

ΕΛΛΗΝΙΚΟ ΜΕΣΟΓΕΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ



Διατμηματικό Μεταπτυχιακό Πρόγραμμα Σπουδών

«ΟΡΓΑΝΩΣΗ ΚΑΙ ΔΙΟΙΚΗΣΗ ΓΙΑ ΜΗΧΑΝΙΚΟΥΣ»

6ος κύκλος σπουδών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Διαχείριση κινδύνου σε έργα ανάπτυξης λογισμικού»

Όνομα: ΤΙΜΙΩΤΕΡΑΚΗ ΣΤΑΥΡΟΥΛΑ, MSc.-Eng.

ΑΜ: mto215

Επιβλέπουσα Καθηγήτρια: Δρ. Μαρία Μαρκάκη

Μόναχο/Ηράκλειο, Ιανουάριος, 2024

HELLENIC MEDITERRANEAN UNIVERSITY



Interdisciplinary Postgraduate Study Program

«MBA for Engineers»

Master Thesis

«Risk Management in Software Development Projects»

Name: Timioteraki Stavroula, MSc.-Eng.

Reg.Nr: mto215

Supervising Professor: Dr. Markaki Maria

Munich/Heraklion, January, 2024

Copyright © Σταυρούλα Τιμιωτεράκη, 2024

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Η έγκριση της πτυχιακής εργασίας από το πρόγραμμα δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα εκ μέρους του Τμήματος.

Ευχαριστίες

Καθότι μόνιμος κάτοικος Μονάχου Γερμανίας τα τελευταία 10 χρόνια, θα ήθελα να αρχίσω με αυτή τη φράση:

„Auch aus Steinen, die einem in den Weg gelegt werden, kann man Schönes bauen...“

«Ακόμη και από τις πέτρες που τοποθετούνται στο δρόμο σου για να σκοντάψεις, μπορείς να δημιουργήσεις κάτι όμορφο...»

- Goethe (Johan Wolfgang von Goeth)

Παρ' όλες τις δυσκολίες που μπορεί να αντιμετωπίσει κάποιος όντας κάτοικος εξωτερικού, μπορεί να χτίσει πολλά και όμορφα πράγματα. Ένα από αυτά είναι και μια επιτυχημένη καριέρα. Θέλω να ευχαριστήσω την εταιρεία που εργάζομαι εδώ και 6 χρόνια εδώ στη Γερμανία, καθώς με στήριξε με την κατανόηση και τη βοήθειά της σε όλη την επαγγελματική μου πορεία αλλά και την πορεία μου στο MBA, γεμίζοντάς με με όλα τα απαραίτητα εφόδια για την καριέρα μου στο Μάνατζμεντ και για την εκπόνηση της διπλωματικής μου εργασίας.

Ακόμη, θα ήθελα να ευχαριστήσω την οικογένειά μου που με στηρίζει και με εμπυχώνει τόσα χρόνια σε κάθε επιλογή μου και με στήριξε ακόμα μια φορά στο να επιτύχω ακόμα ένα στόχο μου, το MBA.

Τέλος, ευχαριστώ την επιβλέπουσα καθηγήτριά μου κ. Μαρκάκη για τη βοήθεια και την καθοδήγησή της για την εκπόνηση της διπλωματικής εργασίας μου.

Περίληψη

Στη σύγχρονη εποχή οι νέες τεχνολογίες έχουν εισβάλει σημαντικά στην καθημερινότητα των ανθρώπων. Για αυτό το λόγο ο αριθμός των οργανισμών που επενδύουν σε αυτές τις τεχνολογίες έχει αυξηθεί ραγδαία και η λειτουργία τους εξαρτάται από αυτές. Κάθε δυσλειτουργία, διακοπή ή παράνομη διείσδυση στα συστήματα αυτά μπορεί να επιφέρει οικονομικές απώλειες ή την αδυναμία του οργανισμού να ανταποκριθεί και να λειτουργήσει αποδοτικά.

Για να αποφευχθεί ένα σύνολο από ποικίλες απειλές που επιφέρει η υλοποίηση των έργων ανάπτυξης λογισμικού, είναι απαραίτητη η αναγνώριση και η αντιμετώπιση των κινδύνων που μπορεί να προκληθούν σε όλα τα στάδια επίτευξής τους.

Σκοπός της παρούσα διπλωματικής εργασίας είναι να ερευνηθούν και να αναγνωριστούν οι πιο σημαντικοί κίνδυνοι που αφορούν τα έργα ανάπτυξης λογισμικού σε κάθε φάση του κύκλου ζωής τους, οι οποίοι προκαλούν την αποτυχία τους. Η έρευνα διεξάγεται με τη μορφή ερωτηματολογίου, όπου οι ερωτηθέντες απασχολούνται σε έργα ανάπτυξης λογισμικού, είναι διαφόρων ειδικοτήτων και υπηρεσιών ενώ εργάζονται σε επιχειρήσεις της Ελλάδας και του εξωτερικού.

Πιο συγκεκριμένα, η παρούσα έρευνα χωρίζεται σε **3 ενότητες**.

Στην **πρώτη ενότητα** αναλύονται σημαντικές έννοιες του έργου, της διαχείρισης έργου και της διαχείρισης των έργων ανάπτυξης λογισμικού. Επίσης, αναλύεται ο κύκλος ζωής του έργου και των έργων ανάπτυξης λογισμικού, παρουσιάζονται σημαντικές διαφορές λόγω της φύσεως των έργων λογισμικού σε σχέση με άλλα έργα και δίνεται η έννοια της επιτυχίας τους. Τέλος, παρουσιάζονται κάποιες τεχνικές διαχείρισης.

Η **δεύτερη ενότητα** αφορά τη διαχείριση κινδύνου γενικότερα και ειδικότερα στα έργα ανάπτυξης λογισμικού. Δίνεται η έννοια της διαχείρισης κινδύνου, η διαδικασία που ακολουθείται καθώς ξεχωρίζεται η έννοια του κινδύνου όταν εκείνος αποτελεί ευκαιρία και όχι απειλή. Ακόμη, αναλύεται εκτενώς η βιβλιογραφία και συγκεκριμένα οι παράγοντες κινδύνου που παρατηρούνται στα έργα λογισμικού όπως αυτοί αναγνωρίζονται από άλλους ερευνητές.

Στην **τρίτη και τελευταία ενότητα** περιγράφεται η έρευνα, η μεθοδολογία που ακολουθήθηκε, τα αποτελέσματα αυτής όπως επίσης και τα συμπεράσματα που εξάγονται από αυτήν, τα οποία δείχνουν πως η διαχείριση κινδύνου στα έργα ανάπτυξης λογισμικού, και κυρίως στη φάση της ανάλυσης των προδιαγραφών του συστήματος, είναι σημαντική για την επιτυχία του έργου ενώ κάθε είδους καταγραφή στα έργα αυτά πρέπει να γίνεται με λεπτομερή και ποιοτικό τρόπο. Ακόμη, ο ανθρώπινος παράγοντας και η επικοινωνία μεταξύ διαχειριστών και ομάδας χρήζουν προσοχής για την αποφυγή της αποτυχίας.

Λέξεις κλειδιά: κίνδυνος, ρίσκο, παράγοντες κινδύνου, διαχείριση κινδύνου, έργα ανάπτυξης λογισμικού

Abstract

In modern times, new technologies have significantly invaded people's daily lives. For this reason, the number of organizations investing in these technologies has grown rapidly and their operation depends on them. Any malfunction, interruption or illegal intrusion into these systems may result in financial losses or the inability of the organization to respond and operate efficiently.

To avoid a set of various threats that brings the implementation of software development projects, it is necessary to recognize and deal with the risks that can be caused at all stages of their development.

The purpose of this thesis is to investigate and identify the most important risks related to software development projects in each phase of their life cycle, which cause their failure. The research is conducted in the form of a questionnaire, where the respondents are employed in software development projects, are of various specialties and nationalities while working in companies in Greece and abroad.

More specifically, this research is divided into **3 sections**.

The **first section** analyzes the meaning of project, project management and software development project management. Also, the life cycle of the project and software development projects is analyzed, significant differences due to the nature of software projects compared to other projects are presented and the concept of their success is given. Finally, some management techniques are presented.

The **second section** deals with risk management in general and in software development projects in particular. The concept of risk management and the followed process is given, as well as the concept of risk when it is an opportunity and not a threat is distinguished. Also, the literature is extensively analyzed and specifically the risk factors observed in software projects as recognized by other researchers.

The **third and last section** describes the research, the methodology followed, its results as well as the conclusions drawn from it, which show that risk management in software development projects, and especially in the phase of analyzing system specifications, is important for the success of the project, while any type of documentation in these projects must be done in a detailed and qualitative manner. Also, the human factor and communication between managers and team need attention to avoid failure.

Keywords: risk, risk factors, risk management, software development projects

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΕΧΟΜΕΝΑ	1
ΠΕΡΙΕΧΟΜΕΝΑ ΕΙΚΟΝΩΝ	3
ΠΕΡΙΕΧΟΜΕΝΑ ΠΙΝΑΚΩΝ	4
ΚΕΦΑΛΑΙΟ 1 - ΕΙΣΑΓΩΓΗ	6
1.1 ΕΙΣΑΓΩΓΗ	6
1.2 ΣΚΟΠΟΣ ΤΗΣ ΕΡΕΥΝΑΣ	7
1.3 ΔΟΜΗ ΤΗΣ ΕΡΕΥΝΑΣ	7
ΚΕΦΑΛΑΙΟ 2 - ΔΙΑΧΕΙΡΙΣΗ ΕΡΓΩΝ	9
2.1 ΔΙΑΧΕΙΡΙΣΗ ΕΡΓΟΥ	9
2.1.1 Η ΕΝΝΟΙΑ ΤΟΥ ΕΡΓΟΥ	9
2.1.2 Ο ΚΥΚΛΟΣ ΖΩΗΣ ΤΟΥ ΕΡΓΟΥ	10
2.1.3 Η ΕΝΝΟΙΑ ΤΗΣ ΔΙΑΧΕΙΡΙΣΗΣ ΕΡΓΟΥ	12
2.1.4 ΤΟ ΤΡΙΓΩΝΟ ΔΙΑΧΕΙΡΙΣΗΣ ΕΡΓΟΥ	12
2.1.5 Η ΕΝΝΟΙΑ ΤΗΣ ΕΠΙΤΥΧΙΑΣ	15
2.1.6 ΕΡΓΑΛΕΙΑ ΔΙΟΙΚΗΣΗΣ ΕΡΓΟΥ	17
2.2 ΔΙΑΧΕΙΡΙΣΗ ΕΡΓΩΝ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ	19
2.2.1 ΕΙΣΑΓΩΓΗ ΣΤΗ ΔΙΑΧΕΙΡΙΣΗ ΕΡΓΩΝ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ	19
2.2.2 Η ΦΥΣΗ ΤΩΝ ΕΡΓΩΝ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ	20
2.2.3 Ο ΚΥΚΛΟΣ ΖΩΗΣ ΤΩΝ ΕΡΓΩΝ ΛΟΓΙΣΜΙΚΟΥ - SOFTWARE DEVELOPMENT LIFECYCLE (SDLC).....	21
2.2.4 ΤΕΧΝΙΚΕΣ ΔΙΑΧΕΙΡΙΣΗΣ ΕΡΓΩΝ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ	23
ΚΕΦΑΛΑΙΟ 3 – ΔΙΑΧΕΙΡΙΣΗ ΚΙΝΔΥΝΟΥ	26
3.1 Η ΕΝΝΟΙΑ ΤΗΣ ΔΙΑΧΕΙΡΙΣΗΣ ΚΙΝΔΥΝΟΥ	26

3.2 ΤΥΠΟΙ ΚΙΝΔΥΝΩΝ	26
3.3 Η ΔΙΑΔΙΚΑΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΚΙΝΔΥΝΟΥ	27
3.5 Ο ΚΙΝΔΥΝΟΣ ΩΣ ΕΥΚΑΙΡΙΑ	31
ΚΕΦΑΛΑΙΟ 4 – ΔΙΑΧΕΙΡΙΣΗ ΚΙΝΔΥΝΟΥ ΣΕ ΕΡΓΑ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ	33
4.1 ΕΙΣΑΓΩΓΗ	33
4.2 ΠΑΡΑΓΟΝΤΕΣ ΚΙΝΔΥΝΟΥ ΕΡΓΩΝ ΛΟΓΙΣΜΙΚΟΥ - ΒΙΒΛΙΟΓΡΑΦΙΑ	34
4.3 ΔΙΑΧΕΙΡΙΣΗ ΚΙΝΔΥΝΟΥ ΣΕ ΕΡΓΑ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ	37
4.3.1 Η ΦΥΣΗ ΤΩΝ ΕΡΓΩΝ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ ΣΤΗ ΔΙΑΧΕΙΡΙΣΗ ΚΙΝΔΥΝΩΝ	39
4.3.2 Η ΕΝΝΟΙΑ ΤΗΣ ΕΠΙΤΥΧΙΑΣ ΣΤΑ ΕΡΓΑ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ	41
ΚΕΦΑΛΑΙΟ 5 – ΜΕΘΟΔΟΛΟΓΙΑ	43
5.1 ΣΥΛΛΟΓΗ ΤΩΝ ΔΕΔΟΜΕΝΩΝ	43
5.2 ΣΥΝΤΑΞΗ ΤΟΥ ΕΡΩΤΗΜΑΤΟΛΟΓΙΟΥ	45
5.3 ΑΝΑΛΥΣΗ ΤΩΝ ΔΕΔΟΜΕΝΩΝ	46
ΚΕΦΑΛΑΙΟ 6 – ΑΠΟΤΕΛΕΣΜΑΤΑ	51
6.1 ΔΗΜΟΓΡΑΦΙΚΑ ΣΤΟΙΧΕΙΑ	51
6.2 ΑΠΟΤΕΛΕΣΜΑΤΑ ΑΝΑΛΥΣΗΣ ΔΕΔΟΜΕΝΩΝ	56
6.3 ΠΙΝΑΚΑΣ ΚΙΝΔΥΝΟΥ	64
ΚΕΦΑΛΑΙΟ 7 – ΣΥΜΠΕΡΑΣΜΑΤΑ	68
7.1 ΣΥΜΠΕΡΑΣΜΑΤΑ ΕΡΕΥΝΑΣ ΚΑΙ ΣΥΖΗΤΗΣΗ	68
7.2 ΣΥΓΚΡΙΣΗ ΕΡΕΥΝΑΣ ΜΕ ΤΗ ΒΙΒΛΙΟΓΡΑΦΙΑ	70
ΒΙΒΛΙΟΓΡΑΦΙΑ	73
ΠΑΡΑΡΤΗΜΑ	76

