το Octopus μπορεί να εγκατασταθεί σε διάφορες πλατφόρμες, συμπεριλαμβανομένων των Windows, Mac και Linux. Μπορεί επίσης να ενσωματωθεί με τα περισσότερα εργαλεία ελέγχου εκδόσεων, συμπεριλαμβανομένων των VSTS και GIT. Το Octopus υποστηρίζει ένα ευρύ φάσμα ανάπτυξης εφαρμογών από εφαρμογές Web έως εφαρμογές για κινητά. Επιτρέπει επίσης την αυτοματοποιημένη ανάπτυξη εφαρμογών σε πλατφόρμα cloud, συμπεριλαμβανομένων των AWS και Azure.



Σχήμα 25 – Octopus platform (www.octopus.com)

Τα πλεονεκτήματα χρήσης του Octopus Deploy είναι τα ακόλουθα:

- Το Octopus χρησιμοποιεί τα ίδια πακέτα ανάπτυξης σε όλα τα περιβάλλοντα και διευκολύνει την παρακολούθηση της προόδου τους σε αυτά τα περιβάλλοντα.
- Διευκολύνει τον χειρισμό της διαμόρφωσης σε περιβάλλοντα που είναι συνήθως μια ιδιαίτερα δύσκολη εργασία.
- Χρησιμοποιείται για να διευκολύνει την ανάπτυξη πολλών εφαρμογών σε πολλούς διακομιστές.
- Το Octopus Deploy καθιστά πολύ εύκολη την ανάπτυξη εφαρμογών σε έναν ή περισσότερους διακομιστές με το πάτημα ενός κουμπιού.

- Υποστηρίζονται πολλές διαφορετικές πλατφόρμες, γλώσσες και λειτουργικά συστήματα. Μπορεί να εγκατασταθεί στον διακομιστή ή στο cloud. Μπορεί να αναπτυχθεί σε Windows, Linux κ.λπ.
- Περιλαμβάνονται πολλά διαφορετικά "πρότυπα βημάτων", τα οποία καθιστούν πολύ εύκολη την ανάπτυξη αυτού που θέλει το έργο, όπως η ανάπτυξη μέσω SSH, FTP κ.λπ.(www.devopsschool.com/blog/what-is-octopus-deployment/)

Σε σύγκριση με την πλατφόρμα Azure DevOps αναφορικά με τα features έχουμε:

<i>Features</i>	<i>Azure DevOps</i>	<i>Octopus</i>
<i>Approval Workflow</i>	+	+
<i>Change Management</i>	×	×
<i>Collaboration Tools</i>	+	×
<i>Configuration Management</i>	×	+
<i>Continuous Deployment</i>	+	+
<i>Continuous Integration</i>	+	×
<i>Dashboard</i>	+	+
<i>Monitoring</i>	×	×
<i>Prioritization</i>	+	×
<i>Release Management</i>	+	+
<i>Reporting/Analytics</i>	×	+
<i>Testing Management</i>	×	×
<i>Third-Party Integrations</i>	×	+

Σχήμα 26 – Features of azure and octopus (www.capterra.com)

3.3 Συγκριτική ανάλυση των πλατφορμών DevOps

Εν συνεχεία της μελέτης των εργαλείων DevOps, δημιουργήθηκε ένας πίνακας σύγκρισης σύμφωνα με τα κοινά λειτουργικά χαρακτηριστικά τους που περιεγράφηκαν παραπάνω. Τα στοιχεία που λήφθηκαν υπόψιν είναι:

- Τα licenses
- Οι υποστηριζόμενες πλατφόρμες
- Εάν απαιτούνται εξαρτήσεις άλλων εφαρμογών
- Η ασφάλεια
- Η υποστήριξη του προμηθευτή και η δυνατότητα documentation
- Εάν υπάρχει διαθεσιμότητα σε cloud

	License	Supported Platform	Dependencies	Security	Vendor Support & Documentation	Availability over Cloud
AWS	Free Tier	Windows, Linux	NA	Yes	Yes	Yes
Azure	Licensed	Windows	NA	Yes	Yes	Yes
Git	Free	Linux, Windows, OS X, Solaris	NA	Moderate	Yes	Yes
Jenkins	Free	Windows, macOS, Unix	NA	Moderate	Yes	Yes
Docker	Licensed	Windows, Linux	NA	Yes	Yes	Yes
Slack	Free, Licensed	Windows, macOS, Linux, Android,	NA	Moderate	Yes	Yes

		iOS				
Selenium	Free	Windows, macOS, Linux, Android, iOS	NA	Moderate	Yes	Yes
Jira	Licensed	Windows, macOS, Linux, Android, iOS	NA	Yes	Yes	Yes
Raygun	Licensed	Windows, macOS, Linux, Android, iOS	NA	Moderate	Yes	Yes
Octopus	Licensed	Windows, macOS, Linux	Διακομιστής Octopus και Octopus Tentacles	Moderate	Yes	Yes

Σχήμα 27- Συγκριτική Ανάλυση

3.3.1 DevOps Market

Η συνεχής ενσωμάτωση καινοτόμων τεχνολογιών όπως η μηχανική μάθηση (ML) και η τεχνητή νοημοσύνη (AI) για την παροχή πιο κλιμακωτών και αξιόπιστων πλατφορμών και λύσεων DevOps εκτιμάται ότι θα δημιουργήσει τεράστιες ευκαιρίες ανάπτυξης για την παγκόσμια αγορά τα επόμενα χρόνια. Για παράδειγμα, τεχνικοί γίγαντες όπως η Microsoft, η IBM και η Google επικεντρώνονται στον τρόπο καινοτομίας των συστημάτων DevOps μέσω ML, ανάλυσης blockchain, 5G, AI, κβαντικού υπολογισμού και τεχνολογιών αιχμής για διάφορες δραστηριότητες, όπως λύσεις συνεχούς ανάπτυξης, δοκιμές

απόδοσης εφαρμογών, λογισμικό πρόβλεψη δοκιμών και ανάπτυξης και πολλά άλλα. Αυτοί οι παράγοντες εκτιμάται ότι θα δώσουν ώθηση στην ανάπτυξη της αγοράς τα επόμενα χρόνια.



Σχήμα 28-DevOps Market (www.researchdive.com)

Η παγκόσμια αγορά DevOps για την παρακολούθηση και τη διαχείριση απόδοσης κατέγραψε κυρίαρχο μερίδιο τα προηγούμενα χρόνια και προβλέπεται να συνεχίσει τη σταθερή ανάπτυξή της και να φτάσει τα 6.410,3 εκατομμύρια δολάρια έως το 2027. Αυτή η κυριαρχία οφείλεται στην ευρεία χρήση των εργαλείων DevOps συμπεριλαμβανομένων των διακομιστών εφαρμογών και ιστού, των δικτύων cloud, των βάσεων δεδομένων και πολλών άλλων. Επιπλέον, η συνεχής παρακολούθηση της συμπεριφοράς των πελατών είναι ένα ουσιαστικό στοιχείο για την απόκτηση έγκαιρων γνώσεων σχετικά με τους χρόνους απόκρισης και τις τάσεις χρήσης για την παροχή πλήρους ικανοποίησης των πελατών. Έτσι, το DevOps θα διαδραματίσει σημαντικό ρόλο στην παρακολούθηση της εμπειρίας των πελατών μέσω της διαχείρισης απόδοσης υπηρεσιών, η οποία προβλέπεται να προωθήσει την ανάπτυξη της αγοράς κατά τα προβλεπόμενα έτη. (www.researchdive.com)

4

Πλατφόρμα Microsoft

Azure DevOps

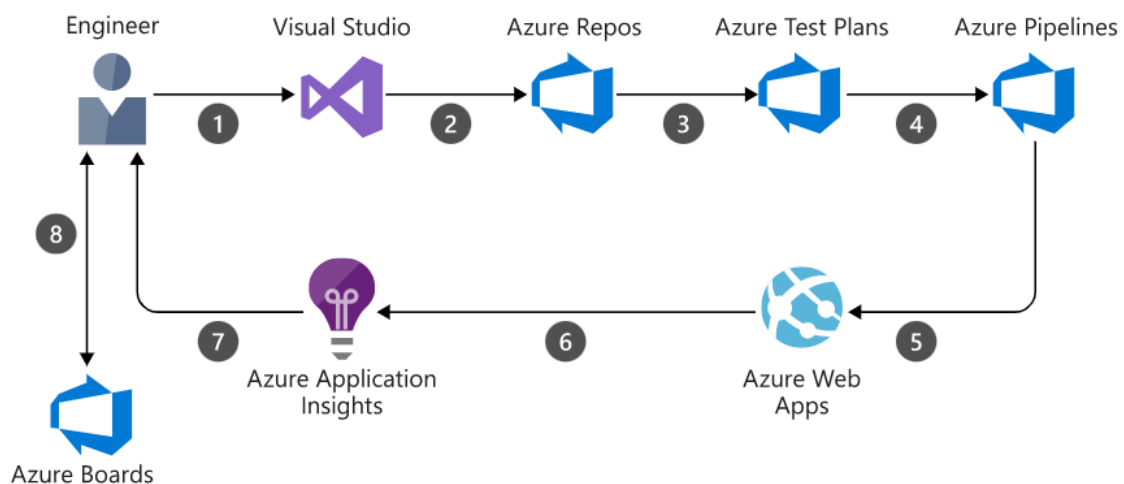
4.1 Αρχιτεκτονική του Azure DevOps

Το Azure DevOps είναι μια πλατφόρμα λογισμικού ως υπηρεσία (SaaS) από τη Microsoft που παρέχει μια αλυσίδα εργαλείων DevOps από άκρο σε άκρο για την ανάπτυξη και την ανάπτυξη λογισμικού. Επίσης, ενσωματώνεται με τα περισσότερα κορυφαία εργαλεία της αγοράς και είναι μια εξαιρετική επιλογή για την ενορχήστρωση μιας αλυσίδας εργαλείων DevOps. Στην αρχιτεκτονική DevOps, όλες οι απαραίτητες ομάδες εργάζονται ταυτόχρονα και κυκλικά παρέχοντας έναν βρόχο παρακολούθησης ανατροφοδότησης στην πορεία. Η προσέγγιση διαχείρισης έργου DevOps βασίζεται σε μια βάση για την επίτευξη μεγαλύτερης ευελιξίας μέσω της επικοινωνίας και της συνεργασίας. Τα δεδομένα ρέουν μέσω του σεναρίου ως εξής:

1. Ένας προγραμματιστής αλλάζει τον πηγαίο κώδικα της εφαρμογής.
2. Ο κώδικας εφαρμογής, συμπεριλαμβανομένου του αρχείου web.config, δεσμεύεται στο χώρο αποθήκευσης πηγαίου κώδικα στο Azure Repos.
3. Η συνεχής ενσωμάτωση ενεργοποιεί τη δημιουργία εφαρμογών και τις δοκιμές μονάδας χρησιμοποιώντας σχέδια δοκιμών Azure.

4. Η συνεχής ανάπτυξη εντός του Azure Pipelines ενεργοποιεί μια αυτοματοποιημένη ανάπτυξη artifacts εφαρμογών με τιμές διαμόρφωσης ειδικών για το περιβάλλον.
5. Τα artifacts αναπτύσσονται στην υπηρεσία Azure App Service.
6. Το Azure Application Insights συλλέγει και αναλύει δεδομένα απόδοσης και χρήσης.
7. Οι προγραμματιστές παρακολουθούν και διαχειρίζονται πληροφορίες απόδοσης και χρήσης.
8. Οι πληροφορίες Backlog χρησιμοποιούνται για την ιεράρχηση νέων λειτουργιών και διορθώσεων σφαλμάτων χρησιμοποιώντας τα Azure Boards

(docs.microsoft.com/en-gb/azure/architecture/example-scenario/apps/devops-dotnet-webapp)



Σχήμα 29 – Azure DevOps architecture (dos.microsoft.com)

4.2 Υπηρεσίες του Azure DevOps

Το Azure DevOps παρέχει υπηρεσίες προγραμματιστών που επιτρέπουν στις ομάδες να σχεδιάζουν εργασίες, να συνεργάζονται στην ανάπτυξη κώδικα και να δημιουργούν και να αναπτύσσουν εφαρμογές. Το Azure DevOps υποστηρίζει μια συλλογική κουλτούρα και ένα σύνολο διαδικασιών που συγκεντρώνουν προγραμματιστές, διαχειριστές έργων και συντελεστές για την

ανάπτυξη λογισμικού. Επιτρέπει στους οργανισμούς να δημιουργούν και να βελτιώνουν προϊόντα με ταχύτερο ρυθμό από ό,τι μπορούν με τις παραδοσιακές προσεγγίσεις ανάπτυξης λογισμικού. Οι ομάδες μπορούν να εργαστούν στο cloud χρησιμοποιώντας τις Υπηρεσίες Azure DevOps ή εντός εγκατάστασης χρησιμοποιώντας τον διακομιστή Azure DevOps. Παρέχει ενσωματωμένες δυνατότητες στις οποίες μπορούν να έχουν πρόσβαση μέσω του προγράμματος περιήγησης ιστού ή του προγράμματος-πελάτη. Παρακάτω παρουσιάζονται αναλυτικά οι αυτόνομες υπηρεσίες που περιλαμβάνει. (azure.microsoft.com/en-us/services/devops/)