ΠΕΡΙΕΧΟΜΕΝΑ ΕΙΚΟΝΩΝ

Εικόνα 1: Κύκλος ζωής του έργου (Σιώκου, 2021)	10
Εικόνα 2: Το τρίγωνο του Barnes (Ζαχαροπούλου, 2021)	13
Εικόνα 3: Στάδια SDLC (Abdelrafe , Burairah, & Norhaziah, 2016)	22
Εικόνα 4: Παραδείγματα μοντέλων διαχείρισης έργων λογισμικού (Abdelrafe , Burairah, & Norhaziah, 2016)	25
Εικόνα 5: Τύποι κινδύνων (Βίτσιος, 2018)	27
Εικόνα 6: Διαδικασία διαχείρισης κινδύνου (Φιτσιλής, 2015)	28
Εικόνα 7: Πίνακας κινδύνου (Μαρκάκη, 2021)	29
Εικόνα 8: Σχέδιο απόκρισης σε κίνδυνο (Μαρκάκη, 2021)	30
Εικόνα 9: Βήματα διαχείρισης κινδύνου στα έργα ανάπτυξης λογισμικού (Enfei, 2015)	38
Εικόνα 10: Μεθοδολογία συλλογής και ανάλυσης δεδομένων	44
Εικόνα 11: Διάγραμμα φύλου των ερωτηθέντων	51
Εικόνα 12: Διάγραμμα εκπαίδευσης των ερωτηθέντων	52
Εικόνα 13: Διάγραμμα εκπαίδευσης των ερωτηθέντων	52
Εικόνα 14: Έτη προϋπηρεσίας των ερωτηθέντων	53
Εικόνα 15: Τίτλοι εργασίας των ερωτηθέντων	54
Εικόνα 16: Κλάδος απασχόλησης των ερωτηθέντων	55
Εικόνα 17: Μέγεθος των επιχειρήσεων που απασχολούνται οι ερωτηθέντες	56

ΠΕΡΙΕΧΟΜΕΝΑ ΠΙΝΑΚΩΝ

Πίνακας 1: Υπολογισμός μέσης σταθμισμένης απόκρισης απαντήσεων κάθε ερώτησης	46
Πίνακας 2: Συνοπτικός πίνακας υπολογισμού μέσης σταθμισμένης απόκρισης απαντήσεων κάθε ερώτησης	47
Πίνακας 3: Ενδεικτικός πίνακας υπολογισμού αποδοχής ή μη αποδοχής παραγόντων κινδύνου.....	48
Πίνακας 4: Πίνακας υπολογισμού πιθανότητας κινδύνου κατά PMI (Project Management Institute).....	49
Πίνακας 5: Πίνακας υπολογισμού συνέπειας κινδύνου κατά PMI (Project Management Institute).....	49
Πίνακας 6: Πίνακας υπολογισμού έκθεσης κινδύνου κατά PMI (Project Management Institute).....	50
Πίνακας 7: Δημοφιλείς παράγοντες κινδύνου κατά τη φάση του Σχεδιασμού του έργου	57
Πίνακας 8: Δημοφιλείς παράγοντες κινδύνου κατά τη φάση της Ανάλυσης/Δημιουργίας προδιαγραφών συστήματος.....	57
Πίνακας 9: Δημοφιλείς παράγοντες κινδύνου κατά τη φάση του Σχεδιασμού του συστήματος	58
Πίνακας 10: Δημοφιλείς παράγοντες κινδύνου κατά τη φάση της Υλοποίησης	59
Πίνακας 11: Δημοφιλείς παράγοντες κινδύνου κατά τη φάση των Δοκιμών.....	60
Πίνακας 12: Δημοφιλείς παράγοντες κινδύνου κατά τη φάση της Υποστήριξης.....	61
Πίνακας 13: Δημοφιλείς παράγοντες κινδύνου σε όλες τις φάσεις που αφορούν τον άνθρωπο	61
Πίνακας 14: Δημοφιλείς παράγοντες κινδύνου σε όλες τις φάσεις που αφορούν τον πελάτη, την επιχείρηση και τη διαχείριση του έργου	62
Πίνακας 15: Κορυφαίοι 19 κίνδυνοι στα έργα ανάπτυξης λογισμικού σύμφωνα με την έρευνα.....	63
Πίνακας 16: Έκθεση των 19 κορυφαίων κινδύνων	64

Πίνακας 17: Πίνακας κινδύνου των 19 κορυφαίων κινδύνων	66
Πίνακας 18: Σχέδιο απόκρισης των 19 κορυφαίων κινδύνων	66
Πίνακας 19: Συγκριτικός πίνακας κινδύνων έρευνας με τη βιβλιογραφία	70

ΚΕΦΑΛΑΙΟ 1 - ΕΙΣΑΓΩΓΗ

1.1 ΕΙΣΑΓΩΓΗ

Η λέξη κίνδυνος (risk) παίρνει τις ρίζες της από τη λατινική λέξη “risicare”, η οποία σημαίνει τολμώ. Ο κίνδυνος, δηλαδή, δεν αποτελεί τόσο πεπρωμένο αλλά περισσότερο επιλογή. Οι δράσεις που τολμούμε να αναλάβουμε, οι οποίες εξαρτώνται από τις επιλογές που διαθέτουμε, ουσιαστικά χαρακτηρίζουν την «ιστορία» της διαχείρισης κινδύνων αλλά και της κοινωνικής εξέλιξης. Ο κίνδυνος σημαίνει έκθεση στην αβεβαιότητα που αφορά αφενός τις ανθρώπινες δραστηριότητες ως καθημερινός τρόπος ζωής και σκέψης, και αφετέρου όλες τις επιχειρήσεις. Για το λόγο αυτό, χρειάζεται να υπάρχει κάποια προετοιμασία για την υποδοχή και αντιμετώπισή του (Τσαλαμάνδρης, 2014).

Στην έννοια του ρίσκου εμπεριέχεται και η έννοια της αβεβαιότητας, η οποία πολλές φορές επηρεάζει τις αποφάσεις ενός ανθρώπου ή οργανισμού επιφέροντας επιθυμητά ή ανεπιθύμητα αποτελέσματα. Η αβεβαιότητα προκύπτει συνήθως λόγω έλλειψης γνώσης, πολυπλοκότητας του έργου, χρηματοοικονομικής ανακρίβειας ή ανταγωνισμού (Νταρζάνου, 2012). Η διαφορά μεταξύ του ρίσκου και της αβεβαιότητας είναι ότι το ρίσκο παρουσιάζεται κυρίως σε κάποιον άνθρωπο ή οργανισμό ενώ η αβεβαιότητα έχει ένα γενικό χαρακτήρα και αφορά πολλούς ανθρώπους ή επιχειρήσεις.

Ο κίνδυνος είναι ένα αβέβαιο γεγονός που μπορεί να έχει αρνητικό αποτέλεσμα (απειλές) ή θετικό (ευκαιρίες). Οι απειλές και οι ευκαιρίες οφείλουν να εξετάζονται όχι μόνο στα πλαίσια του έργου αλλά και σε σχέση με τους ενδιαφερόμενους που μπορεί να επηρεασθούν (Νταρζάνου, 2012). Η διαχείριση του κινδύνου είναι μια σημαντική και απαραίτητη διαδικασία για κάθε οργανισμό, καθώς αποτελεί τον κεντρικό πυρήνα της διαχείρισης στρατηγικής. Είναι η διεργασία με την οποία οι οργανισμοί προσεγγίζουν μεθοδικά τους κινδύνους που σχετίζονται με τις δραστηριότητές τους με σκοπό να επωφεληθούν σε κάθε μια από αυτές (Τσαλαμάνδρης, 2014).

Η διαχείριση κινδύνου έχει άμεση σχέση με την επιτυχία του έργου, καθώς όταν αναγνωρίζονται και ο χειρισμός των παραγόντων κινδύνου γίνεται πιο εύκολος και αυξάνει τις πιθανότητες να επιτευχθεί ένα επιτυχημένο αποτέλεσμα που θα αποτελέσει

και ένα επιτυχημένο έργο ή, έστω, μπορεί να αποφευχθεί η αποτυχία του έργου. Ένα έργο θεωρείται επιτυχημένο όταν ισορροπούν οι παράγοντες «χρόνος», «κόστος» και «ποιότητα», που θεωρούνται και οι πιο σημαντικοί παράγοντες για την επιτυχία του έργου, ή όταν το αποτέλεσμα ανταποκριθεί στις προσδοκίες των πελατών (Νταρζάνου, 2012). Στην επιτυχία του έργου συμβάλλει στενά ο διαχειριστής του έργου και η ομάδα του.

Στα έργα ανάπτυξης λογισμικού η διαχείριση του έργου καθώς και η διαχείριση κινδύνου είναι λίγο περισσότερο πολύπλοκη από ότι σε έργα άλλης φύσεως, για το λόγο ότι απαιτούν διαχειριστές με υψηλές τεχνικές δεξιότητες για την ανάλυση, την εκτέλεση και την υποστήριξη του παραδοτέου λογισμικού. Ως εκ τούτου, οι κίνδυνοι που οδηγούν σε αποτυχία ενός έργου λογισμικού είναι απαραίτητο να αναγνωρισθούν πριν από την εφαρμογή μιας οργανωτικής καινοτομίας.

1.2 ΣΚΟΠΟΣ ΤΗΣ ΕΡΕΥΝΑΣ

Σκοπός της παρούσας εργασίας είναι η ανάλυση της έννοιας «Διαχείριση κινδύνου» στα έργα ανάπτυξης λογισμικού και η αναγνώριση των σημαντικών παραγόντων που την πλαισιώνουν. Μέσα από αυτή την έρευνα θα γίνουν κατανοητοί οι παράγοντες αυτοί και θα είναι εφικτή η αντιμετώπιση των ρίσκων που προκαλούν.

1.3 ΔΟΜΗ ΤΗΣ ΕΡΕΥΝΑΣ

Η παρούσα έρευνα αποτελείται από **7 κεφάλαια**:

Στο **κεφάλαιο 1** γίνεται αναφορά στην έννοια του κινδύνου, της διαχείρισης αυτού και των έργων ανάπτυξης λογισμικού, ενώ προσδιορίζεται ο σκοπός της έρευνας.

Στο **κεφάλαιο 2** αναλύεται, αρχικά, η έννοια του έργου και ο κύκλος ζωής αυτού όπως επίσης και η διαχείριση του έργου και το τρίγωνο διαχείρισης. Στη συνέχεια, περιγράφεται η έννοια της επιτυχίας, τι θεωρείται, δηλαδή, επιτυχημένο έργο σύμφωνα με τη βιβλιογραφία και έπειτα παρουσιάζονται κάποια εργαλεία διαχείρισης.

Στη συνέχεια δίνεται η έννοια της διαχείρισης όπως αυτή συμβαίνει στα έργα ανάπτυξης λογισμικού, περιγράφεται η φύση των έργων αυτών και σε τι διαφέρουν από τα έργα άλλης φύσεως καθώς και ο κύκλος ζωής αυτών. Τέλος, παρουσιάζονται κάποιες τεχνικές διαχείρισης που χρησιμοποιούνται στα έργα ανάπτυξης λογισμικού.

Στο **κεφάλαιο 3** αναφέρεται η έννοια της διαχείρισης κινδύνου και αναλύεται η διαδικασία αυτής ενώ παρουσιάζεται η θετική πλευρά του κινδύνου όταν αυτός μπορεί να θεωρηθεί ευκαιρία.

Στο **κεφάλαιο 4** γίνεται αναφορά στη διαχείριση κινδύνου στα έργα ανάπτυξης λογισμικού και στη διαφορά αυτής με τη διαχείριση κινδύνου σε άλλα έργα λόγω της φύσεως αυτών. Στη συνέχεια, παρουσιάζεται η βιβλιογραφική επισκόπηση της έρευνας και οι παράγοντες κινδύνου που παρουσιάζονται από άλλους ερευνητές. Τέλος, περιγράφεται η έννοια της επιτυχίας των έργων ανάπτυξης λογισμικού, το τι, δηλαδή, θεωρείται επιτυχημένο έργο λογισμικού σύμφωνα με τη βιβλιογραφία.

Στο **κεφάλαιο 5** περιγράφεται η μεθοδολογία της έρευνας που ακολουθήθηκε, το πως συλλέχθηκαν και αναλύθηκαν τα δεδομένα και πως συντάχθηκε το ερωτηματολόγιο της έρευνας.

Στο **κεφάλαιο 6** παρουσιάζονται τα αποτελέσματα της έρευνας (δημογραφικά στοιχεία και αποτελέσματα ανάλυσης δεδομένων) ενώ παράλληλα δημιουργείται ο πίνακας κινδύνου.

Τέλος, στο **κεφάλαιο 7**, αναφέρονται τα συμπεράσματα της έρευνας.