Σχήμα 30 – Azure DevOps services (dos.microsoft.com)

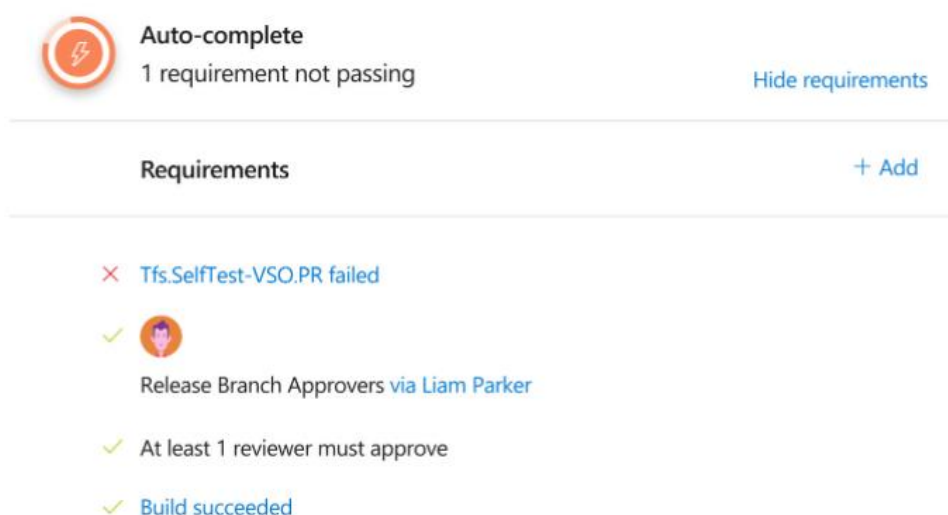
4.2.1 Azure Repos



Το Azure Repos είναι ένα σύνολο εργαλείων ελέγχου έκδοσης που χρησιμοποιούνται για την διαχείριση του κώδικα. Τα συστήματα ελέγχου έκδοσης είναι λογισμικό που βοηθά στην παρακολούθηση των αλλαγών που γίνονται στον κώδικά με την πάροδο του χρόνου. Καθώς γίνεται επεξεργασία του κώδικα, το σύστημα ελέγχου έκδοσης επιτρέπει στον χρήστη να λαμβάνει ένα στιγμιότυπο των αρχείων. Το σύστημα ελέγχου έκδοσης αποθηκεύει μόνιμα αυτό το στιγμιότυπο, ώστε να μπορεί η ομάδα να το ανακαλέσει αργότερα, εάν το χρειάζεται. Ο έλεγχος έκδοσης διατηρεί ένα ιστορικό της ανάπτυξης, ώστε η ομάδα να μπορεί να ελέγξει και ακόμη και να επιστρέψει σε οποιαδήποτε

έκδοση του κώδικά με ευκολία. Το Azure Repos παρέχει δύο τύπους ελέγχου έκδοσης:

- **Git:** είναι ένα σύστημα ελέγχου κατανεμημένων εκδόσεων. Κάθε προγραμματιστής έχει ένα αντίγραφο του αποθετηρίου πηγής. Οι προγραμματιστές μπορούν να πραγματοποιήσουν κάθε σύνολο αλλαγών στο μηχάνημα τους και να εκτελέσουν λειτουργίες ελέγχου εκδόσεων, όπως το ιστορικό και τη σύγκριση χωρίς σύνδεση δικτύου. Όταν χρειάζεται να αλλάξουν περιβάλλοντα, μπορούν να δημιουργήσουν ένα ιδιωτικό τοπικό branch. Μπορούν να μεταβούν γρήγορα από τον έναν κλάδο στον άλλο για να περιστρέφουν μεταξύ διαφορετικών παραλλαγών της βάσης κωδικών τους. Αργότερα, μπορούν να συγχωνεύσουν, να δημοσιεύσουν ή να διαθέσουν το branch.
- **Team Foundation Version Control (TFVC):** είναι ένα κεντρικό σύστημα ελέγχου έκδοσης. Συνήθως, τα μέλη της ομάδας έχουν μόνο μία έκδοση κάθε αρχείου στους υπολογιστές τους. Τα ιστορικά δεδομένα διατηρούνται μόνο στον διακομιστή. Οι κλάδοι βασίζονται σε μονοπάτια και δημιουργούνται στον διακομιστή. (azure.microsoft.com/en-us/services/devops/)

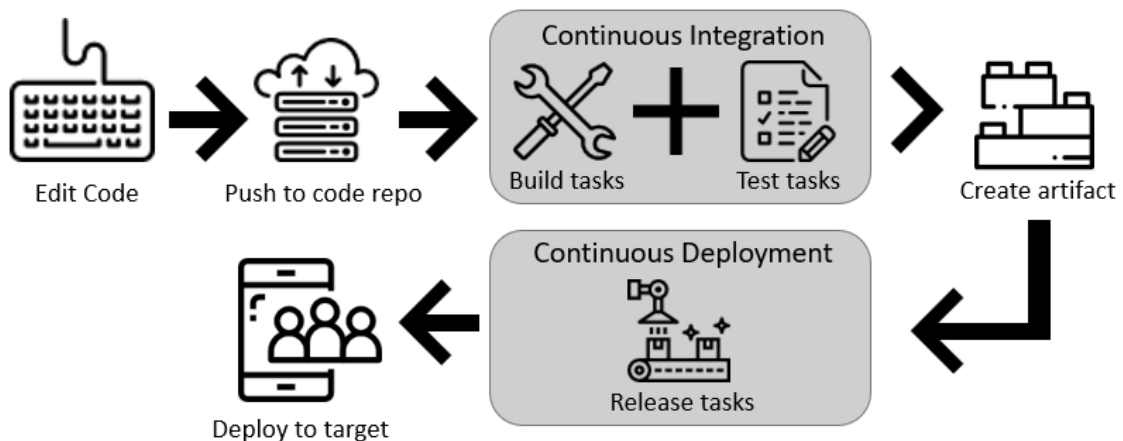


Σχήμα 31- Azure Repos (dev.azure.com/)

4.2.2 Azure Pipelines



Το Azure Pipelines δημιουργεί και δοκιμάζει αυτόματα έργα κώδικα για να τα κάνει διαθέσιμα σε άλλους. Λειτουργεί με σχεδόν οποιαδήποτε γλώσσα ή τύπο έργου. Το Azure Pipelines συνδυάζει συνεχή ενοποίηση (CI) και συνεχή παράδοση (CD) για να δοκιμάσει και να δημιουργήσει τον κώδικά και να τον στείλει σε οποιονδήποτε στόχο. Η συνεχής ολοκλήρωση (CI) είναι η πρακτική που χρησιμοποιούν οι ομάδες ανάπτυξης της αυτοματοποίησης συγχώνευσης και δοκιμής κώδικα. Η εφαρμογή CI βοηθά να εντοπιστούν σφάλματα νωρίς στον κύκλο ανάπτυξης, γεγονός που καθιστά λιγότερο δαπανηρή τη διόρθωσή τους. Οι αυτοματοποιημένες δοκιμές εκτελούνται ως μέρος της διαδικασίας CI για τη διασφάλιση της ποιότητας. Τα τεχνουργήματα παράγονται από συστήματα CI και τροφοδοτούνται για την απελευθέρωση διεργασιών για την επίτευξη συχνών αναπτύξεων. Η υπηρεσία Build στον διακομιστή Azure DevOps βοηθά την ομάδα να ρυθμίσει και να διαχειριστεί το CI για τις εφαρμογές της. Η συνεχής παράδοση (CD) είναι μια διαδικασία με την οποία δημιουργείται, δοκιμάζεται και αναπτύσσεται κώδικας σε ένα ή περισσότερα περιβάλλοντα δοκιμής και παραγωγής. Η ανάπτυξη και η δοκιμή σε πολλαπλά περιβάλλοντα αυξάνει την ποιότητα. Τα συστήματα CI παράγουν τεχνουργήματα που μπορούν να αναπτυχθούν, συμπεριλαμβανομένων των υποδομών και των εφαρμογών. Οι διαδικασίες αυτόματης έκδοσης καταναλώνουν αυτά τα τεχνουργήματα για την έκδοση νέων εκδόσεων και επιδιορθώσεων σε υπάρχοντα συστήματα. Τα συστήματα παρακολούθησης και ειδοποίησης λειτουργούν συνεχώς για να αυξήσουν την ορατότητα σε ολόκληρη τη διαδικασία CD. Το Continuous Testing (CT) στις εγκαταστάσεις ή στο cloud είναι η χρήση αυτοματοποιημένων ροών εργασιών κατασκευής-ανάπτυξης-δοκιμών, με μια επιλογή τεχνολογιών και πλαισίων, που δοκιμάζουν τις αλλαγές σας συνεχώς με γρήγορο, επεκτάσιμο και αποτελεσματικό τρόπο. Οι χρήστες μπορούν να χρησιμοποιήσουν πολλές γλώσσες με το Azure Pipelines, συμπεριλαμβανομένων των Python, Java, JavaScript, PHP, Ruby, C#, C++ και Go. (azure.microsoft.com/en-us/services/devops/)

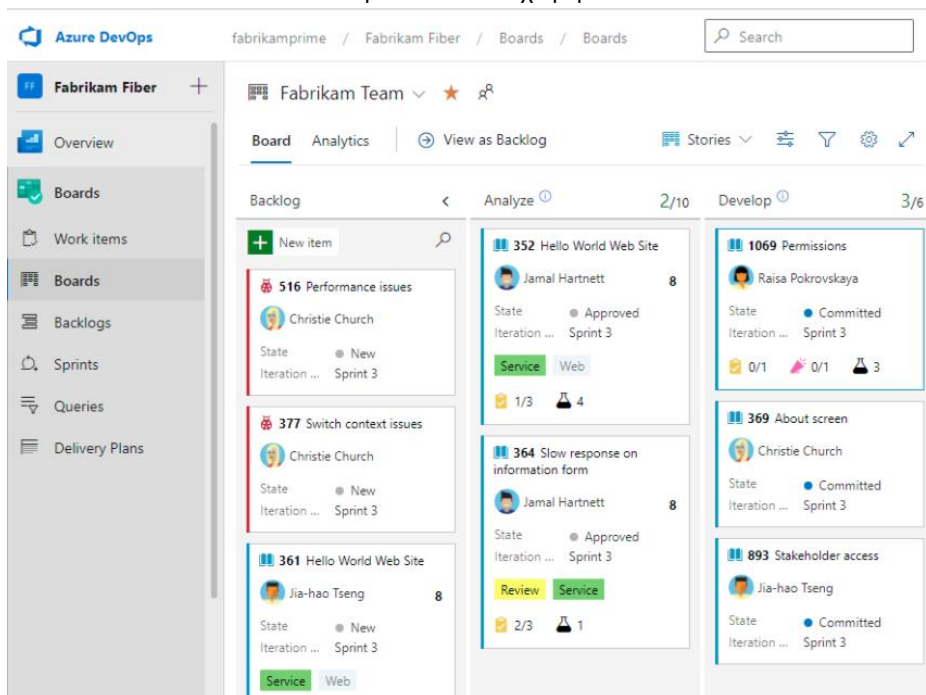


Σχήμα 32- Azure Pipelines (dos.microsoft.com)

4.2.3 Azure Boards



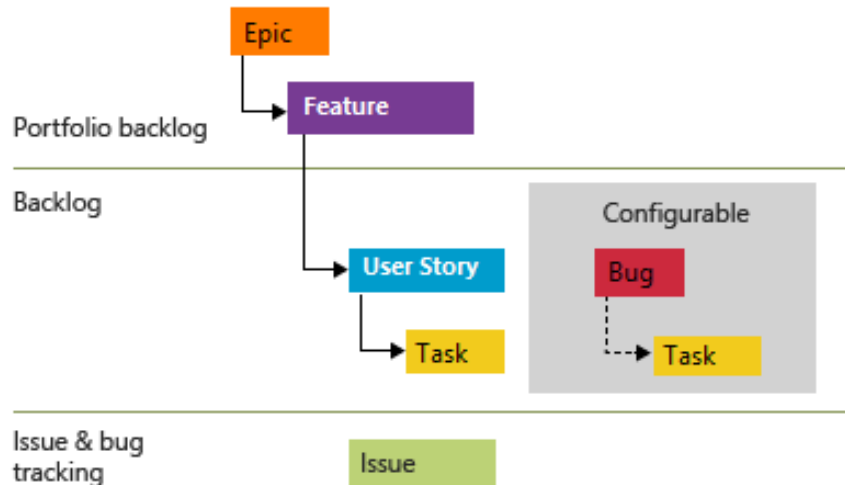
Το Azure Boards παρέχει στις ομάδες ανάπτυξης λογισμικού τα διαδραστικά και προσαρμόσιμα εργαλεία που χρειάζονται για τη διαχείριση των έργων λογισμικού. Παρέχει ένα πλούσιο σύνολο δυνατοτήτων, συμπεριλαμβανομένης της εγγενούς υποστήριξης για διαδικασίες Agile, Scrum και Kanban, προβολές ημερολογίου, διαμορφώσιμους πίνακες εργαλείων και ενσωματωμένες αναφορές. Αυτά τα εργαλεία κλιμακώνονται καθώς η επιχείρησή μεγαλώνει. Ο πίνακας Kanban, είναι μόνο ένα από τα πολλά εργαλεία που επιτρέπει στους χρήστες να προσθέτουν, να ενημερώνουν και να φιλτράρουν ιστορίες, σφάλματα, features και epics. Κάθε φόρμα στοιχείου εργασίας παρέχει ένα τυπικό σύνολο πεδίων συστήματος και στοιχείων ελέγχου, συμπεριλαμβανομένης της συζήτησης για την προσθήκη και παρακολούθηση σχολίων, το ιστορικό, τους συνδέσμους και τα συνημμένα. Τα στοιχεία ελέγχου ανάπτυξης και σχετικών εργασιών υποστηρίζουν την παρακολούθηση κατά την release ή την αλλαγή κώδικα και τις σχέσεις μεταξύ των στοιχείων εργασίας. (azure.microsoft.com/en-us/services/devops/)



Σχήμα 33- Azure boards (dos.microsoft.com)

Το Azure Boards παρέχει μια επιλογή από εργαλεία προγραμματισμού Agile, πολλά από τα οποία λειτουργούν σε συνδυασμό μεταξύ τους. Μερικές από τις δυνατότητες είναι οι ακόλουθες:

- Διαμόρφωση των ομάδων για να υποστηρίζουν τη συγκέντρωση ιστοριών χρηστών ανάπτυξης σε λειτουργίες διαχείρισης έργου
- Ανάθεση εργασίας σε sprint
- Χρήση των ιστοριών και των δυνατοτήτων χρήστη για την παρακολούθηση των παραδοτέων
- Χρήση των tags για την υποστήριξη ερωτημάτων και φιλτραρίσματος τους
- Διαχείριση των εξαρτημένων εργασιών συνδέοντας τα μεταξύ τους
- Έλεγχος της προόδου και των παραδοτέων χρησιμοποιώντας τα σχέδια ανεκτέλεστων λειτουργιών, συνάθροισης και παράδοσης (azure.microsoft.com/en-us/services/devops/)



Σχήμα 34- Agile programming in Azure DevOps (dos.microsoft.com)

4.2.4 Azure Test Plans



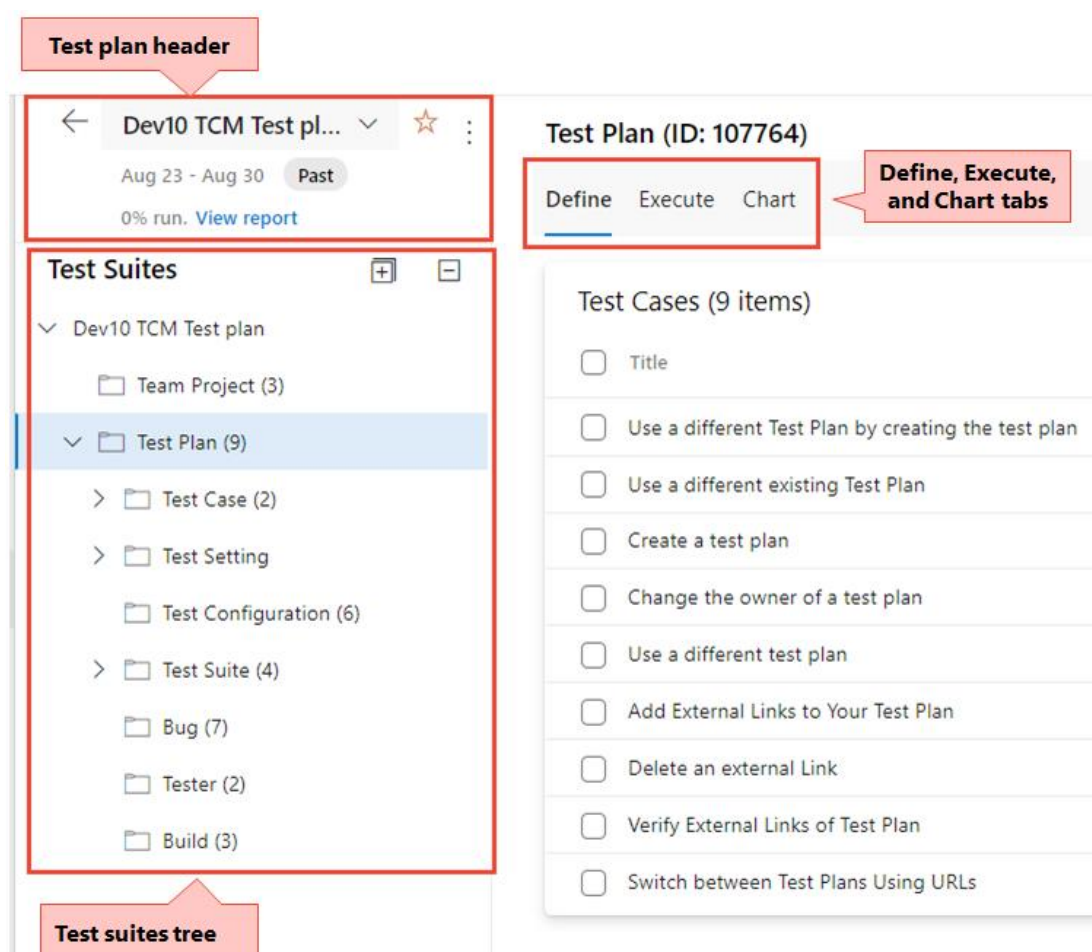
Το Azure Test Plans παρέχει ισχυρά εργαλεία που μπορούν να χρησιμοποιήσουν όλοι στην ομάδα για να αυξήσουν την ποιότητα και τη συνεργασία σε όλη τη διαδικασία ανάπτυξης. Η εύχρηστη λύση διαχείρισης δοκιμών που βασίζεται σε πρόγραμμα περιήγησης παρέχει όλες τις δυνατότητες που απαιτούνται για προγραμματισμένες μη αυτόματες δοκιμές, δοκιμές αποδοχής από τον χρήστη, διερευνητικές δοκιμές και συλλογή σχολίων από τα ενδιαφερόμενα μέρη. Μέσω ενός συνδυασμού εργαλείων που βασίζονται σε πρόγραμμα περιήγησης, σχεδίων δοκιμής, αναφοράς προόδου, παραμέτρων, διαμορφώσεων, εργαλείων δοκιμής και λειτουργιών ενσωμάτωσης DevOps, τα σχέδια δοκιμών Azure υποστηρίζουν τους ακόλουθους στόχους δοκιμής:

- **Χειροκίνητη και διερευνητική δοκιμή:** που περιλαμβάνει τις ακόλουθες δραστηριότητες δοκιμής:
 - Προγραμματισμένη χειροκίνητη δοκιμή οργανώνοντας τις δοκιμές σε σχέδια δοκιμών και σειρές δοκιμών από καθορισμένους ελεγκτές και υποψηφίους δοκιμών.
 - Δοκιμή αποδοχής χρήστη που πραγματοποιούνται από καθορισμένους ελεγκτές αποδοχής χρηστών για να επαληθευτεί

ότι η προδοθείσα τιμή πληροί τις απαιτήσεις των πελατών, ενώ χρησιμοποιούνται εκ νέου τα τεχνουργήματα δοκιμής που δημιουργήθηκαν από ομάδες μηχανικών.

- ο Διερευνητικές δοκιμές που πραγματοποιούνται από ομάδες ανάπτυξης, συμπεριλαμβανομένων προγραμματιστών, δοκιμαστών, ομάδων UX, ιδιοκτητών προϊόντων και άλλων, με την εξερεύνηση των συστημάτων λογισμικού χωρίς τη χρήση δοκιμαστικών σχεδίων ή σειρών δοκιμών.
 - ο Σχόλια από ενδιαφερόμενους φορείς. Δοκιμές που πραγματοποιούνται από ενδιαφερόμενα μέρη εκτός της ομάδας ανάπτυξης, όπως χρήστες από τμήματα μάρκετινγκ και πωλήσεων.
- **Αυτοματοποιημένες δοκιμές:** Το Azure Test Plans είναι πλήρως ενσωματωμένο με το Azure Pipelines για υποστήριξη δοκιμών στο πλαίσιο συνεχούς ενοποίησης/συνεχούς ανάπτυξης (CI/CD). Τα σχέδια δοκιμών και οι περιπτώσεις δοκιμών μπορούν να συσχετιστούν με αγωγούς κατασκευής ή απελευθέρωσης. Οι εργασίες αγωγών μπορούν να προστεθούν στους ορισμούς των αγωγών για την καταγραφή και τη δημοσίευση των αποτελεσμάτων δοκιμών. Τα αποτελέσματα των δοκιμών μπορούν να αναθεωρηθούν μέσω ενσωματωμένων αναφορών προόδου και αναφορών δοκιμών αγωγών.
- **Ιχνηλασιμότητα:** Οι δοκιμαστικές περιπτώσεις και οι σειρές δοκιμών που συνδέονται με ιστορίες, δυνατότητες ή απαιτήσεις χρηστών υποστηρίζουν την ιχνηλασιμότητα από άκρο σε άκρο. Οι δοκιμές και τα ελαττώματα συνδέονται αυτόματα με τις απαιτήσεις και τις κατασκευές που δοκιμάζονται, γεγονός που βοηθά επίσης στην παρακολούθηση της ποιότητας των απαιτήσεων. Οι χρήστες μπορούν να προσθέσουν και να εκτελέσουν δοκιμές από τον πίνακα Kanban ή για μεγαλύτερες ομάδες, να χρησιμοποιήσουν τον κόμβο σχεδίων δοκιμής για να ορίσουν σχέδια δοκιμών και σειρές δοκιμών. Τα αποτελέσματα του Pipeline και το γραφικό στοιχείο Requirements παρέχουν ένα μέσο παρακολούθησης των δοκιμών των απαιτήσεων.

Αναφορά και ανάλυση: Η παρακολούθηση αποτελεσμάτων δοκιμής και η παρακολούθηση προόδου υποστηρίζονται μέσω διαμορφώσιμων γραφημάτων παρακολούθησης, γραφικών στοιχείων για συγκεκριμένες δοκιμές που οι χρήστες μπορούν να προσθέσουν στους πίνακες εργαλείων και ενσωματωμένων αναφορών, όπως η αναφορά προόδου, οι αναφορές αποτελεσμάτων δοκιμής διοχέτευσης και η υπηρεσία Analytics. (azure.microsoft.com/en-us/services/devops/)



Σχήμα 35 – Azure Test Plans ([dos.microsoft.com](https://docs.microsoft.com))

4.2.5 Azure Artifacts

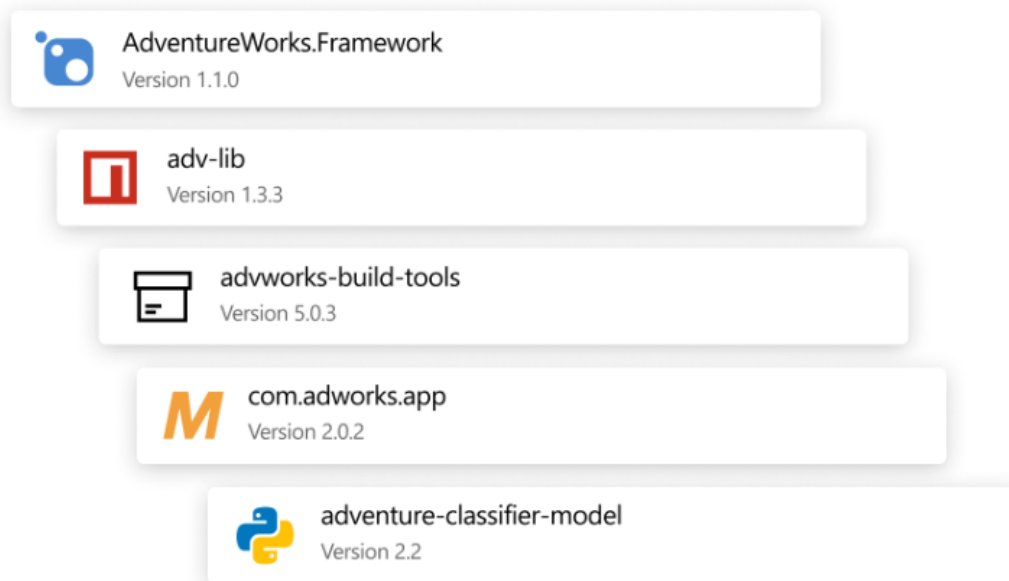


Το Azure Artifacts επιτρέπει στους προγραμματιστές να μοιράζονται και να καταναλώνουν πακέτα από διαφορετικές ροές και δημόσια μητρώα. Τα πακέτα μπορούν να κοινοποιηθούν στην ίδια ομάδα, στον ίδιο οργανισμό, ακόμη και

δημόσια. Το Azure Artifacts υποστηρίζει πολλούς τύπους πακέτων όπως πακέτα NuGet, npm, Python, Maven και Universal. Παρακάτω περιγράφονται οδηγίες και τις βέλτιστες πρακτικές κατά τη δημιουργία ή τη δημοσίευση των πακέτων.