ΚΕΦΑΛΑΙΟ 2 - ΔΙΑΧΕΙΡΙΣΗ ΕΡΓΩΝ

2.1 ΔΙΑΧΕΙΡΙΣΗ ΕΡΓΟΥ

2.1.1 Η ΕΝΝΟΙΑ ΤΟΥ ΕΡΓΟΥ

Ως έργο ορίζεται από το Ινστιτούτο Διαχείρισης Έργων (Project Management Institute, PMI) «μια προσωρινή προσπάθεια που αναλαμβάνεται για τη δημιουργία ενός μοναδικού προϊόντος, υπηρεσίας ή αποτελέσματος» (PMBOK, 2008)

Η προσωρινή του φύση υποδηλώνει ότι το έργο είναι ορισμένου χρόνου αλλά ανεξαρτήτου διάρκειας. Έχει, δηλαδή, αρχή και τέλος, το οποίο επιτυγχάνεται όταν επιτυγχάνονται και οι στόχοι του έργου ή όταν δεν μπορούν να επιτευχθούν ή ακόμη και όταν διακόπτεται η ανάγκη για τη δημιουργία του έργου. Παράλληλα, με τον όρο «μοναδικό» εννοείται ότι κάθε προϊόν ή υπηρεσία που παράγεται από το έργο διαφέρει με κάποιο διακριτό τρόπο από άλλα συναφή προϊόντα ή υπηρεσίες. (Νταρζάνου, 2012)

Κατά τον Turner, το έργο ορίζεται ως «μια προσπάθεια κατά την οποία συνεργάζονται με νέο και διαφορετικό τρόπο ανθρωπίνοι, οικονομικοί και υλικοί πόροι με σκοπό την ανάληψη ενός μοναδικού αντικειμένου εργασίας, δεδομένων προδιαγραφών και εντός περιορισμένου κόστους και χρόνου, έτσι ώστε να επιτευχθούν ωφέλιμες αλλαγές μέσω ποσοτικών και ποιοτικών στόχων.» (Turner, 1993)

Σε ένα έργο μπορεί να εμπλέκεται μόνο ένα πρόσωπο, ένα τμήμα μιας επιχείρησης ή και πολλαπλά τμήματα μιας επιχείρησης σε συνεργασία. Ένα έργο μπορεί να αποτελέσει ένα προϊόν (ή μια υπηρεσία) μόνο του ή να δημιουργήσει ένα προϊόν ή υπηρεσία από ένα στοιχείο κάποιου άλλου προϊόντος ή υπηρεσίας. Ακόμη, μπορεί να θεωρηθεί έργο και η εκτέλεση μιας υπηρεσίας ή το αποτέλεσμα που προκύπτει από μια έρευνα. (PMBOK, 2008)

Συνοψίζοντας, μπορούμε να ορίσουμε το έργο ως ένα σύνολο διαδικασιών που σχεδιάζονται για την επίτευξη ενός μοναδικού στόχου και πραγματοποιούνται εντός προσδιορισμένων προδιαγραφών και εντός συγκεκριμένου χρόνου και κόστους ενώ τα χαρακτηριστικά του είναι: (Νταρζάνου, 2012) (Σιώκου, 2021)

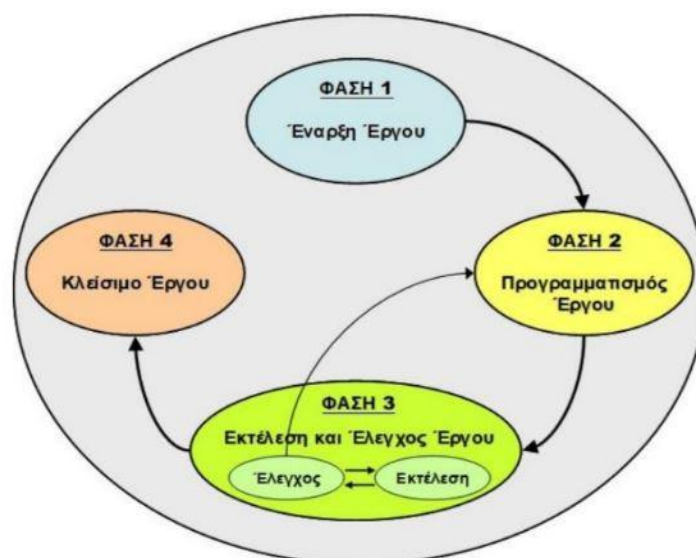
- Η χρονική διάρκεια (έχει αρχή και τέλος)

- Η μοναδικότητα (έχει συγκεκριμένο περιεχόμενο και αποσκοπεί στη δημιουργία ενός μοναδικού προϊόντος, υπηρεσίας ή αποτελέσματος)
- Οι περιορισμοί (χρονικοί, οικονομικοί, ποιοτικοί κτλ)
- Οι διαθέσιμοι πόροι (απασχολεί ανθρώπινους και υλικούς πόρους)
- Η ύπαρξη ομάδας (υπάρχει συνεργασία μεταξύ διαφορετικών ειδικοτήτων για να υλοποιηθεί ένα έργο)
- Είναι προσωρινό (με την ολοκλήρωσή του θα μετασχηματιστεί η ομάδα ή θα διαλυθεί)
- Διαθέτει διάφορες φάσεις και μη επαναλαμβανόμενες δραστηριότητες
- Ο σχεδιασμός (επιτυγχάνονται οι στόχοι με επιτυχία)

2.1.2 Ο ΚΥΚΛΟΣ ΖΩΗΣ ΤΟΥ ΕΡΓΟΥ

Όπως αναφέρθηκε παραπάνω, ένα έργο χαρακτηρίζεται από τις διάφορες φάσεις που διασχίζει από την έναρξη έως την ολοκλήρωσή του. Η όλη διαδικασία που διανύει καθώς και ο τρόπος που συνδέονται οι φάσεις μεταξύ τους, ονομάζεται κύκλος ζωής του έργου. Ο κύκλος ζωής διαφέρει από έργο σε έργο λόγω της μοναδικότητας και των αναγκών κάθε έργου. Έτσι, δεν υπάρχει κάποιος συγκεκριμένος ιδανικός κύκλος για όλα τα έργα. Παρ' όλα αυτά, υπάρχουν τέσσερις βασικές φάσεις (Εικόνα 1): (Σιώκου, 2021)

1. Έναρξη του έργου
2. Προγραμματισμός του έργου
3. Εκτέλεση και έλεγχος των εργασιών του έργου
4. Λήξη έργου



Εικόνα 1: Κύκλος ζωής του έργου (Σιώκου, 2021)

Αναλυτικότερα, στην πρώτη φάση εντοπίζεται η ανάγκη ή το πρόβλημα, τα οποία θα επιλύσει το έργο και γίνεται η σύλληψη της ιδέας. Περιγράφονται λεπτομερώς οι εναλλακτικές λύσεις που καλείται το έργο να δώσει με την εκτέλεσή του, αναλύονται οι στόχοι και ο σκοπός του, οι οποίοι θα συμφωνούν με τους στόχους της επιχείρησης ή του οργανισμού, ορίζονται οι προδιαγραφές, η ομάδα εκτέλεσης του έργου, οι επιχειρηματικοί κίνδυνοι, το κόστος, τα κριτήρια επιτυχίας καθώς και τα χρονοδιαγράμματα υλοποίησης. Τέλος, αυτό το σχέδιο (business case), παραδίδεται στην επιχείρηση ή τον οργανισμό και αξιολογείται η εφικτότητά του για να εγκριθεί.

Εφόσον εγκριθεί από την οργανωτική επιτροπή, μπορεί να συνεχίσει στο επόμενο στάδιο που είναι ο προγραμματισμός ή αλλιώς, ο σχεδιασμός του έργου. Σε αυτό το στάδιο προσδιορίζονται λεπτομερώς οι δραστηριότητες που χρειάζεται να ακολουθηθούν, τα χρονοδιαγράμματα (συνήθως με ένα διάγραμμα Gantt), οι πόροι, τα κόστη, η ποιότητα του έργου, οι κίνδυνοι, η επικοινωνία που θα χρειαστεί κατά τη διάρκεια του έργου (εσωτερική μεταξύ εργαζομένων και οργανισμού αλλά και εξωτερική μεταξύ οργανισμού και πελάτη) και τέλος το σχέδιο αποδοχής του παραδοτέου. Επίσης, γίνεται και ο καθορισμός των δεικτών απόδοσης, με τους οποίους θα παρακολουθείται η πορεία του έργου και θα αξιολογείται θέτοντας μέτρο σύγκρισης τους προκαθορισμένους στόχους και σκοπούς του έργου.

Η φάση της εκτέλεσης και του ελέγχου είναι η πιο χρονοβόρα αλλά ταυτόχρονα και η πιο σημαντική. Στον οδηγό (PMBOK, 2008) αλλά και σε άλλες βιβλιογραφίες, συχνά αναφέρεται αυτή η φάση ως δύο διαφορετικές φάσεις, μια αυτή της εκτέλεσης και μια αυτή του ελέγχου. Σε αυτή τη φάση εκτελούνται οι δραστηριότητες και οι εργασίες που έχουν προσδιοριστεί σύμφωνα με το χρονοδιάγραμμα της προηγούμενης φάσης ώστε να παραχθεί το προϊόν ή υπηρεσία και να δοθεί στον πελάτη για να το αποδεχθεί. Το έργο, σε αυτό το στάδιο, παρακολουθείται από τον υπεύθυνο και ελέγχεται για τυχόν σφάλματα που μπορεί να προκύπτουν κατά τη διάρκεια της υλοποίησης. Παράλληλα, ελέγχονται και οι κίνδυνοι που ορίστηκαν στο προηγούμενο στάδιο, ενώ γίνεται αναζήτηση νέων που μπορεί να εμφανιστούν κατά τη διάρκεια εκτέλεσης του έργου (risk monitoring) (Νταρζάνου, 2012). Τέλος, ελέγχεται κατά πόσο η ομάδα του έργου έχει επιτύχει τους στόχους που έχουν ορισθεί.

Αφού ολοκληρωθούν οι εργασίες και υπάρχει ικανοποίηση για την πορεία του έργου τότε μπορεί να οδηγηθεί στην επόμενη και τελική φάση. Η λήξη του έργου σε αυτό το στάδιο, σημαίνει η παράδοση του τελικού προϊόντος ή υπηρεσίας στον πελάτη, η παράδοση της τεκμηρίωσής του στον οργανισμό, η αποδέσμευση των πόρων καθώς και όλα τα απαραίτητα βήματα για τη λήξη της σύμβασης μεταξύ πελάτη και οργανισμού. Επίσης, γίνεται η τελική αξιολόγηση του έργου ώστε να ορισθεί το επίπεδο επιτυχίας του. Ακόμη, σε αυτή τη φάση, αναγνωρίζονται και αξιολογούνται τυχόν λάθη που προέκυψαν κατά της διάρκεια του κύκλου ζωής του έργου ώστε να αποφευχθούν σε μελλοντικά έργα (Lessons Learned). (Νταρζάνου, 2012)

2.1.3 Η ΕΝΝΟΙΑ ΤΗΣ ΔΙΑΧΕΙΡΙΣΗΣ ΕΡΓΟΥ

Όπως αναφέρθηκε παραπάνω, το έργο αποτελεί ένα προσωρινό και μοναδικό εγχείρημα, το οποίο διανύει ορισμένες φάσεις ώστε να ολοκληρωθεί. Η οργάνωση όλων των διαδικασιών που θα επιφέρουν την ικανοποιητική απόδοση αυτού του εγχειρήματος και την ορθή εκτέλεση των δραστηριοτήτων, ονομάζεται διαχείριση έργου. (Σιώκου, 2021)

Σύμφωνα με το Ινστιτούτο Διαχείρισης Έργων (Project Management Institute, PMI) η διαχείριση έργου ορίζεται ως η διαδικασία κατά την οποία εφαρμόζονται γνώσεις, δεξιότητες, εργαλεία και τεχνικές καθ' όλη τη διάρκεια εκτέλεσης των δραστηριοτήτων του έργου, με στόχο να ικανοποιηθούν οι απαιτήσεις και οι προσδοκίες των ενδιαφερόμενων μερών (stakeholders), καθώς επίσης και να επιτευχθούν οι στόχοι του έργου. (PMBOK, 2008)

Η διαχείριση έργου στοχεύει στην αξιοποίηση και την αποτελεσματική χρήση όλων των παραγωγικών πόρων που διαθέτει ώστε να βελτιώσει το σχεδιασμό και τους ελέγχους σε όλη τη διάρκεια των διαδικασιών του έργου αλλά και να μειώσει το κόστος υλοποίησης και το χρόνο ολοκλήρωσης καθώς και αυτοί είναι σημαντικοί παράγοντες που επηρεάζουν άμεσα την εξέλιξη του έργου. Παράλληλα, αξιοποιώντας τους παραπάνω παράγοντες στοχεύει επίσης στην αύξηση της ποιότητας του έργου και του τελικού προϊόντος ή υπηρεσίας, συνεπώς και την αξιοπιστία τους. (Σιώκου, 2021)

Η διαχείριση έργου μπορεί να εφαρμόζεται είτε από ένα άτομο (όπως τον υπεύθυνο του έργου) είτε από διαφορετικές ομάδες διαχείρισης, από τις οποίες η καθεμία αναλαμβάνει να διαχειριστεί και από ένα διαφορετικό κομμάτι του έργου ή του κύκλου ζωής. Αν το έργο θα είναι διαχειρίσιμο από ένα άτομο ή από μια ομάδα ατόμων, επιλέγεται αναλόγως τις ανάγκες του. Η ανάγκη για διαχείριση του έργου από διαφορετικές ομάδες διαχείρισης προκύπτει από την πολυπλοκότητα των έργων καθώς τα τελευταία έτη παρουσιάζεται αύξηση στο μέγεθος των έργων και των απαιτήσεών του. (Σιώκου, 2021)

2.1.4 ΤΟ ΤΡΙΓΩΝΟ ΔΙΑΧΕΙΡΙΣΗΣ ΕΡΓΟΥ

“On time, On budget, On scope, On quality”

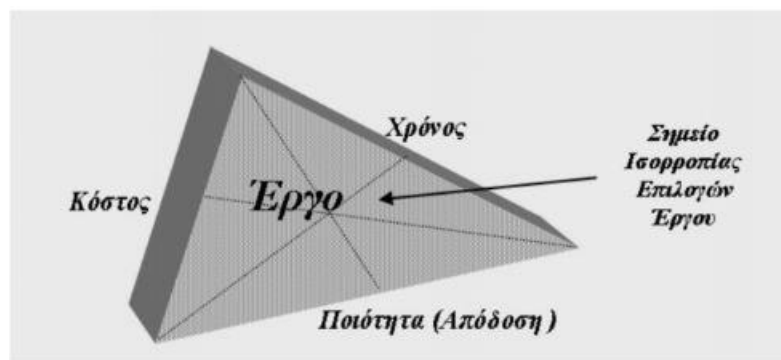
Η διαχείριση έργου καλείται να απαντήσει σε ερωτήματα όπως «πόσο θα διαρκέσει», «πόσο θα κοστίσει», «γιατί το συγκεκριμένο έργο» κτλ. Συνεπώς, η διαχείριση έργου είναι εκείνη που καθορίζει ποιες ενέργειες είναι απαραίτητο να εκτελεστούν ώστε να οδηγηθεί στην επιτυχία.