- Κάθε αποθετήριο θα πρέπει να αναφέρεται μόνο σε μία ροή: Μια ροή είναι μια οργανωτική κατασκευή για τη φιλοξενία πακέτων. Μπορούν να υπάρχουν πολλές ροές για ένα έργο, αλλά ένα συγκεκριμένο έργο θα πρέπει να αναφέρεται μόνο σε μία ροή.
- Αυτόματη δημοσίευση πακέτων που δημιουργήθηκαν πρόσφατα στη ροή.
- Ενεργοποίηση της πολιτικής διατήρησης για να καθαριστούν αυτόματα οι παλιές εκδόσεις πακέτων: Η διαγραφή παλαιών εκδόσεων πακέτων βελτιώνει την απόδοση του πελάτη και απελευθερώνει χώρο αποθήκευσης. Μπορεί να γίνει επιλογή πόσες εκδόσεις ενός πακέτου θα διατηρηθούν κατά τη ρύθμιση της πολιτικής διατήρησης.

Προώθηση του πακέτου στη σωστή προβολή: Όταν ένα πακέτο είναι έτοιμο για πρώιμους χρήστες, είτε κριθεί ότι έχει επαρκή ποιότητα για να κυκλοφορήσει. Η προώθηση εκδόσεων πακέτων σε μια προβολή διασφαλίζει ότι δεν θα διαγραφούν από τις πολιτικές διατήρησης.
(azure.microsoft.com/en-us/services/devops/)



Σχήμα 36- Azure artifacts (dos.microsoft.com)

5

Διαχείριση έργου για το Ευρωπαϊκό Κέντρο Πρόληψης και Ελέγχου Νόσων (ECDC) μέσω Azure DevOps



Το Ευρωπαϊκό Κέντρο Πρόληψης και Ελέγχου Νόσων (ECDC) είναι μια ανεξάρτητη υπηρεσία της Ευρωπαϊκής Ένωσης (ΕΕ), ιδρύθηκε το 2005 και βρίσκεται στην πόλη Solna της Σουηδίας. Η αποστολή του ECDC είναι να εντοπίζει, να αξιολογεί και να κοινοποιεί τις τρέχουσες και τις αναδυόμενες απειλές για την ανθρώπινη υγεία που προκαλούνται από μολυσματικές ασθένειες. Για την επίτευξη αυτής της αποστολής, το ECDC συνεργάζεται με εθνικούς φορείς προστασίας της υγείας σε όλη την Ευρώπη για την ενίσχυση και την ανάπτυξη συστημάτων επιτήρησης ασθενειών και έγκαιρης προειδοποίησης σε ολόκληρη την ήπειρο. Συνεργαζόμενος με ειδικούς σε όλη την Ευρώπη, το ECDC συγκεντρώνει τις γνώσεις σχετικές με την υγεία για να αναπτύξει έγκυρες επιστημονικές απόψεις σχετικά με τους κινδύνους που ενέχουν οι τρέχουσες και οι αναδυόμενες μολυσματικές ασθένειες. Στο πλαίσιο της αποστολής του, το Κέντρο:

- αναζητά, συλλέγει, ταξινομεί, αξιολογεί και διαδίδει σχετικά επιστημονικά δεδομένα

- παρέχει επιστημονικές γνώμες και τεχνική βοήθεια, συμπεριλαμβανομένης της κατάρτισης
- παρέχει έγκαιρες πληροφορίες στην Επιτροπή, τα κράτη μέλη, τους κοινοτικούς φορείς και τους διεθνείς οργανισμούς που δραστηριοποιούνται στον τομέα της δημόσιας υγείας
- συντονίζει την ευρωπαϊκή δικτύωση των φορέων που δραστηριοποιούνται στους τομείς της αποστολής του Κέντρου, συμπεριλαμβανομένων των δικτύων που προκύπτουν από δραστηριότητες δημόσιας υγείας που υποστηρίζονται από την Επιτροπή και λειτουργούν τα ειδικά δίκτυα επιτήρησης
- ανταλλάσσει πληροφορίες, εμπειρογνωμοσύνη και βέλτιστες πρακτικές.

Συγκεκριμένα, το ECDC συλλέγει, αναλύει και διανέμει δεδομένα επιτήρησης για 56 μεταδοτικές ασθένειες και σχετικά ειδικά θέματα υγείας από τα 27 κράτη μέλη της Ευρωπαϊκής Ένωσης (ΕΥ) και δύο από τις τρεις υπόλοιπες χώρες του Ευρωπαϊκού Οικονομικού Χώρου (ΕΕΑ) (Ισλανδία και Νορβηγία). Η υποβολή δεδομένων και η επακόλουθη επικύρωση είναι ευθύνη των ευρωπαϊκών δικτύων εμπειρογνομόνων σε ασθένειες που ορίζονται από τα κράτη μέλη και συντονίζονται από το ECDC μέσω των προγραμμάτων ασθενειών του και διευκολύνει την ανάπτυξη και εφαρμογή κοινών δράσεων. Τα δεδομένα επιτήρησης που συλλέγονται σε ευρωπαϊκό επίπεδο βασίζονται κατά κύριο λόγο σε περιπτώσεις και περιλαμβάνουν δημογραφικές, κλινικές, επιδημιολογικές και εργαστηριακές πληροφορίες. Αναφέρονται ετησίως (για τις περισσότερες ασθένειες και ειδικά θέματα υγείας), μηνιαία (για σαλμονέλα, ιλαρά και ερυθρά), εβδομαδιαία (για γρίπη και πυρετό του Δυτικού Νείλου) ή καθημερινά (για τη νόσο των Λεγεωνάριων που σχετίζεται με ταξίδια), όπως απαιτείται για συγκεκριμένες στόχους, αποτελέσματα και συνακόλουθες δράσεις δημόσιας υγείας. Το ECDC και τα ευρωπαϊκά δίκτυα ασθενειών διασφαλίζουν τυποποιημένη αναφορά και σύγκριση δεδομένων σε ολόκληρη την ΕΥ μέσω της χρήσης κοινών (εξωτερικά εξασφαλισμένης ποιότητας) μεθόδων διάγνωσης και τυποποίησης, ορισμών περιπτώσεων, metadata και πρωτοκόλλων αναφοράς. Το ECDC

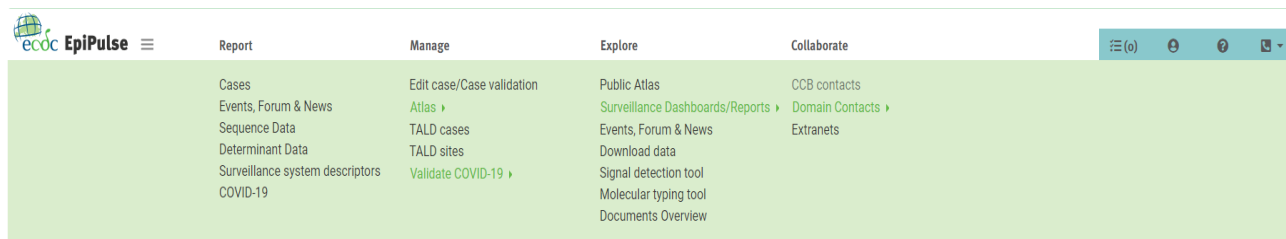
και τα κράτη μέλη προσπαθούν επίσης για τακτική παρακολούθηση της ποιότητας των δεδομένων και αξιολόγηση των συστημάτων επιτήρησης. Η τεχνική πλατφόρμα για την υποβολή δεδομένων, την αποθήκευση και τη διάδοση δεδομένων μέσω Web είναι το Ευρωπαϊκό Σύστημα Επιτήρησης (TESSy) μια βάση δεδομένων που φιλοξενείται από το ECDC.

(www.ecdc.europa.eu)

5.1 EpiPulse Platform

Το EpiPulse κυκλοφόρησε στις 22 Ιουνίου 2021. Ενσωματώνει πολλά συστήματα επιτήρησης που ήταν προηγουμένως ανεξάρτητα (Το Ευρωπαϊκό Σύστημα Επιτήρησης (TESSy), οι πέντε πλατφόρμες Epidemic Intelligence Information System (EPIS) και το Threat Tracking Tool (TTT)), παρέχει νέες λειτουργίες και απρόσκοπτη πρόσβαση σε δεδομένα σε μια ενιαία πλατφόρμα. Η πύλη διευκολύνει τη συλλογή, την ανάλυση και τη διάδοση δεδομένων επιτήρησης βάσει δεικτών και συμβάντων για μολυσματικές ασθένειες και συναφή ζητήματα υγείας, συμπεριλαμβανομένης της παγκόσμιας επιδημικής νοημοσύνης, της αλληλουχίας ολόκληρου του γονιδιώματος και των καθοριστικών παραγόντων υγείας. Διορισμένοι εμπειρογνώμονες από την EU/EEA και χώρες εκτός EU στο πλαίσιο συνεργασίας του ECDC, το προσωπικό του ECDC και εκπρόσωποι ευρωπαϊκών αρχών και διεθνών οργανισμών μπορούν να έχουν πρόσβαση στην πύλη για να αναφέρουν και να αναλύουν περιπτώσεις μολυσματικών ασθενειών και παθογόνων παραγόντων που ενδέχεται να απειλήσουν τη δημόσια υγεία στην EU/EEA. Η πλατφόρμα διευκολύνει τη διεπιστημονική συνεργασία και συνδέει χρήστες από διαφορετικούς τομείς στο πλαίσιο μιας προσέγγισης One-Health. Μέσω αυτής της πλατφόρμας, το ECDC στοχεύει να ενισχύσει την πρόληψη και τον έλεγχο των μολυσματικών ασθενειών ενισχύοντας την έγκαιρη ανίχνευση και αξιολόγηση απειλών. Θα επιτρέψει την καλύτερη ετοιμότητα και διαχείριση των απειλών από μολυσματικές ασθένειες σε επίπεδο EU και σε παγκόσμιο επίπεδο, μέσω της παρακολούθησης σε πραγματικό χρόνο των σημάτων και των γεγονότων επιδημίας.

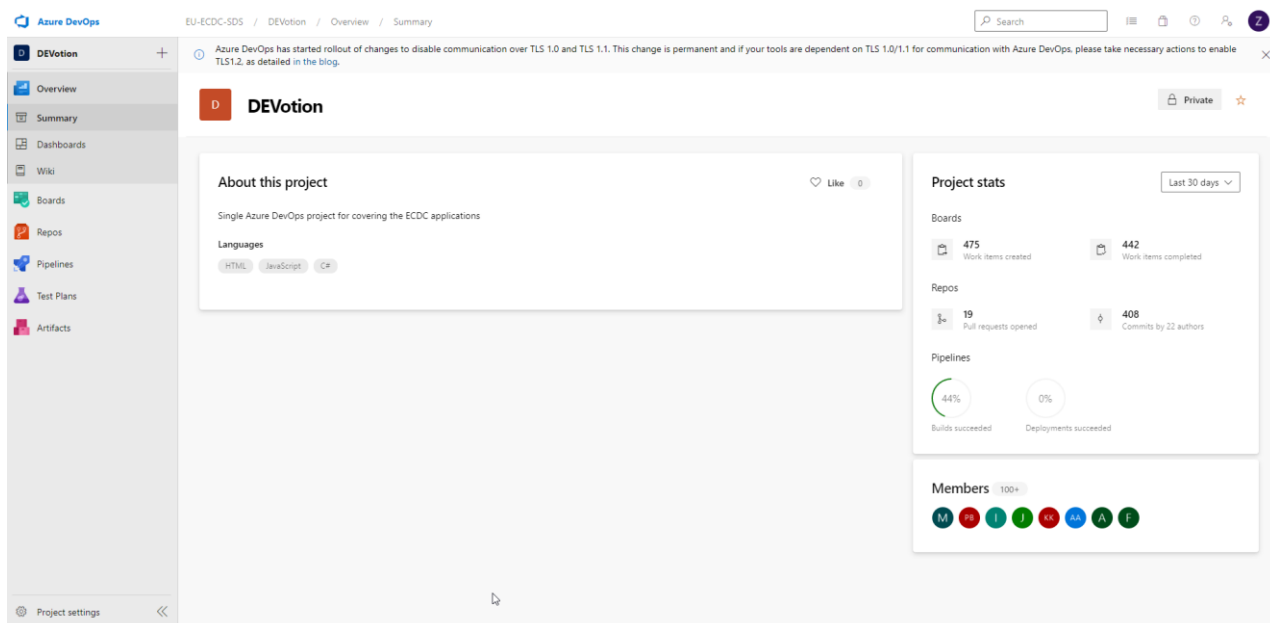
(www.ecdc.europa.eu)



Σχήμα 37- EpiPulse platform (from the system)

5.2 Διαχείριση του έργου στο Azure DevOps

Ολοκληρώνοντας το πρώτο βήμα της σύνδεσης στο Azure DevOps, ο χρήστης μπορεί να πλοηγηθεί στο βασικό μενού. Στο overview παρουσιάζονται οι βασικές πληροφορίες που αφορούν το έργο, συνοψίζοντας το μέγεθος των υπηρεσιών που έχουν λάβει χώρα σε επιλέξιμο χρονικό διάστημα. Επίσης, οι χρήστες έχουν την δυνατότητα να δημιουργήσουν dashboards που περιλαμβάνουν δραστηριότητες καθώς και το στάδιο ολοκλήρωσής τους. Ενδεικτικά, για το EpiPulse Release C όπου αφορά τις απαιτήσεις που θα υλοποιηθούν και θα παραδοθούν στον οργανισμό το πρώτο εξάμηνο του 2022 δημιουργήθηκαν pie charts και bar charts όπου απεικονίζουν τα τρέχοντα sprints, τα requirements και τα τεστ σενάρια που έχουν ήδη δημιουργηθεί.



Σχήμα 38- Azure DevOps Summary (from the system)

Διοίκηση έργων μέσω σύγχρονων μεθόδων: Μελέτη περίπτωσης το εργαλείο Microsoft Azure DevOps Χρυσούλα Ζάχαρη



Σχήμα 39- Azure DevOps dashboard (from the system)


Εν συνεχεία, στα Boards που αποτελούν και το μεγαλύτερο μέρος εργασιών των χρηστών της UniSystems αλλά και του ECDC, περιλαμβάνονται τα work items σε λίστα όπου υπάρχει δυνατότητα να γίνει αναζήτηση, φιλτράρισμα και πλοήγηση εντός του καθένα ξεχωριστά.

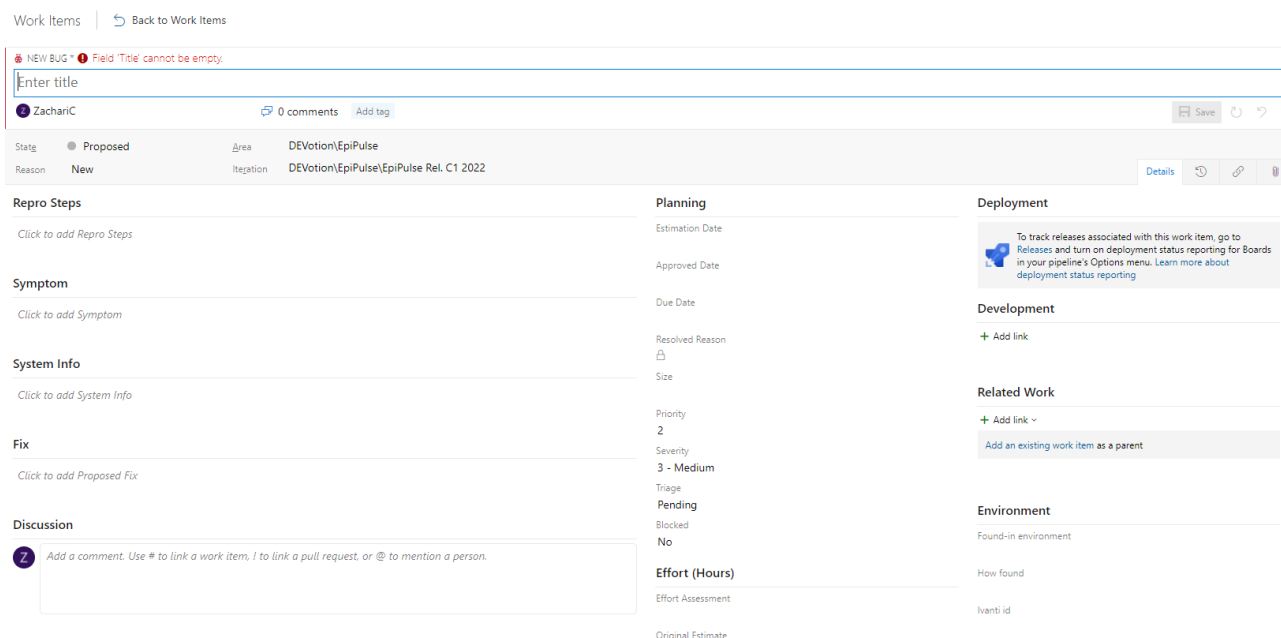
The screenshot displays the Azure DevOps Boards / Work Items view. The table below represents the data shown in the screenshot:

ID	Title	Assigned To	State	Area Path	Tags	Comments	Activity Date
20248	D2.5 Technical Documentation (TD) - EpiPulse TALD Release B 202...		Review	DEVotion(EpiPulse	Further Development	5	5/4/2022 6:36:52 PM
23156	Adaptive Maintenance: Upgrade .Net CORE: Molecular typing Tool ...		Resolved	DEVotion(EpiPulse	Adaptive Maintenance		5/4/2022 12:01:54 PM
23152	Adaptive Maintenance: Upgrade .Net CORE: EpiPulse Portal v2.2 ->...		Resolved	DEVotion(EpiPulse	Adaptive Maintenance		5/4/2022 12:01:35 PM
23414	SREQ.21.12.05.01 Output Details: Output history log		Proposed	DEVotion(EpiPulse	CDTR Output History	1	5/4/2022 10:28:59 AM
22727	TALD migration tool run 0104 - log error		Proposed	DEVotion(EpiPulse	migration tool	7	5/3/2022 2:26:47 PM
23154	Adaptive Maintenance: Upgrade .Net CORE: EpiPulse Events and D...		Active	DEVotion(EpiPulse	Adaptive Maintenance		5/3/2022 1:41:15 PM
23373	DB data migration		Proposed	DEVotion(EpiPulse			4/29/2022 2:56:45 PM
23290	TALD TESSy-EpiPulse integration - Multiple Cases transfer - Interna...		Proposed	DEVotion(EpiPulse	UAT finding	2	4/29/2022 2:45:52 PM
21307	EpiPulse: ETMS: "Send Notification" dialogue box is missing from It...		Proposed	DEVotion(EpiPulse	existing remain Sprint 4	1	4/28/2022 11:52:35 AM
21136	EpiPulse: System: UI Details		Active	DEVotion(EpiPulse	Pause REOPENED Sprint 3	11	4/28/2022 11:52:35 AM
20700	EpiPulse: There is NO Internal (ECDC) user		Proposed	DEVotion(EpiPulse	remain Sprint 2 Sprint 3	5	4/28/2022 11:52:35 AM
22050	Editing comment with attachment removes the attachment (bug)		Active	DEVotion(EpiPulse		5	4/28/2022 10:47:04 AM
23018	D2.1 Project Management Plan (PmP) - EpiPulse Release C1 2022		Review	DEVotion(EpiPulse	Further Development		4/27/2022 9:12:42 PM
18671	D2.5 Technical Documentation (TD) - EpiPulse Rel.B 2021 Acceptan...		Review	DEVotion(EpiPulse	Further Development	2	4/27/2022 8:47:58 PM
19664	SREQ.21.06.01.08 System: Digest notification template		Proposed	DEVotion(EpiPulse	Further Development	1	4/21/2022 5:53:59 PM
19000	TALD item management		Proposed	DEVotion(EpiPulse	MustForGOLive TALD	1	4/20/2022 2:22:48 PM


Σχήμα 40- Work Items (from the system)

Συγκεκριμένα, τα work items που χρησιμοποιούνται από τις ομάδες κατηγοριοποιούνται στις ακόλουθες κατηγορίες:

-  **Bugs**, χρησιμοποιούνται κυρίως από τις ομάδες ανάπτυξης και testing και περιγράφουν αναλυτικά τα στάδια αναπαραγωγής του προβλήματος, τα συμπτώματα που παρουσιάζονται και τον τρόπο επίλυσης του. Δίνεται δυνατότητα σχεδιασμού βάσει κρισιμότητας, εκτίμησης χρόνου διόρθωσης, ταξινόμησης βάσει της φύσης του και αλλαγή statuses μέχρι την ολοκλήρωσή του. Επίσης, οι χρήστες του ECDC χρησιμοποιούν το λογισμικό Ivanti για την καταγραφή των σφαλμάτων και ο κωδικός καταγράφεται και στο DevOps για να υπάρχει συνέπεια.



Σχήμα 41- Insert a bug (from the system)

-  **Test Cases**, χρησιμοποιούνται από την ομάδα testing κατά την διαδικασία ελέγχου της εφαρμογής πριν παραδοθεί στον ECDC. Η test ομάδα χρησιμοποιεί τα σενάρια που έχει φτιάξει η ομάδα των Business Analysts, αναπαραγάγει τα βήματα και καταγράφει τα αποτελέσματα. Εφόσον τα αποτελέσματα είναι τα προσδοκώμενα το test case ολοκληρώνεται αλλάζοντας status σε completed. Σε περίπτωση

Διοίκηση έργων μέσω σύγχρονων μεθόδων: Μελέτη περίπτωσης το εργαλείο Microsoft Azure DevOps

Χρυσούλα Ζάχαρη

σφαλμάτων, δημιουργείται το αντίστοιχο bug όπου συνδέεται με το test case με related connection.

The screenshot shows a test case in the Azure DevOps system. The test case is titled "Molecular Typing Tool Field Type" and is in a "Ready" state. It is associated with the area "DEV0101/EpPulse" and the iteration "DEV0101/EpPulse/EpPulse Rel. C1 2022". The test case has 10 steps, each with an action, expected result, and attachments. The steps are:


1. Login to Molecular typing tool
2. Verify the existence of field 'Type' before the Cluster field
3. Try selecting more at field 'Type'
4. Try selecting at the field 'Type'
5. Try selecting at the field 'Type' and check the Country drop down list
6. Verify the 'Type' label.
7. The 'Type' field is empty verify the message appearing in box and the message at hover over
8. Select a value and hover over
9. Try searching for a value
10. Click on 'Search'

The expected results for these steps are:

- The 'Type' field is located before Cluster field
- Its single select field
- The type filter does not affect any other field
- The Type input should dynamically define the list of clusters in the Clusters field. Only the related clusters will be displayed in the drop-down menu, not preselected
- It should not affect any other field
- The countries that appear are related to the Cluster
- The 'Type' label is dynamically defined from the API pathogen
- For both cases the message is as follows: "Please, select one «label»" in-box
- The hover shows the selected value
- The system defines real time the multi selection lists
- The system loads the relevant data

The test case also has a "Related Work" section on the right, which includes a "Shared Steps" section and a "Status" section.

Σχήμα 42- Test case example (from the system)

-  **Epics**, δημιουργούνται από τους χρήστες του ECDC και περιγράφονται αναλυτικά στο 'Software Requirements Specification' (SRS) έγγραφο που δίνεται από τον οργανισμό. Είναι το πρώτο στάδιο του έργου όπου ο ECDC περιγράφει σε high level την απαίτηση προς υλοποίηση. Στο Azure DevOps καταγράφονται σύμφωνα με το SRS έγγραφο και με related connection συνδέονται τα requirements και τα use cases.