Ένα έργο θεωρείται πως έχει επιτύχει όταν με την ολοκλήρωσή του προσεγγίστηκαν οι στόχοι για τους οποίους δημιουργήθηκε, που τέθηκαν στην αρχή, και σύμφωνα με τα κριτήρια αποδοχής τους εντός συγκεκριμένου χρονοπρογραμματισμού και

προϋπολογισμού. (Ζαχαροπούλου, 2021) Η επιτυχία του έργου βασίζεται σε τρεις παράγοντες που αποτελούν τους δείκτες ώστε να εκτιμηθεί η επιτυχία:

- Ο χρόνος (τήρηση του χρονοδιαγράμματος)- time
- Το κόστος (τήρηση του προϋπολογισμού)- budget
- Η ποιότητα (τήρηση των ποιοτικών προδιαγραφών)- quality

Αν αυτοί οι δείκτες υπερβούν ο ένας τον άλλο τότε υπάρχει πιθανότητα αποτυχίας του έργου. Με βάση αυτούς τους παράγοντες δημιουργήθηκε ένα τρίγωνο διαχείρισης έργου από τον Barnes το 1969, όπως αυτό παρουσιάζεται παρακάτω στην (Ζαχαροπούλου, 2021)Εικόνα 2.Εικόνα 2: Το τρίγωνο του Barnes



Εικόνα 2: Το τρίγωνο του Barnes (Ζαχαροπούλου, 2021)

Παρατηρείται, λοιπόν, στο τρίγωνο, πως αυτοί οι τρεις παράγοντες εμφανίζονται ως αλληλοεξαρτώμενες διαστάσεις ώστε να φτάσουν στο επιθυμητό αποτέλεσμα που είναι η επιτυχία (score). Τα ενδιαφερόμενα μέρη πρέπει να ορίσουν από την αρχή την ιεραρχία της προτεραιότητας του κάθε παράγοντα σύμφωνα με το σκοπό του έργου και, έτσι ώστε να προσπαθήσει να επέλθει η χρυσή τομή των τριών αυτών παραγόντων, η οποία θα αποτελέσει την ισορροπία τους. (Ζαχαροπούλου, 2021) Οποιαδήποτε μεταβολή συμβεί στο κόστος, το χρόνο ή την απόδοση, τότε επηρεάζεται αυτόματα κάθε πλευρά του τριγώνου είτε ελαφρώς είτε επιφέροντας σημαντικές αλλαγές στο έργο. (Σιώκου, 2021)

Έστω, για παράδειγμα, ότι σε ένα έργο έχει ορισθεί σε προτεραιότητα το κόστος, έπειτα η ποιότητα και στη συνέχεια ο χρόνος. Τότε ποιότητα και ο χρόνος θα πρέπει να προσαρμοστούν ιεραρχικά σύμφωνα με το κόστος. Αν το κόστος αλλάξει κατά τη διάρκεια του έργου, τότε θα επηρεάσει σημαντικά και την ποιότητα και τον χρόνο, οι οποίοι παράγοντες θα πρέπει να προσαρμοστούν αντίστοιχα ώστε να επέλθει η ισορροπία.

Το σημείο ισορροπίας παρουσιάζεται εκεί που γίνεται ο καλύτερος συνδυασμός των τριών αυτών δεικτών, γι' αυτό και λέγεται πως είναι συνήθως ανταγωνιστικοί. Με τον όρο ανταγωνιστικοί, εννοείται πως η απαραίτητη τήρηση του προϋπολογισμού μπορεί

να οδηγήσει σε εκπτώσεις της ποιότητας ή και σε χρονικές υπερβάσεις, η τήρηση του χρονοδιαγράμματος χωρίς καμία παρέκκλιση μπορεί να οδηγήσει σε κοστολογική υπέρβαση ή και σε εκπτώσεις της ποιότητας ενώ, τέλος, η απαρέγκλιτη τήρηση των ποιοτικών προδιαγραφών μπορεί να οδηγήσει σε κοστολογική ή και σε χρονική υπέρβαση. (Βίτσιος, 2018)

Είναι σημαντικό, συνεπώς, να εκτελεσθούν οι παρακάτω διαδικασίες, οι οποίες σύμφωνα με τον (PMBOK, 2008) ταξινομούνται σε δέκα **γνωστικές περιοχές**, προκειμένου να επέλθει η παραπάνω ισορροπία και να υπάρξει μια επιτυχημένη διοίκηση έργου:

- **Διαχείριση ενοποίησης (project integration management):** ενοποιεί τις βασικές διαδικασίες που εφαρμόζονται στη διοίκηση έργου με σκοπό να τις συντονίσει. Το σημαντικότερο παραδοτέο της διαχείρισης έργου είναι το σχέδιο διοίκησης έργου (project management plan), το οποίο αποτελεί στόχο της ενοποίησης. Κατά την ενοποίηση γίνεται, επίσης, και η διαδικασία της ανασκόπησης των αιτημάτων των αλλαγών (change requests), η έγκριση ή απόρριψή τους όπως και ο έλεγχος των αλλαγών σε παραδοτέα (deliverables). Η διαδικασία, αυτή, ονομάζεται ολοκληρωμένος έλεγχος αλλαγών (integrated change control) και αποτελεί βασική λειτουργία στα πλαίσια της ενοποίησης. (Φιτσιλής, 2015)
- **Διαχείριση αντικειμένου εργασιών (project scope management):** δημιουργείται μια λεπτομερής έκθεση του αντικειμένου του έργου (project scope statement) αλλά και η παρουσίαση των απαραίτητων διεργασιών υπό τη μορφή διαγράμματος WBS (Work Breakdown Structure), το οποίο αναλύεται παρακάτω, ώστε να προσδιοριστούν όλες οι απαραίτητες εργασίες που πρέπει να συμβούν ώστε να ολοκληρωθεί το έργο. (Φιτσιλής, 2015)
- **Διαχείριση χρόνου (project time management):** δημιουργείται το χρονοδιάγραμμα του έργου, το οποίο είναι απαραίτητο για το συντονισμό του προσωπικού, των πόρων, των εργασιών κτλ ενώ στοχεύει στον προσδιορισμό των δραστηριοτήτων, την αλληλουχία τους, τη διάρκειά τους, τους περιορισμούς κ.ά με σκοπό την έγκαιρη παράδοση του έργου. Εάν καθυστερήσουν να εκτελεσθούν οι επιμέρους δραστηριότητες του έργου θα οδηγηθεί σε αύξηση κόστους και καθυστέρηση του έργου. (Φιτσιλής, 2015)
- **Διαχείριση κόστους (project cost management):** συντάσσεται ο προϋπολογισμός του έργου (budget) κοστολογώντας τις δραστηριότητες και τους πόρους που απαιτούνται και παρακολουθείται καθ' όλη τη διάρκεια του έργου ώστε να μην υπάρχουν μεγάλες αποκλίσεις από την αρχική εκτίμηση. (Φιτσιλής, 2015)
- **Διαχείριση ποιότητας (project quality management):** η ποιότητα του έργου εξασφαλίζει ότι το αποτέλεσμα ικανοποιεί τις ανάγκες για τις οποίες δημιουργήθηκε. Κατά τη διαχείριση ποιότητας περιλαμβάνει το σχεδιασμό της ποιότητας (quality planning), τον έλεγχο (quality control) αλλά και τη διασφάλιση (quality assurance) αυτής. Υπάρχουν πολλοί τρόποι να μετρήσει κανείς την ποιότητα και οι δείκτες μέτρησης προσδιορίζονται κατά το σχέδιο ποιότητας του έργου (project quality plan). (Φιτσιλής, 2015)

- **Διαχείριση ανθρώπινων πόρων (project human resource management):** στόχος της διαχείρισης ανθρώπινων πόρων είναι η εύρεση της πιο αποτελεσματικής και αποδοτικής ομάδας του έργου, η οποία χρειάζεται να συνδυάζει ειδικότητες αλλά και τμήματα της επιχείρησης ώστε να φέρει εις πέρας το έργο στην ώρα του και εντός του προϋπολογισμού (on time, on budget). (Φιτσιλής, 2015)
- **Διαχείριση επικοινωνιών (project communications management):** στη θεματική ενότητα της επικοινωνίας προσδιορίζονται τα κανάλια επικοινωνίας καθώς και ο τρόπος, η συχνότητα, οι συμμετέχοντες αλλά και το περιεχόμενο. (Φιτσιλής, 2015) Για παράδειγμα, κάθε εβδομάδα θα υπάρχει μια τηλεδιάσκεψη μεταξύ πελάτη και διαχειριστή έργου όπου θα ενημερώνει για την εξέλιξη του έργου και θα λαμβάνει πληροφορίες από τον πελάτη που θα αναφέρει εσωτερικά στην ομάδα του έργου. Οι πληροφορίες που παρέχονται εσωτερικά και εξωτερικά του οργανισμού πρέπει να είναι πολύ προσεκτικές και να προσδιορίζονται εξ' αρχής.
- **Διαχείριση κινδύνων (project risk management):** οι κίνδυνοι του έργου αναγνωρίζονται, αναλύονται και δημιουργείται ένα σχέδιο ώστε να ελεγχθούν και να αντιμετωπιστούν όταν εμφανιστούν (η διαχείριση κινδύνου αναλύεται σε παρακάτω κεφάλαιο)
- **Διαχείριση προμηθειών (project procurement management):** ορίζονται τα προϊόντα και οι υπηρεσίες που θα προμηθευτεί η επιχείρηση από πηγές εκτός αυτής ή εκτός της ομάδας εκτέλεσης του έργου, καθώς και ο τρόπος που θα τα διαχειριστεί. (Φιτσιλής, 2015)
- **Διαχείριση συμμετεχόντων (stakeholder management):** συμμετέχοντες στο έργο εννοούνται όλοι όσοι επηρεάζονται ή επηρεάζουν το έργο (άτομο, ομάδα ή οργανισμός). Η αποτελεσματική διαχείριση των ενδιαφερόμενων μερών εξασφαλίζει την επιτυχή υλοποίηση του έργου μέσω της διαχείρισης προσδοκιών και συμφωνημένων στόχων ενώ παράλληλα δημιουργεί θετικές εντυπώσεις για την ολοκλήρωση του έργου και το παραδοτέο. (Φιτσιλής, 2015)

2.1.5 Η ΕΝΝΟΙΑ ΤΗΣ ΕΠΙΤΥΧΙΑΣ

Όπως αναφέρθηκε και παραπάνω, ένα έργο χαρακτηρίζεται ως επιτυχημένο όταν ολοκληρώνεται εντός κάποιου προσδιορισμένου χρονικού διαστήματος ή πριν τη λήξη αυτού με βάση ενός ορισμένου χρονοδιαγράμματος, εντός ορισμένου προϋπολογισμού, χωρίς να τον υπερβεί, και εφόσον αποδίδει και συμφωνεί με τις προδιαγραφές του πελάτη, ενώ ταυτόχρονα φτάνει ή έστω προσεγγίζει κατά πολύ τους στόχους για τους οποίους δημιουργήθηκε. Παρ' όλα αυτά, είναι δύσκολο να χαρακτηριστεί ένα έργο επιτυχημένο ή αποτυχημένο καθώς η εκτίμηση αυτή έχει άμεση σχέση με την οπτική των ενδιαφερόμενων μερών ως προς το τι θεωρούν επιτυχία. Η αξιολόγηση της επιτυχίας το έργου γίνεται στη φάση της ολοκλήρωσης του κύκλου ζωής του έργου.

Θεωρείται πως η έννοια της επιτυχίας του έργου και η έννοια της επιτυχίας της διοίκησης του έργου είναι δύο διαφορετικές έννοιες όμως άμεσα συνδεδεμένες η μία με την άλλη και δύσκολο να διαχωριστούν εντελώς. Η επιτυχία του έργου βασίζεται στην επίτευξη των στόχων και την ικανοποίηση των προσδοκιών ενώ η επιτυχία της διοίκησης του έργου σχετίζεται με την αξιολόγηση του χρόνου, του κόστους και της ποιότητας (Radujkonića & Sjekavica, 2017). Παράλληλα, ένα έργο μπορεί να θεωρηθεί επιτυχές ακόμη και αν η διαχείρισή του ήταν κακή και αντίστροφα (Βάκκας, 2020).

Το τρίγωνο του Barnes δημιουργήθηκε για να αξιολογήσει την επιτυχία ενός έργου με βάση το κόστος, το χρόνο, την ποιότητα και το στόχο του έργου. Ήταν από τις πρώτες προσπάθειες αξιολόγησης και αποτέλεσε τον πυρήνα ώστε να αναπτυχθούν και άλλες προσεγγίσεις αξιολόγησης (Βάκκας, 2020), οι οποίες περιλαμβάνουν το στόχο και το σκοπό του έργου, την επιτυχία του έργου ως επιτυχία του προϊόντος (Wideman, 2009) ή την ικανοποίηση των ενδιαφερόμενων μερών (Lewis, 2010). Άλλες, πάλι, προσεγγίσεις αναφέρονται στους επιχειρηματικούς στόχους, στα οφέλη που προκάλεσε και τις επιπτώσεις που είχε η ολοκλήρωση του έργου στον οργανισμό στον οποίο ανήκει το έργο καθώς υποστηρίζουν πως το κόστος, ο χρόνος και η ποιότητα μπορεί να διαφέρουν από το αρχικό σχέδιο όμως το έργο είναι επιτυχές εφόσον ο στόχος του συμφωνεί με το στόχο της επιχείρησης (Radujkonića & Sjekavica, 2017) (Lech, 2013).

Υπεύθυνος για τη διαχείριση του έργου και, συνεπώς, για την επιτυχία του έργου είναι ο διαχειριστής του έργου ή η ομάδα διαχείρισης, εάν πρόκειται για ένα πολύπλοκο έργο που απαιτεί περισσότερους από ένα διαχειριστές. Ο διαχειριστής έργου εκτελεί ταυτόχρονα πολλές εργασίες ενώ λαμβάνει αποφάσεις σχετικά με την χρήση των πόρων, τις προσαρμογές του χρονοδιαγράμματος, αποφάσεις σε ότι αφορά τον προϋπολογισμό, διαχειρίζεται τις ανθρώπινες σχέσεις (για παράδειγμα σχέσεις μεταξύ εργαζομένων και διαχείριση και επίλυση συγκρούσεων μεταξύ τους), την επικοινωνία (ομάδας έργου αλλά και μεταξύ πελάτη και οργανισμού) και κάθε είδους προβλήματα και αλλαγές που μπορεί να προκύψουν. Με την επιτυχία του έργου έρχεται και η επιτυχία του διαχειριστή.

Σύμφωνα με τον (Balachandra, 1984) οι παράγοντες αξιολόγησης επιτυχίας στη διαχείριση ενός έργου έχουν άμεση σχέση με το έργο του διαχειριστή και διατυπώνονται ως εξής: (Βίσιος, 2018)

- **Στόχος έργου:** Ο ορισμός σαφώς διατυπωμένων στόχων αποτελεί βασική αρχή για τον προγραμματισμό και την εκτέλεση ενός έργου. Όλοι οι συμμετέχοντες στο έργο πρέπει να είναι εξ' αρχής ενήμεροι για τους στόχους του έργου και να τους κατανοήσουν.
- **Υποστήριξη από την ανώτατη διοίκηση:** Η συνεχής παρουσία της διοίκησης καθ' όλη τη διάρκεια ζωής του έργου συνεισφέρει στην κατανόηση του στόχου του έργου και της σημασίας του ενώ προλαμβάνει συγκρούσεις που μπορεί να προκύψουν λόγω αβεβαιότητας. Η συχνή επικοινωνία μεταξύ του διαχειριστή του έργου και της διοίκησης δρα καταλυτικά για την επιτυχία ενός έργου.

- **Προγραμματισμός έργου:** Η μετατροπή, δηλαδή, του στόχου και των μέτρων απόδοσης σε ένα εφικτό και λεπτομερές σχέδιο που καλύπτει θέματα τεχνικά, οικονομικά, οργανωτικά, επικοινωνιακά, κινδύνων, ελέγχου και χρονοδιαγράμματος και θα αποτελεί τη βάση για την υλοποίηση.
- **Συνεργασία με τον πελάτη:** Ο πελάτης είναι εκείνος που θα κρίνει κατά κύριο λόγο την επιτυχία του έργου. Ακόμη και αν παραδοθεί σε εκείνον εγκαίρως, σύμφωνα με τις επιθυμητές προδιαγραφές, και εντός προϋπολογισμού, αλλά δε χρησιμοποιηθεί ποτέ ή σπανίως μπορεί με βεβαιότητα να θεωρηθεί αποτυχία.
- **Θέματα προσωπικού:** Η καλή συνεργασία μεταξύ των εμπλεκόμενων στο έργο και η αφοσίωσή τους είναι απαραίτητες για την επιτυχία. Αξίζει να σημειωθεί πως η επιτυχία σχετίζεται άμεσα και με την εργασιακή ικανοποίηση. Αν το προσωπικό που απασχολείται στο έργο δεν μένει ικανοποιημένο με την εργασία του ή την απόδοσή του, θα έχει άμεση επίπτωση και στην επιτυχία του έργου.
- **Τεχνικά θέματα:** Η τεχνική κατάρτιση του προσωπικού και η εκπλήρωση των τεχνικών προδιαγραφών πρέπει να ληφθούν σοβαρά υπόψη από το διαχειριστή του έργου καθώς χωρίς αυτές ένα έργο δεν μπορεί να ολοκληρωθεί.
- **Αποδοχή από τον πελάτη:** Κάθε πελάτης θέτει τα δικά του κριτήρια αποδοχής για το έργο και η αξιολόγηση γίνεται στα τελικά στάδια της υλοποίησης. Η συνεχής επικοινωνία με τον πελάτη σε όλη τη διάρκεια του κύκλου ζωής ενός έργου αυξάνει την πιθανότητα επιτυχίας όσον αφορά την αποδοχή από τον τελικό χρήστη, που είναι ο πελάτης. Σε περίπτωση που το έργο δε γίνει αποδεκτό, τότε είναι αποτυχημένο.
- **Έλεγχος έργου:** Η συνεχής ροή πληροφοριών σχετικά με την πραγματική πρόοδο και η ανατροφοδότηση συμβάλουν στο να αναγνωριστούν νωρίς οποιεσδήποτε παρεκκλίσεις από τα αρχικά σχέδια και να διορθωθούν.
- **Επικοινωνία:** Ο καλός συντονισμός ανάμεσα στους συμμετέχοντες σε κάθε φάση του έργου απαιτεί μία συνεχή ανταλλαγή πληροφοριών. Οι επίσημες γραμμές επικοινωνίας, καθώς και η ανεπίσημη ροή ανάμεσα στα μέλη της ομάδας συνηγορούν στην επιτυχία του έργου.
- **Επίλυση προβλημάτων:** Η διαθεσιμότητα προετοιμασμένων σχεδίων και μηχανισμών για τη διαχείριση προβλημάτων μπορεί να μειώσει τον κόπο που μπορεί να χρειαστεί ώστε να επιλυθούν εάν δεν υπήρχαν αυτά.

2.1.6 ΕΡΓΑΛΕΙΑ ΔΙΟΙΚΗΣΗΣ ΕΡΓΟΥ

Η διαχείριση έργων υλοποιείται με την βοήθεια μιας σειράς εργαλείων και τεχνικών, τα οποία εντάσσονται στις εκάστοτε διαδικασίες. (Μαρκάκη, 2021) Τα εργαλεία αυτά συμπεριλαμβάνουν :

- την Δομή Ανάλυσης Εργασιών (WBS- Work Breakdown Structure),
- τη μέθοδο κρίσιμης διαδρομής (CPM- Critical Path Method)
- το διάγραμμα Gantt,

- Εργαλεία Διαχείρισης Ποιότητας και Κινδύνων

Αναλυτικότερα, η **Δομή Ανάλυσης Εργασιών (WBS- Work Breakdown Structure)** είναι η ιεραρχική ανάλυση του έργου σε πακέτα εργασίας, η οποία καταλήγει σε δραστηριότητες με σκοπό να διευκολύνει την εκτίμηση, το σχεδιασμό, την παρακολούθηση, τον έλεγχο και την ανάθεση των δραστηριοτήτων στα αντίστοιχα άτομα της ομάδας του έργου και αποτελεί το βασικότερο εργαλείο στη διαχείριση έργων. (Μαρκάκη, 2021) (Φιτσιλής, 2015)

Η **μέθοδος κρίσιμης διαδρομής (CPM- Critical Path Method)** είναι η γραφική αναπαράσταση των δραστηριοτήτων που απεικονίζει τη σειρά με την οποία θα εκτελεστούν οι δραστηριότητες συμπεριλαμβανομένης της διάρκειάς τους αλλά και την εξάρτηση κάποιων δραστηριοτήτων από κάποιες άλλες. Για παράδειγμα, για να εκτελεσθεί η δραστηριότητα Γ πρέπει να εκτελεσθεί πρώτα η δραστηριότητα Β ενώ η Α υποδέχεται τη Β.

Από την άλλη, το **διάγραμμα Gantt** είναι ένα οριζόντιο ιστόγραμμα και αποτελεί το χρονοδιάγραμμα του έργου, το οποίο προσδιορίζει τη συνολική διάρκεια του έργου καθώς και την χρονική έναρξη και λήξη κάθε δραστηριότητας. Το διάγραμμα Gantt είναι βασικό εργαλείο παρακολούθησης της συνολικής χρονικής πορείας του έργου ενώ, παράλληλα, μπορεί να ελέγξει κάθε δραστηριότητα ώστε να μην ξεφύγει από τα χρονικά της πλαίσια.

Η ποιότητα μπορεί να μετρηθεί ως προς την αξιοπιστία του προϊόντος ή υπηρεσίας που παράγεται από την εκτέλεση του έργου, ως προς την απόδοσή του, την ευκολία χρήσης κτλ ενώ διαφέρει από έργο σε έργο ανάλογα τη φύση του αλλά αξιολογείται ως προς την ικανοποίηση των αναγκών όλων των ενδιαφερόμενων μερών. Για παράδειγμα, η ποιότητα μιας ιστοσελίδας μπορεί να μετρηθεί ως προς την ταχύτητα φόρτωσης στον ιστό ενώ ένα κατασκευαστικό έργο ανάλογα με τη σταθερότητα του κτηρίου κτλ.

Σε όλα τα έργα υπάρχουν επτά βασικά εργαλεία, τα οποία χρησιμοποιούνται για τη **διαχείριση της ποιότητας και κινδύνου** με σκοπό τον έλεγχο της ποιότητας και του κινδύνου του έργου. (Βικιπαίδεια, χ.χ.)

1. Διάγραμμα ροής ή πίνακας ροής
2. Φύλλο ελέγχου
3. Διάγραμμα αιτίας-αποτελέσματος
4. Διάγραμμα Pareto
5. Πίνακας ελέγχου
6. Ιστόγραμμα
7. Διάγραμμα Scatter

2.2 ΔΙΑΧΕΙΡΙΣΗ ΕΡΓΩΝ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ

2.2.1 ΕΙΣΑΓΩΓΗ ΣΤΗ ΔΙΑΧΕΙΡΙΣΗ ΕΡΓΩΝ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ

Η διαχείριση έργου ανάπτυξης λογισμικού αναφέρεται στον προγραμματισμό, την οργάνωση και τον έλεγχο εργασιών και πόρων για την επίτευξη ενός συγκεκριμένου στόχου, που συνήθως είναι η παράδοση ενός προϊόντος λογισμικού, εντός συγκεκριμένου χρονικού διαστήματος, προϋπολογισμού και σύμφωνα με τις καθορισμένες απαιτήσεις του συστήματος. Περιλαμβάνει τον συντονισμό των προσπάθειών μιας ομάδας επαγγελματιών, όπως προγραμματιστές, σχεδιαστές, δοκιμαστές (testers) και άλλα ενδιαφερόμενα μέρη, για την επίτευξη των στόχων του έργου.

Πολλά έργα πληροφορικής περιλαμβάνουν έρευνα, ανάλυση και στη συνέχεια αγορά και εγκατάσταση νέου υλικού και λογισμικού με ελάχιστη ή καθόλου πραγματική ανάπτυξη λογισμικού. Άλλα έργα περιλαμβάνουν μικρές τροποποιήσεις για την ενίσχυση του υφιστάμενου λογισμικού ή για την ενσωμάτωση μιας εφαρμογής σε μια άλλη. Ακόμα άλλα έργα περιλαμβάνουν ένα μεγάλο μέρος της ανάπτυξης λογισμικού. Πολλοί υποστηρίζουν ότι η ανάπτυξη λογισμικού απαιτεί από τους διαχειριστές έργων να τροποποιούν τις παραδοσιακές μεθόδους διαχείρισης έργου, ανάλογα με τον κύκλο ζωής συγκεκριμένου προϊόντος (Βάκκας, 2020).

Η διαχείριση έργου ανάπτυξης λογισμικού είναι μια σειρά δραστηριοτήτων που σχετίζονται με την όσο το δυνατόν αποτελεσματικότερη εκτέλεση του έργου. Οι βασικές πτυχές της διαχείρισης έργου λογισμικού είναι, όπως αναφέρθηκε και παραπάνω, ο χρόνος, το κόστος και η απόδοση. Οι τρεις αυτές μεταβλητές απαιτούν συνεχή παρακολούθηση για την ομαλή διεξαγωγή των εργασιών και την επίτευξη των στόχων, συνεπώς, και την επιτυχία του έργου. (Jurison, September, 1999).

Ο χρόνος αναφέρεται στο αν η πρόοδος της υλοποίησης είναι εντός σχεδιασμένου χρονοδιαγράμματος, το κόστος αναφέρεται στις δαπάνες για τους πόρους του έργου ενώ η απόδοση είναι ο βαθμός στον οποίο επιτυγχάνονται οι στόχοι ή οι προδιαγραφές. Υπάρχει μια ακόμη μεταβλητή που υπολογίζεται πλέον στη διαχείριση του έργου. Αυτή είναι η ικανοποίηση του πελάτη η οποία συμβάλει στην επιτυχία του έργου (βλ. κεφάλαιο επιτυχία έργου).

Η επιτυχημένη διαχείριση έργου στην ανάπτυξη λογισμικού απαιτεί συνδυασμό τεχνικών γνώσεων, ηγετικών δεξιοτήτων και ικανότητας προσαρμογής στις μεταβαλλόμενες συνθήκες. Ο στόχος είναι να παραδοθεί ένα προϊόν υψηλής ποιότητας που να ανταποκρίνεται στις ανάγκες του πελάτη, ενώ παράλληλα διαχειρίζεται τους περιορισμούς όπως ο χρόνος και ο προϋπολογισμός αποτελεσματικά.

2.2.2 Η ΦΥΣΗ ΤΩΝ ΕΡΓΩΝ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ

Σε γενικές γραμμές, τα έργα ανάπτυξης λογισμικού έχουν αρκετές ομοιότητες με έργα άλλης φύσεως, όπως για παράδειγμα τεχνικά έργα. Στις περισσότερες περιπτώσεις οι στόχοι, οι διαδικασίες, και τα εργαλεία που χρησιμοποιούνται είναι παρόμοια. Διαφοροποιείται όμως ο τρόπος εφαρμογής τους λόγω των διαφορών που προκύπτουν από τη φύση του λογισμικού ως προϊόντος (Φιτσιλής, 2015).

Το λογισμικό είναι ως επί το πλείστον αόρατο ή αλλιώς άυλο και τα έργα λογισμικού τείνουν επίσης να είναι αόρατα. Για να είναι επιτυχημένα, οι διαχειριστές έργου πρέπει να κάνουν ορατό το προϊόν (το λογισμικό που αναπτύσσεται) και το έργο (τη διαδικασία ανάπτυξης) (Φιτσιλής, 2015). Οι στόχοι του έργου, οι απαιτήσεις συστήματος, τα σχέδια έργου, οι κίνδυνοι του έργου, οι ατομικές ευθύνες και η κατάσταση του έργου πρέπει να είναι ορατά και κατανοητά από όλα τα εμπλεκόμενα μέρη. Μόνο τότε μπορεί η ομάδα του έργου να λάβει τεκμηριωμένες αποφάσεις και να έχει μια αρκετά καλή ευκαιρία για επιτυχία (Jurison, September, 1999).