The screenshot shows an Epic in the Azure DevOps system. The Epic is titled "UC.21.14 Output visual comparison of public/restricted versions" and is in an "Active" state. It is associated with the area "DEV0101/EpPulse" and the iteration "DEV0101/EpPulse/EpPulse Rel. C1 2022". The Epic has a description, actors, preconditions, scenarios, and a basic path. The description is:

Output visual comparison of public/restricted versions

The user wants to see the differences between restricted and public version in "Edit CDR pre-production" page using as a reference the public version (no version is considered as reference (similarly to the functioning of track changes at mark-up in word) -see mock-up below)

The actors are:

- ECDC users

The preconditions are:

- The users have selected a CDR output
- The user have clicked the "Edit CDR pre-production" button
- The user have landed in CDR pre-production page

The scenarios are:


1. The user clicks in "Show differences" switch on/off

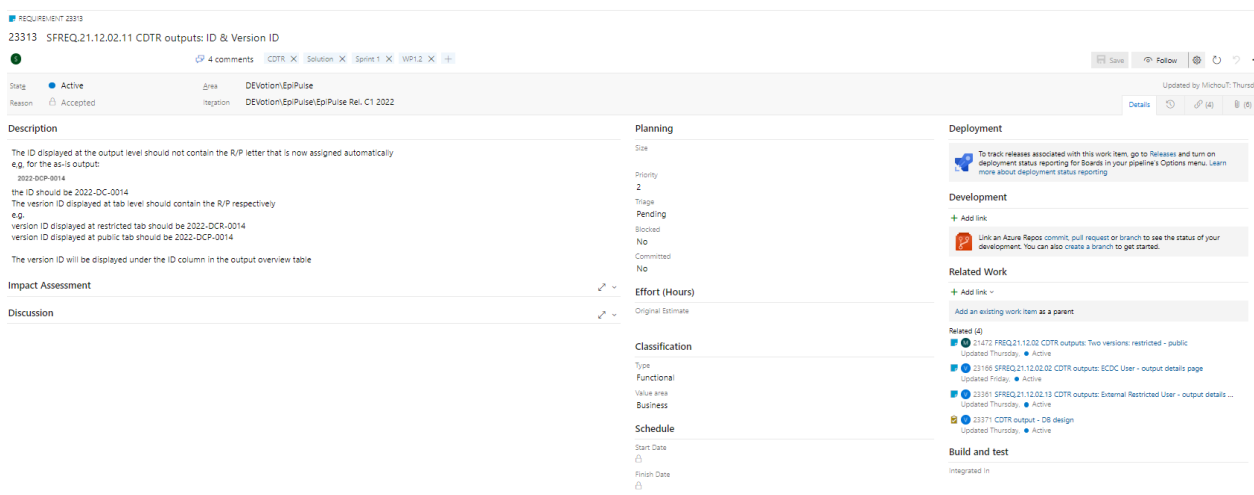
The basic path is:

1. The system displays the differences between restricted and


The Epic also has a "Planning" section on the right, which includes a "Priority" section and a "Classification" section.

Σχήμα 43- Epic example (from the system)

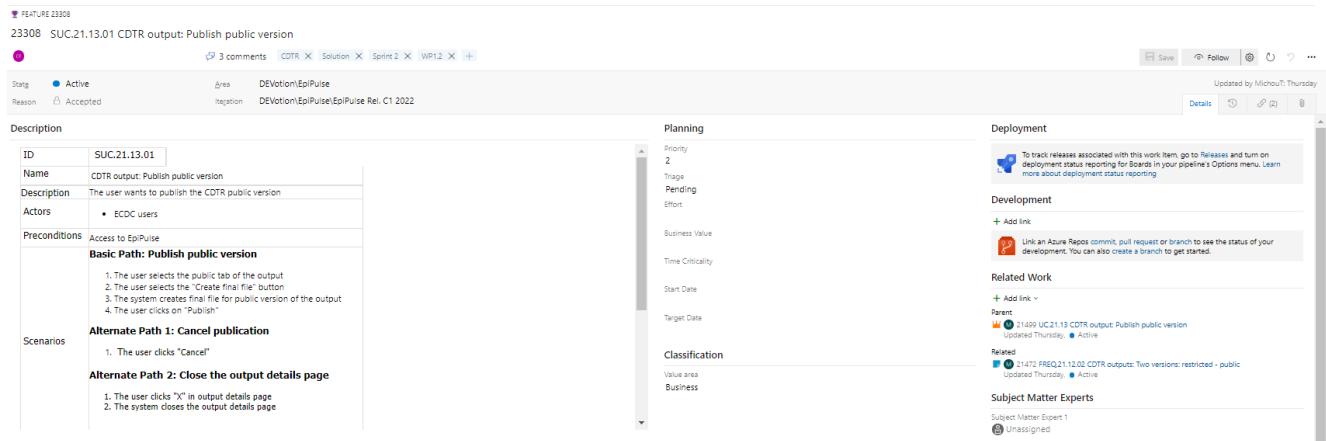
-  **Requirements**, αποτελούν το μεγαλύτερο μέρος του έργου. Κάθε epic ενδέχεται να συνδέεται με περισσότερα από δέκα requirements. Τα requirements με την σειρά τους κατηγοριοποιούνται σε ομάδες ανάλογα με την φύση τους, δηλαδή σε functional, non-functional και solution requirements. Η ομάδα των Business Analysts είναι υπεύθυνη να τα καταγράψει στην περίπτωση που είναι functional ή non-functional. Ενώ στην περίπτωση των solution δημιουργούνται έπειτα από συζήτηση και ανάλυση με τους stakeholders. Τα solution requirements εφόσον είναι validated από τους stakeholders γίνονται assigned στην ομάδα ανάπτυξης, όπου βάσει αυτών θα υλοποιηθούν. Σε αυτό το στάδιο, τα solutions requirements υλοποιούνται σε sprints, συνήθως των 2-3 εβδομάδων.




Σχήμα 44- Requirement example (from the system)

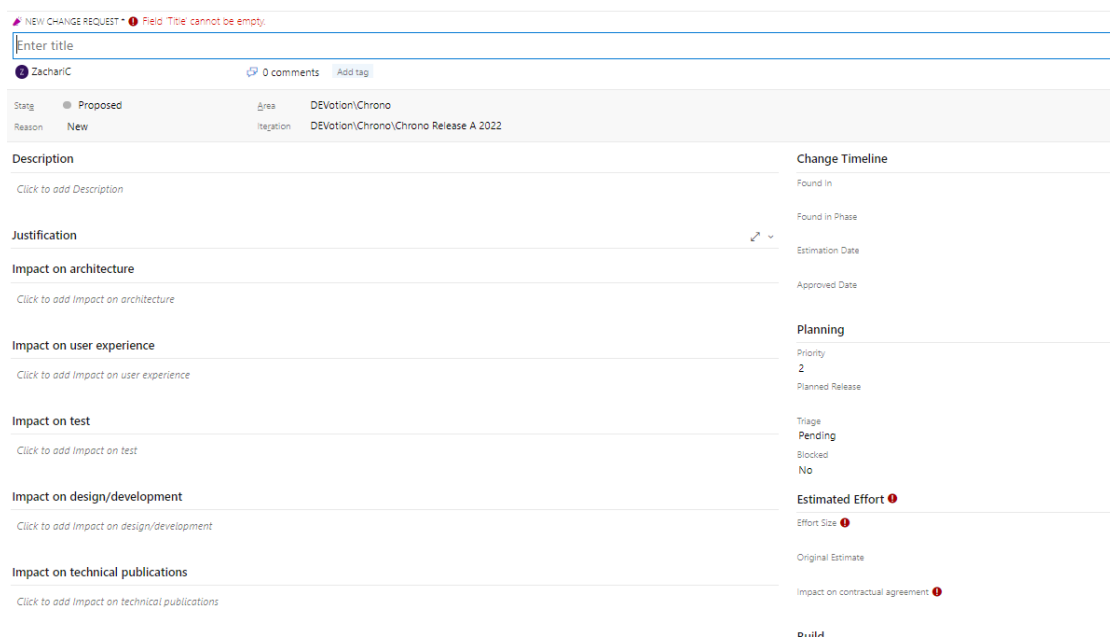
-  **Features**, δημιουργούνται από την ομάδα των αναλυτών και αφορούν τα solution use cases βάσει των solution requirements. Τα εγκρίνουν οι stakeholders του ECDC και μετά την υλοποίηση, η ομάδα testing θα τα χρησιμοποιήσει ως σενάρια αναπαραγωγής.

Διοίκηση έργων μέσω σύγχρονων μεθόδων: Μελέτη περίπτωσης το εργαλείο
Microsoft Azure DevOps
Χρυσούλα Ζάχαρη




Σχήμα 45- Feature example (from the system)

-  **Change Requests**, αφορούν αλλαγές που ζητούνται από τον ECDC κατά την διαδικασία του UAT (User Acceptance Testing) είτε εφόσον έχει παραδοθεί το έργο και κατά την χρήση από τους χρήστες ζητηθεί να γίνει κάποια αλλαγή που δεν είχε καταγραφεί στις αρχικές απαιτήσεις.



Σχήμα 46- Change request example (from the system)


-  **Tasks**, δημιουργούνται από την ομάδα ανάπτυξης κατά την υλοποίηση των solution requirements και ουσιαστικά διαχωρίζουν την προδιαγραφή σε ξεχωριστά τμήματα που πρέπει το καθένα να ολοκληρωθεί.

Διοίκηση έργων μέσω σύγχρονων μεθόδων: Μελέτη περίπτωσης το εργαλείο Microsoft Azure DevOps

Χρυσούλα Ζάχαρη

The screenshot shows a work item in the 'Proposed' state. The description field is empty, and the discussion section has a placeholder for adding comments. The right-hand pane contains sections for Planning (Priority 2, No tags, No blocked items, No effort), Deployment (To track releases associated with this work item, go to Releases and turn on deployment status reporting for Boards in your pipeline's Options menu), Development (Add link), Related Work (Add link), and Build (Integrated in).

Σχήμα 47- Insert a task (from the system)

-  **Release Plan**, καταχωρείται από τον Project Manager της UniSystems και αφορά το release package των εργασιών που έχουν συμφωνηθεί να παραδοθούν στις αναγραφόμενες ημερομηνίες περιλαμβάνοντας όλα τα Items που θα παραδοθούν στο ECDC.

The screenshot shows a work item in the 'Active' state. The description field contains a table with the following content:

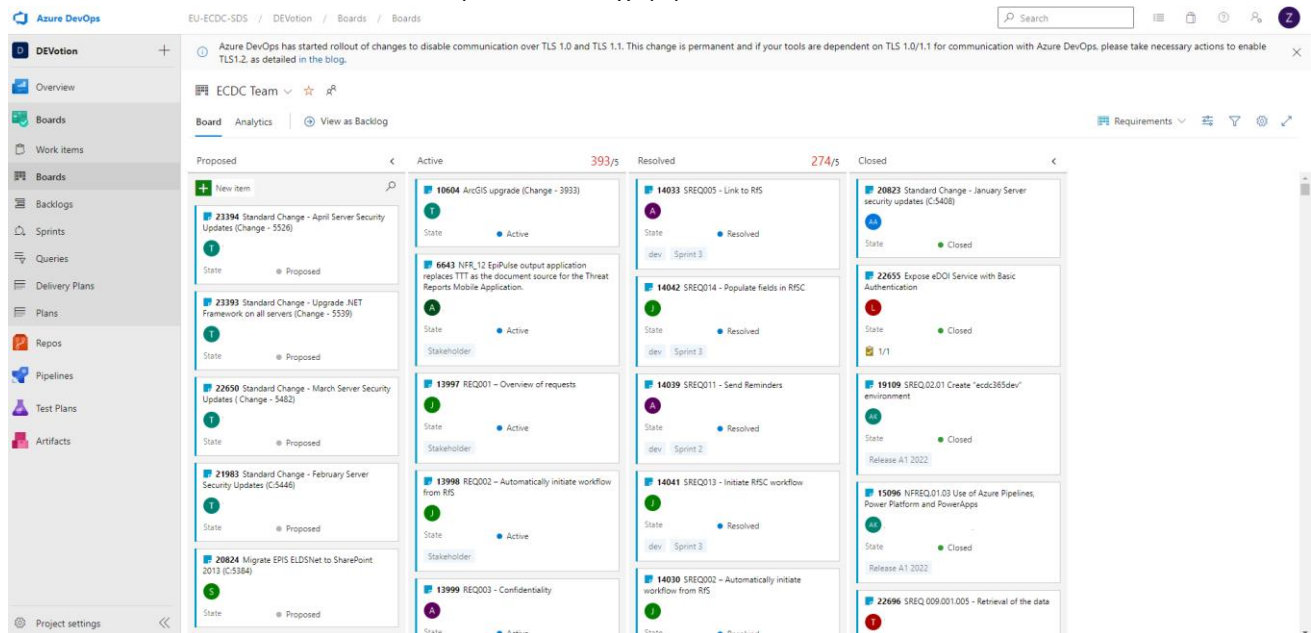
FOR SPECIFIC CONTRACT No SC9 ECD.12964, ECD.12965 ID 14544 UNDER FRAMEWORK CONTRACT No ECDC 2018 033, and SUB-TASK 573 EpiPulse				
Original document - duly signed - to be attached to the invoice				
REQUEST AND DESCRIPTION OF TASK/ WORK				
TITLE: EpiPulse Release C1 2022				
DATE OF REQUEST: 22/03/2022		LAST DATE FOR OFFER: 07/04/2022		
QTM START DATE: 08/04/2022		QTM END DATE: 24/10/2022		
RESPONSIBLE PERSON AT ECDC (QTM MANAGER)				
DESCRIPTION OF SUB-TASK(S) AND EXPECTED DELIVERABLE(S):				
Subtask parameters				
Objectives:				
Deliver further development tasks				
Deliver Adaptive Maintenance Tasks				
Scopes:				
The following items are included in the scope of the QTM:				
Scope of further development				
Work package 1.1 - Molecular Typing (4 items)				
Work items that are related to WGS and each ECDC (validation by the date: 12/04/2022).				
ID	Release	Module	Business Area	Title
20451	C 2022 WP1	Molecular Typing	WGS	UC 21.09 Molecular Typing Tool: User's session handling
20513	C 2022 WP1	Molecular Typing	WGS	FREQ.21.09.01 Molecular Typing: Active session management

Σχήμα 48- Release plan example (from the system)

Συνεχίζοντας, στο Azure DevOps περιλαμβάνεται το Kanban όπου παρουσιάζονται οπτικοποιημένα τα items (Bugs, Requirements) βάσει του status που βρίσκονται την δεδομένη στιγμή.