Μια μεγάλη διαφορά των έργων πληροφορικής σε σχέση με τα υπόλοιπα έργα, όπως για παράδειγμα τα κατασκευαστικά έργα, είναι οι περιορισμοί που έγκειται η φύση του έργου. Οι μόνοι περιορισμοί των έργων πληροφορικής είναι το κόστος και ο χρόνος και αυτό για το λόγο ότι η πληροφορική είναι μια επιστήμη που στηρίζεται στις δυνατότητες του ανθρώπου να αξιοποιεί τα υλικά και τους πόρους που διαθέτει (Παπαδημητρίου, 2009). Τα τελευταία χρόνια, όμως, και η πληροφορική απέκτησε περιορισμούς νομικής φύσεως, όπως για παράδειγμα ένα έργο δεν πρέπει να παραβεί τη νομοθεσία της προστασίας προσωπικών δεδομένων ή να συμφωνεί με τους κανόνες του ασφαλούς διαδικτύου (cybersecurity). Ακόμη, το έργο μπορεί να διεξάγεται για τις ανάγκες του κράτους (ελληνικού ή μη) ή της Ευρωπαϊκής Ένωσης και έτσι να πληροί αντίστοιχες νομοθετικές προδιαγραφές. Για παράδειγμα στις Ηνωμένες Πολιτείες Αμερικής το λογισμικό που χρησιμοποιείται στο αυτοκίνητο οφείλει να συμμορφώνεται σε διαφορετική νομοθεσία από αυτήν των χωρών της Ευρωπαϊκής Ένωσης.

Μια άλλη ιδιαιτερότητα που αντιμετωπίζουν τα έργα πληροφορικής σε σχέση με τα έργα άλλης φύσεως, είναι ότι μπορούν πάντοτε να βελτιώνονται, να αναβαθμίζονται και να επεκτείνονται. Παραδείγματος χάρη, μια γέφυρα δε μπορεί να επεκταθεί αφού τελειώσει ως έργο, παρά μόνο να συντηρηθεί (Παπαδημητρίου, 2009).

Η εξέλιξη της τεχνολογίας αποτελεί βασική διαφορά σε σχέση με τα έργα διαφορετικής φύσεως. Τα έργα λογισμικού πρέπει να συμμορφώνονται στις ραγδαία εξέλιξη της τεχνολογίας με συνέπεια τη δυσκολία διαχείρισης των τεχνολογικών αλλαγών. Αποτελεί συνηθισμένο φαινόμενο στα έργα ανάπτυξης λογισμικού η ύπαρξη προβλημάτων ολοκλήρωσης μεταξύ διαφορετικών εκδόσεων του λογισμικού, η ταυτόχρονη συνύπαρξη εργαλείων διαφορετικών εκδόσεων κ.λπ. Συνεπώς η αλλαγή της τεχνολογίας αποτελεί για τα έργα ανάπτυξης λογισμικού έναν σταθερό παράγοντα αστάθειας που θα πρέπει να λαμβάνουμε πάντα σοβαρά υπόψη. (Φιτσιλής, 2015)

Ένα άλλο χαρακτηριστικό των έργων λογισμικού είναι η πολυπλοκότητα. Τα έργα λογισμικού μπορεί να είναι εξαιρετικά πολύπλοκα, που περιλαμβάνουν περίπλοκους αλγόριθμους, εξελιγμένες αρχιτεκτονικές και ενοποίηση με διάφορα συστήματα. Η

διαχείριση αυτής της πολυπλοκότητας απαιτεί εξειδικευμένες δεξιότητες και βαθιά κατανόηση των αρχών της μηχανικής λογισμικού.

Τέλος, σύνθητες φαινόμενο στα έργα λογισμικού είναι να αντιμετωπίζουν αβεβαιότητες ως προς τις απαιτήσεις, καθώς οι πελάτες μπορεί να μην έχουν πάντα σαφή και πλήρη κατανόηση του τι χρειάζονται. Αυτό καθιστά δύσκολο τον καθορισμό ενός σταθερού συνόλου απαιτήσεων στην αρχή του έργου, οδηγώντας σε μεγαλύτερη εξάρτηση από την επαναληπτική ανάπτυξη και τους βρόχους ανάδρασης.

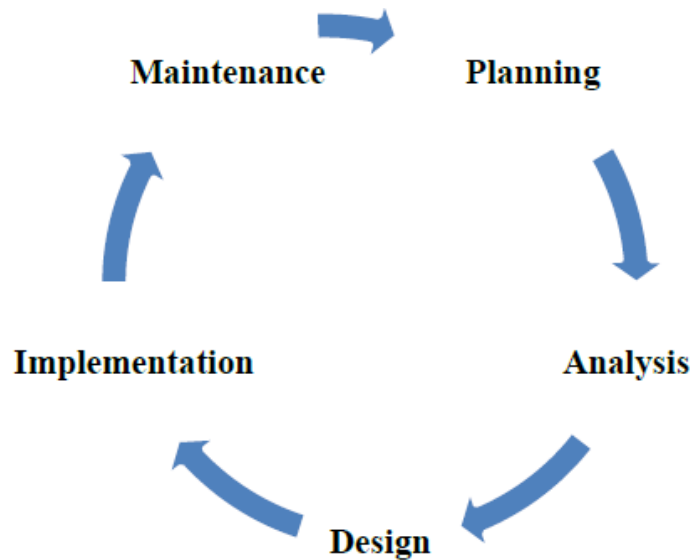
Η κατανόηση αυτών των διαφορών είναι απαραίτητη για τους διαχειριστές έργων και τις ομάδες για τον αποτελεσματικό σχεδιασμό, εκτέλεση και έλεγχο έργων ανάπτυξης λογισμικού. Τονίζεται, επίσης, η ανάγκη για μια προσαρμοσμένη προσέγγιση διαχείρισης έργου που να ευθυγραμμίζεται με τα μοναδικά χαρακτηριστικά της ανάπτυξης λογισμικού.

2.2.3 Ο ΚΥΚΛΟΣ ΖΩΗΣ ΤΩΝ ΕΡΓΩΝ ΛΟΓΙΣΜΙΚΟΥ - SOFTWARE DEVELOPMENT LIFECYCLE (SDLC)

Οι ιδιαιτερότητες των έργων πληροφορικής ή οι διαφορές με τα άλλα έργα που αναφέρθηκαν στο παραπάνω υποκεφάλαιο, αναπαρίστανται και στη μεθοδολογία. Τα προγράμματα ανάπτυξης λογισμικού αποτελούν ένα υποσύνολο των έργων πληροφορικής. Στα έργα ανάπτυξης λογισμικού ακολουθείται η μέθοδος SDLC (Software Development Lifecycle), που αναπαριστά τον κύκλο ζωής του έργου λογισμικού και αποτελείται από ακολουθιακά στάδια (Παπαδημητρίου, 2009).

Στη βιβλιογραφία η μέθοδος SDLC (Software Development Lifecycle) ορίζεται ως μια δομή, η οποία επιβάλλεται να ακολουθείται για την ανάπτυξη έργων λογισμικού και αποτελείται από έξι φάσεις, καθεμία από τις οποίες παράγει παραδοτέα που απαιτούνται από την επόμενη φάση (Hijazi, Alqrainy, Muaidi, & Khmour, A Framework for Integrating Risk Management into the Software Development Process, 2014).

Σε κάποια βιβλιογραφικά άρθρα αριθμούνται τα στάδια ως πέντε βασικές φάσεις (Hijazi, Alqrainy, Muaidi, & Khmour, RISK FACTORS IN SOFTWARE DEVELOPMENT PHASES, January 2014), (Abdelrafe, Burairah, & Norhaziah, 2016) καθώς δεν περιλαμβάνουν τις δοκιμές (testing) ως ξεχωριστή φάση ενώ σε άλλες (Hossain, 2023) παρατηρείται να χωρίζονται σε επτά συμπεριλαμβανομένης και της διαδικασίας της ανάπτυξης πέραν της υλοποίησης.



Εικόνα 3: Στάδια SDLC (Abdelrafe , Burairah, & Norhaziah, 2016)

Πιο αναλυτικά, τα πέντε βασικά στάδια που αναφέρονται στη βιβλιογραφία είναι τα ακόλουθα (Abdelrafe , Burairah, & Norhaziah, 2016), (Arafeh & El-Ahmad, 2017):

- **Προγραμματισμός:** Το αρχικό στάδιο ενός έργου ανάπτυξης λογισμικού είναι αυτό που καθορίζει το σκοπό και το στόχο του έργου, την οργάνωση και την αξιολόγηση των υπάρχοντων συστημάτων λογισμικού καθώς και τι πρέπει να κάνει το νέο σύστημα σε σχέση με τα υπάρχοντα συστήματα.
- **Ανάλυση:** Περιλαμβάνει την εξέταση του υπάρχοντος συστήματος ενώ συγκεντρώνονται όλες οι απαιτήσεις του έργου και καθορίζονται η υπηρεσίες του συστήματος, οι στόχοι και οι περιορισμοί. Δημιουργείται ένα έγγραφο προδιαγραφών απαιτήσεων που θα χρησιμοποιηθεί στην επόμενη φάση.
- **Σχεδιασμός:** Σε αυτή τη φάση καθιερώνεται η συνολική αρχιτεκτονική του συστήματος σύμφωνα με τις απαιτήσεις που ορίστηκαν στην προηγούμενη φάση. Αποτελεί στην ουσία την πραγματική δημιουργία και σχεδιασμό του συστήματος.
- **Υλοποίηση:** Η πραγματική κατασκευή του λογισμικού ξεκινά σε αυτή τη φάση, όπου παράγεται και ο κώδικας του συστήματος. Κάθε λειτουργική μονάδα πηγαίου κώδικα δοκιμάζεται για να διασφαλιστεί ότι κάθε ενότητα πληροί τις προδιαγραφές και τους χαρακτηρισμούς της, και επιτυγχάνει αυτό που υποτίθεται ότι πρέπει να κάνει προτού ενσωματωθούν και δοκιμαστούν ως ολόκληρο σύστημα.
- **Υποστήριξη:** Αυτή είναι η τελευταία φάση της διαδικασίας και περιλαμβάνει τυχόν μελλοντικές ενημερώσεις ή επέκταση του συστήματος. Επίσης, σε αυτή την τελική φάση ο πελάτης ή ο τελικός χρήστης αρχίζει να χρησιμοποιεί το σύστημα όπου μπορεί να αρχίζουν να εμφανίζονται προβλήματα που πρέπει να επιλυθούν σε αυτό το στάδιο. Πρέπει να σημειωθεί πως σε αυτή τη φάση μπορεί να περιλαμβάνεται επανάληψη των προηγούμενων φάσεων.

Τα επιπλέον στάδια που αναφέρονται στη βιβλιογραφία είναι (Abdelrafe , Burairah, & Norhaziah, 2016), (Hossain, 2023):

- **Δοκιμές:** Αυτό το στάδιο έρχεται μετά τη φάση της υλοποίησης. Όπου ελέγχεται ο κώδικας για να επιβεβαιωθεί ότι το προϊόν πληροί τις απαιτήσεις που συγκεντρώθηκαν στην φάση της Ανάλυσης. Παρατηρείται ότι σε κάποια βιβλιογραφικά άρθρα αυτό το στάδιο ενσωματώνεται σε εκείνο της Υλοποίησης, όμως στην πραγματικότητα αποτελεί ξεχωριστή φάση καθώς κατά τις δοκιμές η ομάδα έχει τη δυνατότητα να διορθώσει τυχόν λάθη πριν περάσει στο επόμενο στάδιο, μπορεί να γίνει δοκιμή αποδοχής χρήστη (UAT), ενώ το σύστημα μπορεί να δοκιμαστεί πριν το έργο παραδοθεί στον πελάτη.
- **Ανάπτυξη/Ενσωμάτωση:** Αυτή η φάση ακολουθεί τη φάση των δοκιμών και αποτελεί το προτελευταίο στάδιο, πριν, δηλαδή, την Υποστήριξη. Το περιβάλλον παραγωγής προετοιμάζεται να υποδεχτεί το νέο έργο, το οποίο ενσωματώνεται στο ήδη υπάρχον σύστημα, γίνεται η τελική δοκιμή και επικύρωση και, τέλος, διατίθεται στους τελικούς χρήστες ή πελάτες.

2.2.4 ΤΕΧΝΙΚΕΣ ΔΙΑΧΕΙΡΙΣΗΣ ΕΡΓΩΝ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ

Η αποτελεσματική διαχείριση έργων ανάπτυξης λογισμικού είναι ζωτικής σημασίας για την παράδοση επιτυχημένων έργων έγκαιρα και εντός του προϋπολογισμού. Για τον έλεγχο της διαδικασίας ανάπτυξης λογισμικού, μπορούν να χρησιμοποιηθούν διάφορες τεχνικές και μεθοδολογίες για τον εξορθολογισμό της διαδικασίας ανάπτυξης που αποτελούν ένα πλαίσιο που περιλαμβάνει τεχνικές, διαδικασίες και εργαλεία που βοηθούν τους προγραμματιστές στην εργασία τους (Arafeh & El-Ahmad, 2017).

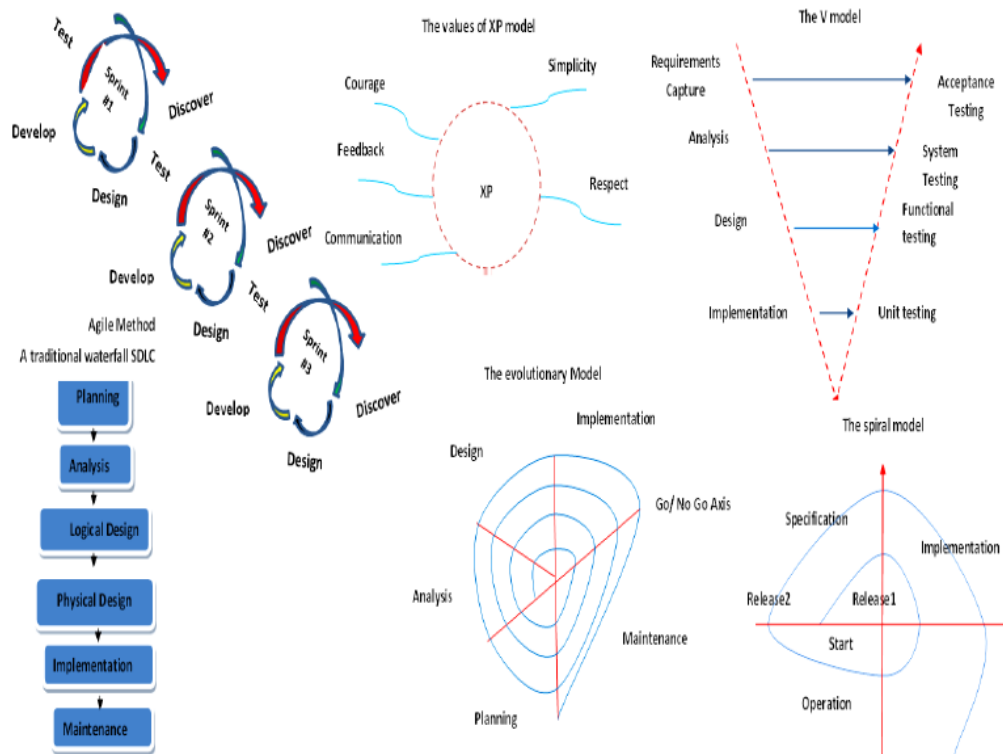
Ακολουθούν αναλυτικά ορισμένες δημοφιλείς τεχνικές ή μοντέλα διαχείρισης έργων ανάπτυξης λογισμικού (Arafeh & El-Ahmad, 2017), (Abdelrafe , Burairah, & Norhaziah, 2016), (Βάκκας, 2020), (Sujit Kumar & Pushkar, 2013):

- **Το μοντέλο κύκλου ζωής καταρράκτη (Waterfall):** είναι το κλασικό μοντέλο στη διαδικασία ανάπτυξης λογισμικού και αναφέρεται ως μια γραμμική διαδικασία σχεδιασμού. Ορίζει ορισμένες βασικές εργασίες, οι οποίες εκτελούνται σε ακολουθία και η διαδικασία δεν μπορεί να προχωρήσει στην επόμενη φάση εάν δεν έχει ολοκληρωθεί η τρέχουσα φάση. Ξεκινά με τη δημιουργία προδιαγραφών του συστήματος και συνεχίζει με σχεδιασμό, υλοποίηση, δοκιμή και συντήρηση. Το μοντέλο αυτό χρησιμοποιείται όταν οι απαιτήσεις του έργου είναι σαφείς και όταν οι αλλαγές είναι σταθερές.
- **Το σπειροειδές μοντέλο (Spiral):** Το σπειροειδές μοντέλο έχει τέσσερις φάσεις: Σχεδιασμός, Ανάλυση Κινδύνων, Ανάπτυξη και Αξιολόγηση. Ένα έργο λογισμικού διέρχεται επανειλημμένα από αυτές τις φάσεις σε επαναλήψεις (καλούνται σπείρες σε αυτό το μοντέλο). Η βασική σπείρα, ξεκινά από τη φάση σχεδιασμού, όπου συγκεντρώνονται οι απαιτήσεις και αξιολογείται ο κίνδυνος. Κάθε επόμενη σπείρα βασίζεται στη βασική σπείρα. Στη φάση της ανάλυσης

κινδύνου, πραγματοποιείται μια διαδικασία εντοπισμού κινδύνου και εναλλακτικών λύσεων. Ένα πρωτότυπο παράγεται στο τέλος της φάσης ανάλυσης κινδύνου. Το λογισμικό παράγεται στη φάση της ανάπτυξης, μαζί με τη δοκιμή στο τέλος της φάσης. Η φάση αξιολόγησης επιτρέπει στον πελάτη να αξιολογήσει την παραγωγή του έργου πριν το έργο συνεχίσει στην επόμενη σπείρα. Αυτή η προσέγγιση είναι κατάλληλη για έργα στα οποία οι αλλαγές μπορούν να ενσωματωθούν με λογικές αυξήσεις κόστους ή με αποδεκτές χρονικές καθυστερήσεις. Το σπειροειδές μοντέλο είναι ένας συνδυασμός μοντέλου πρωτοτύπου και καταρράκτη που εστιάζει στην ανάλυση κινδύνων.

- **V-model (Verification and Validation model):** Όπως το μοντέλο καταρράκτη, έτσι και αυτό είναι διαδοχικό μοντέλο και κάθε φάση πρέπει να ολοκληρωθεί πριν ξεκινήσει η επόμενη φάση. Κάθε φάση συνδέεται απευθείας με μια φάση επικύρωσης (φάση δοκιμής). Αυτό το μοντέλο είναι κατάλληλο όταν δεν υπάρχουν ασαφείς ή διφορούμενες απαιτήσεις και όταν το έργο είναι σύντομο.
- **Το μοντέλο κύκλου ζωής ταχείας ανάπτυξης εφαρμογών (Rapid application development RAD):** χρησιμοποιεί μια προσέγγιση στην οποία οι προγραμματιστές συνεργάζονται με ένα εξελισσόμενο πρωτότυπο. Αυτό το μοντέλο κύκλου ζωής απαιτεί επίσης μεγάλη συμμετοχή των χρηστών και βοηθά στην γρήγορη παραγωγή συστημάτων χωρίς να θυσιάζεται η ποιότητα. Οι προγραμματιστές χρησιμοποιούν εργαλεία RAD, όπως μηχανή λογισμικού με τη βοήθεια υπολογιστή (computer assisted software engineering - CASE), σχεδιασμό κοινών απαιτήσεων (joint requirements planning - JRP) και κοινό σχεδιασμό εφαρμογών (joint application development - JAD) για τη διευκόλυνση του γρήγορου πρωτοτύπου και της δημιουργίας κώδικα.
- **Το μοντέλο κύκλου ζωής πρωτοτύπου:** χρησιμοποιείται για την ανάπτυξη πρωτότυπων λογισμικού για τη διευκρίνιση των απαιτήσεων των χρηστών για ένα επιχειρησιακό λογισμικό. Απαιτεί μεγάλη συμμετοχή των χρηστών και οι προγραμματιστές χρησιμοποιούν ένα μοντέλο για να παράγουν ταυτόχρονα λειτουργικές απαιτήσεις και φυσικές προδιαγραφές σχεδιασμού. Αυτή η προσέγγιση χρησιμοποιείται συχνά σε συστήματα που περιλαμβάνουν μεγάλο βαθμό σχεδιασμού διεπαφής χρήστη, όπως έργα ιστοτόπου, σε συστήματα που αυτοματοποιούν προηγούμενες χειροκίνητες λειτουργίες ή σε συστήματα που αλλάζουν τη φύση του τρόπου με τον οποίο γίνεται κάτι, όπως οι κινητές εφαρμογές.
- **Το μοντέλο κύκλου ζωής της σταδιακής παράδοσης (incremental build):** προβλέπει την προοδευτική ανάπτυξη λειτουργικού λογισμικού, με κάθε έκδοση να παρέχει πρόσθετες δυνατότητες. Αυτή η προσέγγιση βοηθά να οργανωθούν τα χαρακτηριστικά και οι λειτουργίες των λογισμικών με βάση τις προτεραιότητες χρήστη ή το κόστος, το χρόνο, και το εύρος των αναθεωρήσεων.
- **Ευέλικτη ανάπτυξη λογισμικού (Agile Software Development):** Η ευέλικτη μέθοδος είναι μια ένωση επαναληπτικού και μοντέλου σταδιακής παράδοσης, το οποίο χωρίζει τη διαδικασία ανάπτυξης σε μικρά επαναληπτικά και επαυξητικά μπλοκ. Το προϊόν ελέγχεται συνεχώς, μέσω επαναλήψεων παράδοσης, μετριάζοντας τον κίνδυνο οποιασδήποτε αστοχίας στο μέλλον. Το κλειδί για ένα επιτυχημένο έργο λογισμικού είναι η επικοινωνία, η ευελιξία και η

καλή ανάλυση. Αυτός είναι ο λόγος για τον οποίο υιοθετείται η ευέλικτη προσέγγιση για την ανάπτυξη λογισμικού. Σημαίνει την ευελιξία προσαρμογής στις αλλαγές. Η μεθοδολογία ευέλικτης ανάπτυξης είναι ένα εννοιολογικό πλαίσιο για την ανάληψη οποιωνδήποτε έργων ανάπτυξης λογισμικού και είναι αυτό που χρησιμοποιείται ευρέως στη σημερινή εποχή. Υπάρχει ένας αριθμός ευέλικτων μεθόδων ανάπτυξης λογισμικού, αλλά οι πιο δημοφιλείς ευέλικτες μέθοδοι είναι ο Extreme Programming (XP), οι μέθοδοι κρυστάλλου, το Scrum και το Feature Driven Development.



Εικόνα 4: Παραδείγματα μοντέλων διαχείρισης έργων λογισμικού (Abdelrafe , Burairah, & Norhaziah, 2016)

Είναι σημαντικό να σημειωθεί ότι η επιλογή της μεθοδολογίας ή του συνδυασμού τεχνικών εξαρτάται από τις συγκεκριμένες απαιτήσεις του έργου, τη δυναμική της ομάδας και την κουλτούρα κάθε επιχείρησης. Πολλά σύγχρονα έργα ενσωματώνουν στοιχεία από πολλαπλές μεθοδολογίες σε μια υβριδική προσέγγιση για να ανταποκρίνονται στις μοναδικές ανάγκες τους.

ΚΕΦΑΛΑΙΟ 3 – ΔΙΑΧΕΙΡΙΣΗ ΚΙΝΔΥΝΟΥ

3.1 Η ΕΝΝΟΙΑ ΤΗΣ ΔΙΑΧΕΙΡΙΣΗΣ ΚΙΝΔΥΝΟΥ

Ως κίνδυνος ορίζεται από το διεθνές πρότυπο διαχείρισης κινδύνου ISO 31000, το αποτέλεσμα της αβεβαιότητας σε σχέση με την επίτευξη στόχων. Αποτελεί, δηλαδή, ένα ανεπιθύμητο αποτέλεσμα κατά την εκτέλεση του έργου το οποίο θα έχει αρνητικό αντίκτυπο στην επίτευξη του στόχου. Παρακάτω θα δούμε πως δεν είναι πάντοτε, όμως, αρνητικός. Οι κίνδυνοι είναι πολλοί και πολύπλευροι ενώ διαφέρουν ανάλογα με τη φύση του έργου, γι' αυτό και είναι δύσκολο να εκτιμηθούν όλες οι πτυχές τους ή να ελεγχθούν όλες οι συνέπειές τους. Η έννοια του κινδύνου αναφέρεται στην εμφάνιση απρόβλεπτων γεγονότων που θα επιφέρουν θετικές ή αρνητικές συνέπειες κατά τη διάρκεια του έργου (Σιώκου, 2021) (PMBOK, 2008).

Ως εκ τούτου, η διαχείριση κινδύνου μπορεί να ορισθεί ως η διαδικασία αναγνώρισης, ανάλυσης και αποδοχής ή μείωσης της αβεβαιότητας στη λήψη αποφάσεων (Κολιοφώτης, 2017) . Καθώς δε μπορούμε να αποφύγουμε κάθε είδους κίνδυνο που μπορεί να προκύψει, η διαχείριση κινδύνου στοχεύει στη μείωσή του και στη δημιουργία ενός σχεδίου αντιμετώπισης που αποτελεί τρόπο ελέγχου των κινδύνων (Φιτσιλής, 2015) (Σιώκου, 2021) μειώνοντας, έτσι, τα ποσοστά αβεβαιότητας και, συνεπώς, και τα ποσοστά αποτυχίας του έργου (Κολιοφώτης, 2017). Ακόμη, σκοπός της διαχείρισης έργου είναι να μεγιστοποιήσει τον αντίκτυπο που έχουν τα θετικά γεγονότα και να ελαχιστοποιήσει τον αντίκτυπο των αρνητικών (Μαρκάκη, 2021).

Τέλος, για τη διαχείριση κινδύνου υπάρχουν πολυάριθμες μέθοδοι και τεχνικές και επιλέγονται ανάλογα τη φύση του έργου. Οι πιο γνωστές που μπορεί να κανείς να συναντήσει είναι οι: Ανάλυση SWOT, δένδρική ανάλυση σφαλμάτων (Fault Tree Analysis), Ανάλυση BPEST (Business, Political, Economic, Social, Technological), Ανάλυση PESTLE (Political, Economic, Social, Technical, Legal, Environmental), Ανάλυση Απειλών (Threat Analysis) κ.ά. (Κολιοφώτης, 2017)

3.2 ΤΥΠΟΙ ΚΙΝΔΥΝΩΝ

Οι κίνδυνοι χωρίζονται με βάση τη φύση και την προέλευσή τους. Όσο αναφορά τη φύση τους, διακρίνονται σε ευκαιρίες και απειλές ενώ όσο αναφορά την προέλευσή τους διακρίνονται σε εσωτερικούς και εξωτερικούς. Για το πρώτο κριτήριο θα γίνει αναφορά σε επόμενο κεφάλαιο (3.5 Ο ΚΙΝΔΥΝΟΣ ΩΣ ΕΥΚΑΙΡΙΑ). Για το δεύτερο

κριτήριο, οι εσωτερικοί κίνδυνοι θεωρείται ότι συμβαίνουν εντός της επιχείρησης ή του οργανισμού, επομένως, επηρεάζεται άμεσα από τις ενέργειες της επιχείρησης, ενώ η πιθανότητα εμφάνισης των εξωτερικών κινδύνων δεν μπορεί να επηρεασθεί από κάποια ενέργεια της επιχείρησης (Νταρζάνου, 2012).