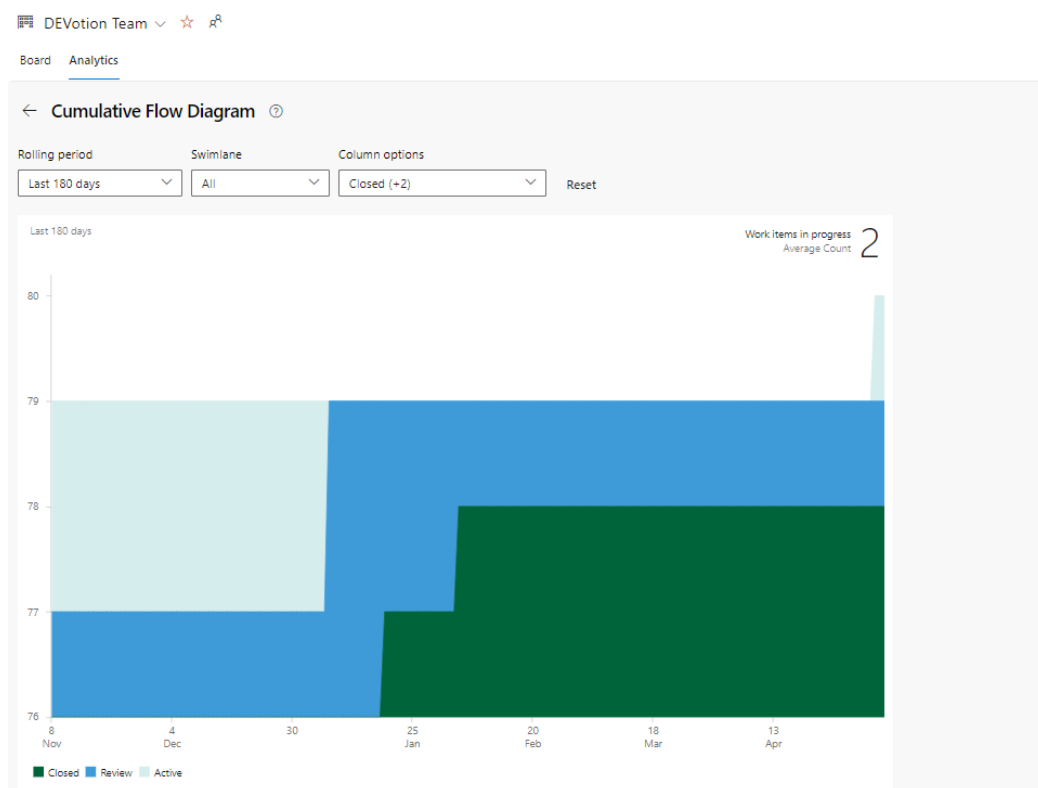
Διοίκηση έργων μέσω σύγχρονων μεθόδων: Μελέτη περίπτωσης το εργαλείο Microsoft Azure DevOps

Χρυσούλα Ζάχαρη



Σχήμα 49- Azure Kanban (from the system)

Επίσης, τα δεδομένα μπορούν να αναπαρασταθούν σε αθροιστικό διάγραμμα ροής, επιλέγοντας τα κριτήρια που ενδιαφέρουν τον χρήστη.

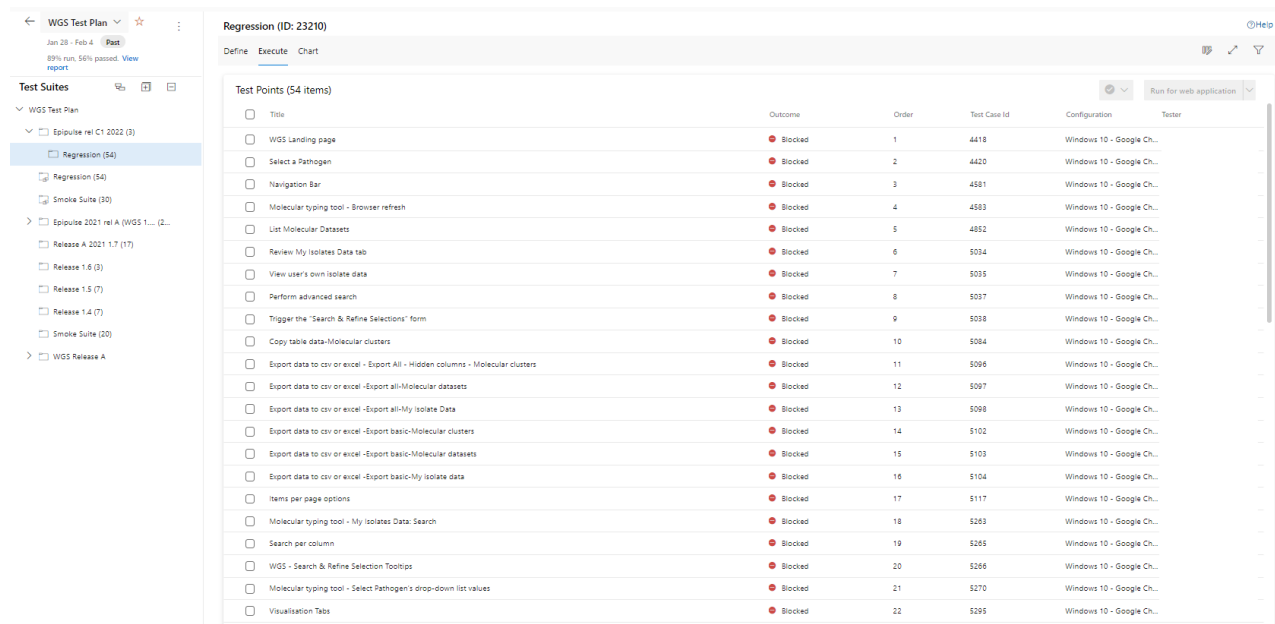


Σχήμα 50- Αθροιστικό διάγραμμα ροής (from the system)

Διοίκηση έργων μέσω σύγχρονων μεθόδων: Μελέτη περίπτωσης το εργαλείο Microsoft Azure DevOps

Χρυσούλα Ζάχαρη

Το Azure DevOps περιλαμβάνει ως ξεχωριστή υπηρεσία το test planning, καθώς αποτελεί και σημαντικό μέρος ενός έργου. Στην UniSystems οι testers δημιουργούν test plans όπου ορίζουν την σειρά των δοκιμών που θα πραγματοποιήσουν.

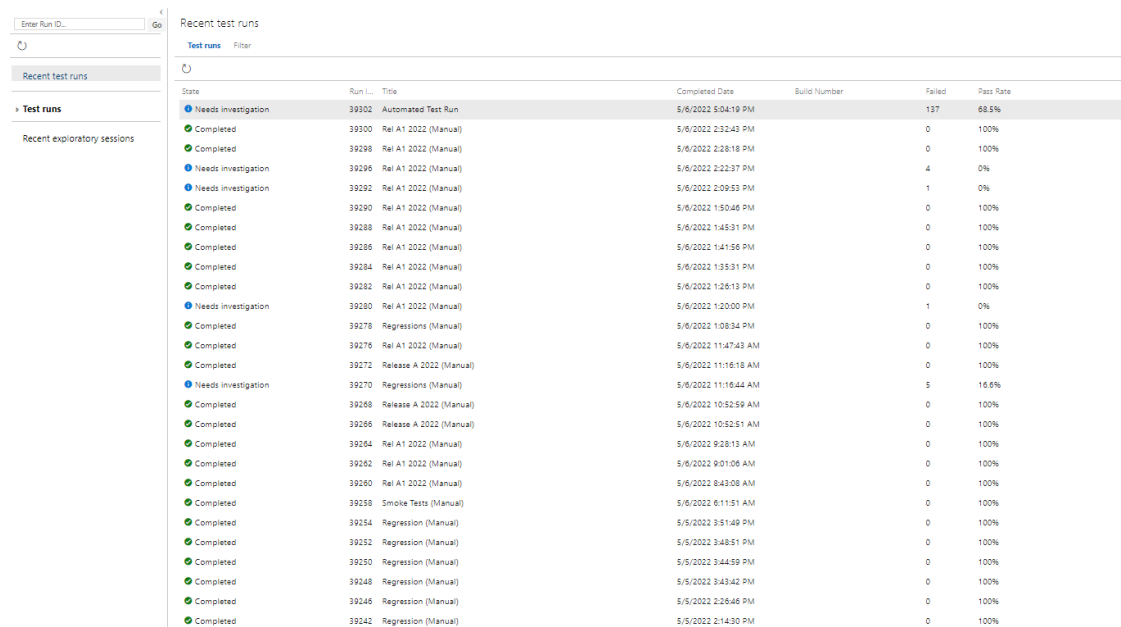


The screenshot displays the 'Regression (ID: 23210)' test suite in the Azure DevOps Test Planning tool. The interface includes a sidebar with 'Test Suites' and a main area showing a list of 54 test points. All test points are marked as 'Blocked' with a red dot icon. The table columns are: Title, Outcome, Order, Test Case Id, Configuration, and Tester.

Title	Outcome	Order	Test Case Id	Configuration	Tester
WGS Landing page	Blocked	1	4418	Windows 10 - Google Ch...	
Select a Pathogen	Blocked	2	4420	Windows 10 - Google Ch...	
Navigation Bar	Blocked	3	4581	Windows 10 - Google Ch...	
Molecular typing tool - Browser refresh	Blocked	4	4583	Windows 10 - Google Ch...	
List Molecular Datasets	Blocked	5	4852	Windows 10 - Google Ch...	
Review My Isolates Data tab	Blocked	6	5034	Windows 10 - Google Ch...	
View user's own isolate data	Blocked	7	5035	Windows 10 - Google Ch...	
Perform advanced search	Blocked	8	5037	Windows 10 - Google Ch...	
Trigger the 'Search & Refine Selections' form	Blocked	9	5038	Windows 10 - Google Ch...	
Copy table data-Molecular clusters	Blocked	10	5084	Windows 10 - Google Ch...	
Export data to csv or excel - Export All - Hidden columns - Molecular clusters	Blocked	11	5096	Windows 10 - Google Ch...	
Export data to csv or excel - Export all-Molecular datasets	Blocked	12	5097	Windows 10 - Google Ch...	
Export data to csv or excel - Export all-My isolate Data	Blocked	13	5098	Windows 10 - Google Ch...	
Export data to csv or excel -Export basic-Molecular clusters	Blocked	14	5102	Windows 10 - Google Ch...	
Export data to csv or excel -Export basic-Molecular datasets	Blocked	15	5103	Windows 10 - Google Ch...	
Export data to csv or excel -Export basic-My isolate data	Blocked	16	5104	Windows 10 - Google Ch...	
Items per page options	Blocked	17	5117	Windows 10 - Google Ch...	
Molecular typing tool - My Isolates Data: Search	Blocked	18	5263	Windows 10 - Google Ch...	
Search per column	Blocked	19	5265	Windows 10 - Google Ch...	
WGS - Search & Refine Selection Tooltips	Blocked	20	5266	Windows 10 - Google Ch...	
Molecular typing tool - Select Pathogen's drop-down list values	Blocked	21	5270	Windows 10 - Google Ch...	
Visualisation Tabs	Blocked	22	5295	Windows 10 - Google Ch...	

Σχήμα 51- Test planning (from the system)

Επίσης για κάθε test run που εκτελείται υπάρχουν dashboard που οπτικοποιούν τα αποτελέσματα και βοηθούν την ομάδα να είναι ευθυγραμμισμένη με τις απαιτήσεις που παραμένουν να ολοκληρωθούν.

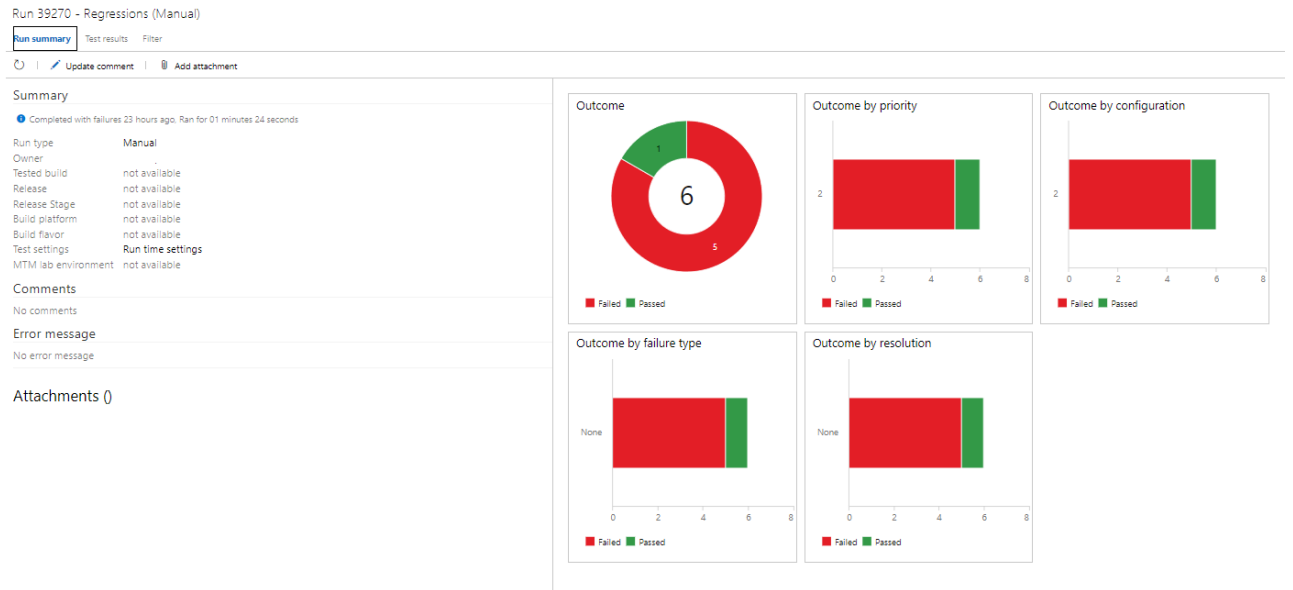


The screenshot shows the 'Recent test runs' dashboard in Azure DevOps. It includes a search bar at the top and a table listing recent test runs. The table columns are: State, Run ID, Title, Completed Date, Build Number, Failed, and Pass Rate.

State	Run ID	Title	Completed Date	Build Number	Failed	Pass Rate
Needs investigation	39302	Automated Test Run	5/6/2022 9:04:19 PM		137	68.5%
Completed	39300	Rel A1 2022 (Manual)	5/6/2022 2:32:43 PM		0	100%
Completed	39298	Rel A1 2022 (Manual)	5/6/2022 2:28:18 PM		0	100%
Needs investigation	39296	Rel A1 2022 (Manual)	5/6/2022 2:22:37 PM		4	0%
Needs investigation	39292	Rel A1 2022 (Manual)	5/6/2022 2:09:53 PM		1	0%
Completed	39290	Rel A1 2022 (Manual)	5/6/2022 1:50:46 PM		0	100%
Completed	39288	Rel A1 2022 (Manual)	5/6/2022 1:45:31 PM		0	100%
Completed	39286	Rel A1 2022 (Manual)	5/6/2022 1:41:56 PM		0	100%
Completed	39284	Rel A1 2022 (Manual)	5/6/2022 1:35:31 PM		0	100%
Completed	39282	Rel A1 2022 (Manual)	5/6/2022 1:26:13 PM		0	100%
Needs investigation	39280	Rel A1 2022 (Manual)	5/6/2022 1:20:00 PM		1	0%
Completed	39278	Regressions (Manual)	5/6/2022 1:08:34 PM		0	100%
Completed	39276	Rel A1 2022 (Manual)	5/6/2022 11:47:43 AM		0	100%
Completed	39272	Release A 2022 (Manual)	5/6/2022 11:16:18 AM		0	100%
Needs investigation	39270	Regressions (Manual)	5/6/2022 11:16:44 AM		5	16.6%
Completed	39268	Release A 2022 (Manual)	5/6/2022 10:52:59 AM		0	100%
Completed	39266	Release A 2022 (Manual)	5/6/2022 10:52:51 AM		0	100%
Completed	39264	Rel A1 2022 (Manual)	5/6/2022 9:28:13 AM		0	100%
Completed	39262	Rel A1 2022 (Manual)	5/6/2022 9:01:06 AM		0	100%
Completed	39260	Rel A1 2022 (Manual)	5/6/2022 8:43:08 AM		0	100%
Completed	39258	Smoke Tests (Manual)	5/6/2022 6:11:51 AM		0	100%
Completed	39254	Regression (Manual)	5/5/2022 3:51:49 PM		0	100%
Completed	39252	Regression (Manual)	5/5/2022 3:48:51 PM		0	100%
Completed	39250	Regression (Manual)	5/5/2022 3:44:59 PM		0	100%
Completed	39248	Regression (Manual)	5/5/2022 3:43:42 PM		0	100%
Completed	39246	Regression (Manual)	5/5/2022 2:26:48 PM		0	100%
Completed	39242	Regression (Manual)	5/5/2022 2:14:30 PM		0	100%

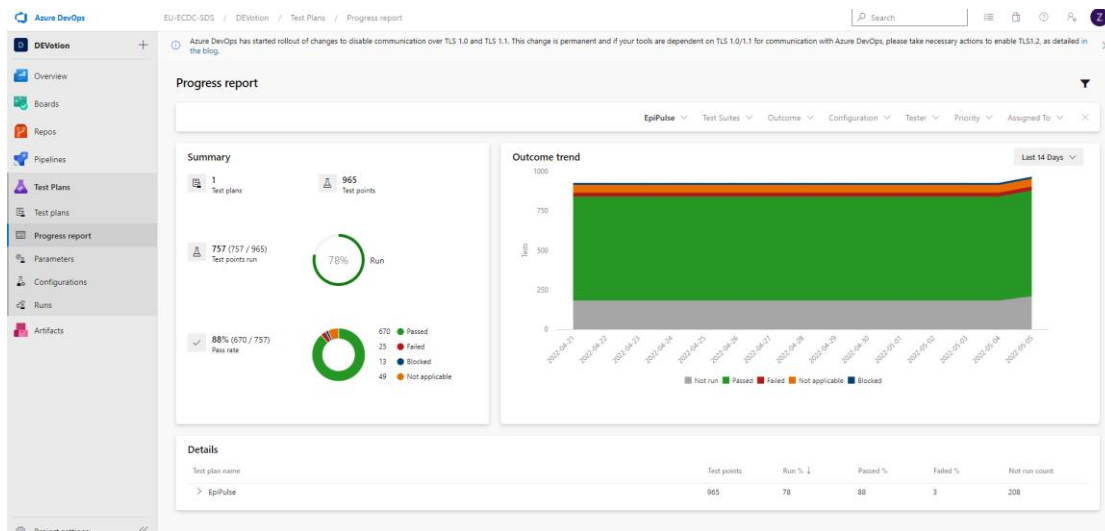
Σχήμα 52- Tests overview (from the system)

Διοίκηση έργων μέσω σύγχρονων μεθόδων: Μελέτη περίπτωσης το εργαλείο Microsoft Azure DevOps Χρυσούλα Ζάχαρη



Σχήμα 53- Testing dashboard (from the system)

Ένα χρήσιμο ακόμη dashboard για την ομάδα testing είναι η αναφορά προόδου όπου παρουσιάζει αθροιστικά τα ολοκληρωμένα επιτυχώς test cases, όσα έχουν σφάλματα και όσα αντίστοιχα εκκρεμούν προς δοκιμή.



Σχήμα 54- Progress report (from the system)

6

Συμπεράσματα

Στην παρούσα διπλωματική εργασία αρχικά παρουσιάστηκαν οι μεθοδολογίες διαχείρισης έργων που χρησιμοποιούνται για την αποτελεσματική υλοποίησή τους. Ιδιαίτερη έμφαση δόθηκε στη μεθοδολογία Agile όπου λόγω της ευελιξίας της αποτελεί πλέον την βέλτιστη πρακτική. Δόθηκε έμφαση στο DevOps που αποτελεί την επέκτασή της και διαδραματίζει σημαντικό ρόλο στην επιτυχή και ολοκληρωμένη υλοποίηση πληροφοριακών έργων. Στη συνέχεια παρουσιάστηκαν εκτενέστερα οι πλατφόρμες DevOps που εφαρμόζονται στο χώρο της πληροφορικής, παραθέτοντας τα οφέλη και τα αδύναμα σημεία της καθεμίας.

Τέλος, ιδιαίτερη έμφαση δόθηκε στην πλατφόρμα Microsoft Azure DevOps, η οποία αποτελεί και βασικό κομμάτι της παρούσας διπλωματικής εργασίας. Παρουσιάστηκε μια μελέτη περίπτωσης που χρησιμοποιεί την πλατφόρμα ως βασικό εργαλείο διαχείρισης του έργου, το οποίο συμπεριλαμβάνει τις ταυτόχρονες εργασίες από τα μέλη διαφορετικών ομάδων τόσο από την πλευρά της ομάδας υλοποίησης όσο και από τον πελάτη. Στο τέλος της εργασίας παρουσιάζονται και επεξηγούνται δεδομένα του έργου όπως αυτά καταγράφονται στην πλατφόρμα ώστε να καταστεί κατανοητή η εφαρμογή του.

Μέσα από το σύνολο της ανασκόπησης, γίνεται σαφές ότι η πλατφόρμα Azure DevOps αποτελεί μια επιλογή που συνδυάζει πολλές έτοιμες λύσεις, προσφέροντας γρήγορη εγκατάσταση και ωφελώντας άμεσα όποιον την επιλέξει. Επιπλέον, το γεγονός ότι πρόκειται για μέρος μιας ολοκληρωμένης

λύσης από μια μεγάλη εταιρεία της δίνει το πλεονέκτημα της γρήγορης ενσωμάτωσης με τα υπόλοιπα συστήματα της σουίτας αλλά και της γρήγορης δημιουργίας διεπαφών.

Κλείνοντας, να σημειωθεί πως αναμένεται ιδιαίτερη πρόοδος στις μεθοδολογίες του DevOps τα επόμενα χρόνια, με την προσθήκη της τεχνητής νοημοσύνης σε ακόμα περισσότερες συνιστώσες του.

7

Βιβλιογραφία

1. <https://kissflow.com/project/project-management-basics/#what-is-project-management>
2. <https://el.wikipedia.org>
3. [https://www.researchgate.net/publication/337744504 Evolution of Project Management Monitoring and Evaluation with Historical Events and Projects that Have Shaped the Development of Project Management as a Profession](https://www.researchgate.net/publication/337744504_Evolution_of_Project_Management_Monitoring_and_Evaluation_with_Historical_Events_and_Projects_that_Have_Shaped_the_Development_of_Project_Management_as_a_Profession)
4. [http://wiki.doing-projects.org/index.php/Teamweek \(virtual resource management tool\)](http://wiki.doing-projects.org/index.php/Teamweek_(virtual_resource_management_tool))
5. <https://www.workfront.com/project-management/methodologies>
6. <https://asana.com/resources/waterfall-project-management-methodology>
7. https://europa.eu/pm2/home_en
8. <https://startinfinity.com/project-management-methodologies/lean>
9. <https://lvivivity.com/lean-development-key-principles>
10. <https://www.projectmanager.com/critical-path-method>
11. <https://www.projectmanager.com/blog/prince2-methodology>
12. <https://kissflow.com/project/prince2-project-methodology/>
13. <https://www.digite.com/agile/agile-methodology/>
14. <https://www.businessnewsdaily.com/4987-what-is-agile-scrum-methodology.html>

15. <https://www.digite.com/agile/scrum-methodology/>
16. <https://www.digite.com/kanban/what-is-kanban/>
17. <https://kissflow.com/project/agile/kanban-methodology/>
18. <https://www.creatio.com/page/rapid-application-development>
19. <https://www.productplan.com/glossary/rapid-application-development/>
20. <https://medium.com/@raycad.seedotech/devops-methodology-and-process-dde388eb65bd>
21. <searchitoperations.techtarget.com/definition/DevOps>
22. <crashtest-security.com>
23. <www.whizlabs.com>
24. <www.simplilearn.com>
25. <www.varonis.com>
26. <aws.amazon.com>
27. <www.bmc.com>
28. <https://raygun.com/blog/best-devops-tools/>
29. <www.simplilearn.com>
30. <https://blog.inedo.com/best-practices-slack-devops>
31. <www.simplilearn.com>
32. <raygun.com>
33. <www.atlassian.com>
34. <www.supportpro.com>
35. <www.emtec.digital>
36. <www.devopsschool.com>
37. <www.ecdc.europa.eu>
38. <azure.microsoft.com>
39. <docs.microsoft.com>
40. <www.capterra.com>
41. <aspiresys.com/infrastructure-managed-services/devops/azure-devops-vs-jenkins-who-wins-the-battle/>

42. www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-selenium-test-automation-tool/
43. data-flair.training/blogs/advantages-and-disadvantages-of-docker/
44. www.researchdive.com