Ευκαιρίες	Αδιάφορο (:)	Πολύ επιθυμητό
Απειλές	Ανεπιθύμητο	Πρόβλημα (:)
	Εξωτερικοί	Εσωτερικοί

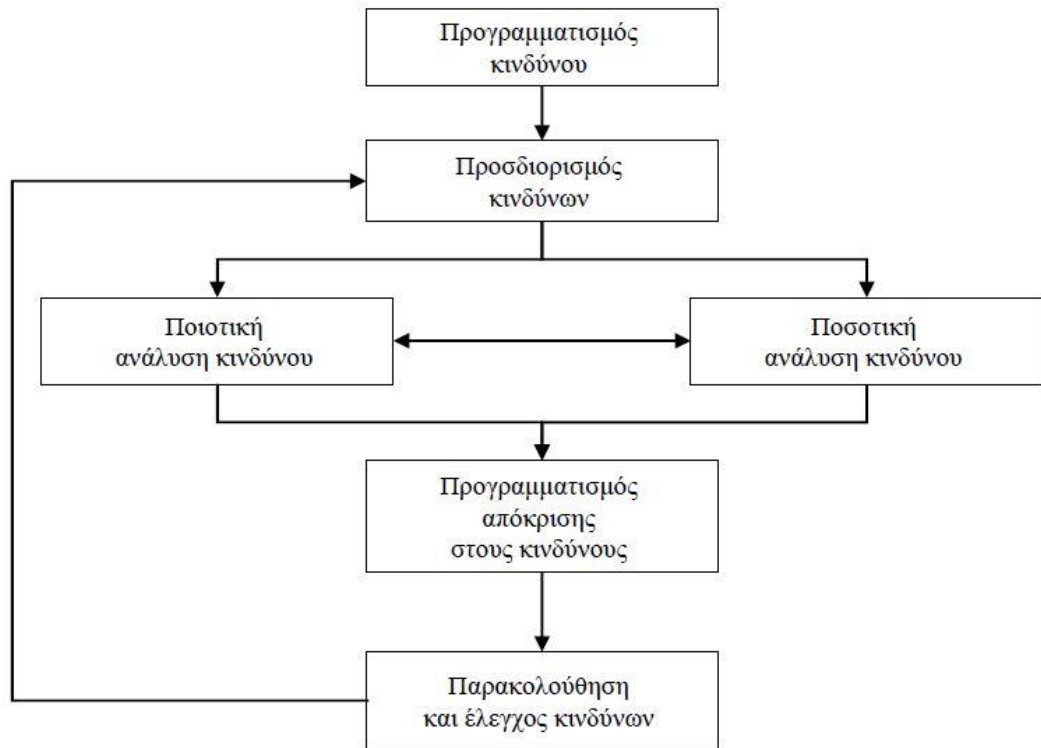
Εικόνα 5: Τύποι κινδύνων (Βίτσιος, 2018)

Στις εσωτερικές πηγές προέλευσης μπορεί να περιλαμβάνεται η διάρκεια του έργου, ο χρονοπρογραμματισμός, το κόστος, η ομάδα ανάπτυξης του έργου καθώς επίσης και η διοίκηση της επιχείρησης ή η εταιρική κουλτούρα. Για παράδειγμα, κακές εταιρικές διαχειρίσεις ή επεμβάσεις της διοίκησης, αλλά και συγκρούσεις μεταξύ της ομάδας ή η έλλειψη εμπειρίας, μπορούν να θεωρηθούν απειλές και να δημιουργήσουν κινδύνους. Αντίστοιχα, στις εξωτερικές πηγές μπορεί να περιλαμβάνονται οι πελάτες, η τεχνολογία, οι εξωτερικοί συνεργάτες ή η παρούσα κατάσταση της αγοράς. Για παράδειγμα, η αλλαγή των απαιτήσεων του έργου από την πλευρά του πελάτη, ή οι ραγδαίες αλλαγές της τεχνολογίας, αλλά και προβλήματα που μπορεί να αντιμετωπίζουν εξωτερικοί συνεργάτες, θεωρούνται επιβλαβείς παράγοντες για το έργο.

3.3 Η ΔΙΑΔΙΚΑΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΚΙΝΔΥΝΟΥ

Η διαχείριση κινδύνου βασίζεται στη θεωρία πιθανοτήτων, το να εντοπιστεί, δηλαδή, η πηγή που θα επιφέρει προβλήματα στην ομαλή εκτέλεση του έργου και να εκτιμηθεί η σοβαρότητα της περίπτωσης εμφάνισής του, ή αλλιώς η επικινδυνότητα του κινδύνου, ώστε να σχεδιαστεί ένα πλάνο ελέγχου που θα μειώσει την αρνητική έκβαση σε όλο το έργο αλλά και στην επιχείρηση. Ακόμη, συνδέεται άμεσα και με τις συνέπειες, κατά πόσο αυτό το πρόβλημα θα επηρεάσει το έργο. Για να συμβεί αυτή η εκτίμηση και να σχεδιαστεί το πλάνο, έχουν καθορισθεί έξι διαδικασίες από τον PMBOK (PMBOK, 2008), τις οποίες ακολουθεί η διαχείριση κινδύνου. Οι διαδικασίες αυτές είναι:

1. σχεδιασμός διαχείρισης κινδύνου (Risk Management Planning)
2. προσδιορισμός κινδύνων (Risk identification)
3. ποιοτική ανάλυση κινδύνου (Qualitative Risk Analysis)
4. ποσοτική ανάλυση κινδύνου (Quantitative Risk Analysis)
5. σχεδιασμός απόκρισης στους κινδύνους (Risk Response Planning)
6. παρακολούθηση και έλεγχος κινδύνων (Risk Monitoring and Control)



Εικόνα 6: Διαδικασία διαχείρισης κινδύνου (Φιτσιλής, 2015)

Στο αρχικό το στάδιο του **σχεδιασμού**, δημιουργείται η στρατηγική η οποία θα εφαρμοστεί για να εκτελεσθούν ορθά οι διαδικασίες διαχείρισης κινδύνου καθώς και ο τρόπος εφαρμογής της. Το σχέδιο διαχείρισης κινδύνου περιέχει τις μεθοδολογίες, την κατηγοριοποίηση των κινδύνων, τις αναθέσεις ρόλων και ευθυνών, τον χρόνο εφαρμογής, τον προϋπολογισμό, τις μεθόδους ποιοτικής και ποσοτικής ανάλυσης που θα εφαρμοσθούν καθώς επίσης και το πλάνο για το πως θα γίνουν οι αναφορές για τους επερχόμενους κινδύνους. (Σιώκου, 2021)

Ένα από τα βασικά εργαλεία που χρησιμοποιούνται είναι η δομή διαχείρισης κινδύνου (Risk Management Planning Breakdown Structure – RMP-BS). Η δομή διαχείρισης κινδύνου αποτελείται από βασικά στοιχεία όπως τους συμμετέχοντες που επηρεάζουν το έργο και τη διαχείριση κινδύνου (πελάτες, διοίκηση, κοινωνία, επενδυτές), το περιβάλλον του έργου (η αγορά, οι επικρατούντες κίνδυνοι, η νοοτροπία της διοίκησης κτλ), την οργάνωση του έργου (για παράδειγμα η πολυπλοκότητα που προκύπτει από

την τεχνολογία, τη νομοθεσία, το μέγεθος του έργου κ.ά.) και, τέλος, το πλαίσιο του έργου (οι στόχοι, οι περιορισμοί κτλ) (Φιτσιλής, 2015).

Ακολουθώς, κατά την **αναγνώριση** των κινδύνων, εξετάζονται και προσδιορίζονται ποιοι κίνδυνοι επηρεάζουν το έργο, καταγράφονται τα χαρακτηριστικά και οι πηγές τους όπως επίσης αξιολογείται και το αν θα επηρεάσουν θετικά ή αρνητικά το έργο. Εξετάζονται όλα τα πιθανά σενάρια που θα μπορούσαν να επηρεάσουν το έργο και ταξινομούνται με βάση τον παράγοντα και το αντικείμενο που θα επηρεασθεί (για παράδειγμα κίνδυνοι τεχνολογίας, περιβαλλοντικοί, νομοθετικοί, κίνδυνοι οργάνωσης έργου, επιχειρησιακοί κτλ). Ο προσδιορισμός των κινδύνων είναι μια επαναλαμβανόμενη διαδικασία καθώς όσο το έργο εξελίσσεται μπορεί να προκύψουν νέοι κίνδυνοι που μπορεί να επηρεάσουν το συνολικό έργο (Σιώκου, 2021) (Μαρκάκη, 2021).

Η **ποιοτική** και η **ποσοτική** ανάλυση αποτελούν την **εκτίμηση του κινδύνου** (Risk estimation). Κατά την εκτίμηση προσδιορίζεται η πιθανότητα που έχει κάθε κίνδυνος να συμβεί και η πιθανή επίδραση που μπορεί να έχει στο έργο. (Κολιοφώτης, 2017) Στην ποιοτική ανάλυση οι κίνδυνοι ταξινομούνται σε πίνακες ανάλογα με την σπουδαιότητά τους κατά προτεραιότητα (χαμηλή ή υψηλή) ή αλλιώς ανάλογα με τις συνέπειες που επιφέρει στο έργο και την πιθανότητα εμφάνισης. Η επικινδυνότητα υπολογίζεται ως το γινόμενο της πιθανότητας του κινδύνου να συμβεί με τη συνέπεια που προκαλεί (Φιτσιλής, 2015).

ΕΚΘΕΣΗ ΈΡΓΟΥ ΣΕ ΚΙΝΔΥΝΟ (κατά PMI)					
ΠΙΘΑΝΟΤΗΤΑ ΕΩΣ	ΕΚΘΕΣΗ = Πιθανότητα x Συνέπεια				
0,9 Πολύ υψηλή	0,05(M)	0,09(M)	0,18(Y)	0,36(Y)	0,72(Y)
0,7 Υψηλή	0,04(X)	0,07(M)	0,14(M)	0,28(Y)	0,56(Y)
0,5 Μέση	0,03(X)	0,05(M)	0,1(M)	0,2(Y)	0,4(Y)
0,3 Χαμηλή	0,02(X)	0,03(X)	0,06(M)	0,12(M)	0,24(Y)
0,1 Πολύ Χαμηλή	0,01(X)	0,01(X)	0,02(X)	0,04(X)	0,08(M)
	Πολύ Χαμηλή 0,05	Χαμηλή 0,1	Μέση 0,20	Υψηλή 0,40	Πολύ υψηλή 0,80
	ΣΥΝΕΠΕΙΑ				
Υ: Υψηλός Κίνδυνος – μη αποδεκτός, χρειάζεται άμεση αντίδραση Μ: Μέσος Κίνδυνος, μπορεί να χρειάζεται αντίδραση Χ: Χαμηλός Κίνδυνος, απλή παρακολούθηση					

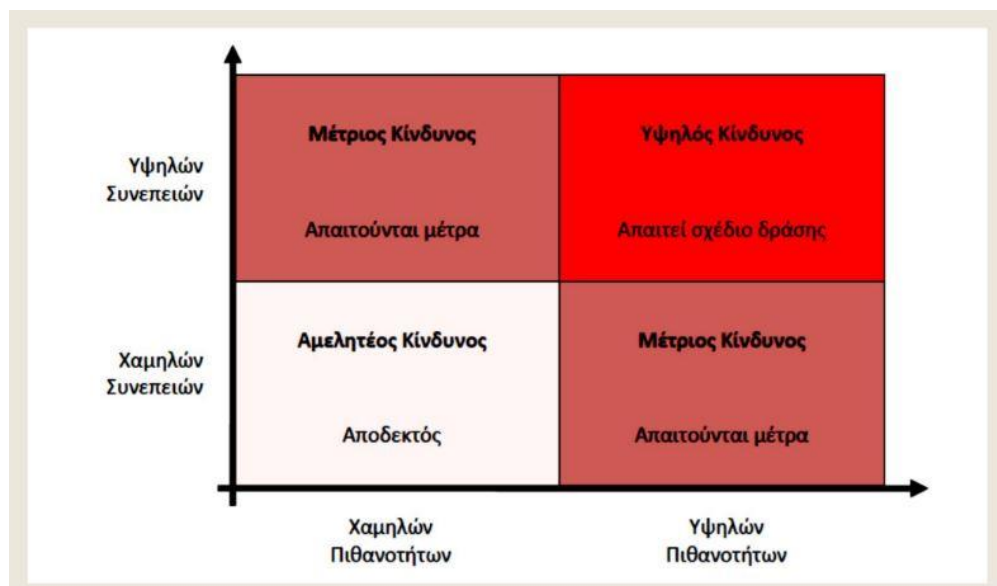
Εικόνα 7: Πίνακας κινδύνου (Μαρκάκη, 2021)

Αντίστοιχα, η ποσοτική ανάλυση ποσοτικοποιεί τη συνολική έκθεση του έργου στον κίνδυνο με σκοπό τον προσδιορισμό της πιθανότητας της επιτυχίας του έργου, τον προσδιορισμό του κόστους και του χρόνου της αντιμετώπισής του κινδύνου, να εντοπίσει κινδύνους που χρήζουν ιδιαίτερης προσοχής αλλά και να συμβάλει στον καθορισμό ρεαλιστικών στόχων σχετικά με το χρονοδιάγραμμα και τον προϋπολογισμό (Φιτσιλής, 2015). Για την ποσοτική ανάλυση των κινδύνων γίνεται χρήση τεχνικών όπως η Monte Carlo, η PERT, τα δέντρα αποφάσεων κτλ.

Έτσι, η ποσοτική ανάλυση εκτιμά τον συνολικό κίνδυνο του έργου και τις αβεβαιότητες και ξεκινά ο σχεδιασμός απόκρισης σε αυτόν. Κατά τον σχεδιασμό απόκρισης δημιουργείται ένα πλάνο στρατηγικών και δράσεων που είναι απαραίτητο να λάβουν χώρα πριν ή όταν εμφανιστεί ο κίνδυνος με σκοπό να μεγιστοποιηθούν τα οφέλη σε πιθανή εμφάνισή του και να ελαχιστοποιηθούν οι απειλές (Σιώκου, 2021).

Οι γενικές στρατηγικές αντίδρασης στον κίνδυνο σύμφωνα με τον (PMBOK, 2008) είναι:

- **Αποφυγή κινδύνου**, η οποία έχει στόχο την ολοκληρωτική εξάλειψη του κινδύνου
- **Μεταφορά κινδύνου**, η οποία μεταφέρει τον κίνδυνο σε κάποιο τρίτο έτσι ώστε να αποποιηθεί τις ευθύνες η επιχείρηση ή ο οργανισμός που έχει αναλάβει το έργο
- **Μετριασμός κινδύνου**, ή αλλιώς μείωση της διακινδύνευσης
- **Αποδοχή κινδύνου**, όπου γίνεται αποδεκτός ο κίνδυνος και οι συνέπειές του εφόσον δεν μπορεί να εφαρμοστεί κάποια από τις παραπάνω στρατηγικές
- **Δημιουργία ευκαιρίας**, η εκμετάλλευση ενός κινδύνου ως ευκαιρίας (αρκεί ο κίνδυνος να μην είναι καταστροφικός)



Εικόνα 8: Σχέδιο απόκρισης σε κίνδυνο (Μαρκάκη, 2021)

Η τελική διαδικασία στη διαχείριση κινδύνου είναι ο έλεγχος και η παρακολούθησή του. Κατά τον έλεγχο κοινοποιείται το σχέδιο διαχείρισης κινδύνων σε όλα τα ενδιαφερόμενα μέρη του έργου και ακολουθείται από εκπαίδευση και δοκιμές, κάτι που προσδίδει μια συνολική κουλτούρα διαχείρισης στην επιχείρηση ή τον οργανισμό. Κατά τον έλεγχο παρακολουθούνται τα πιθανά γεγονότα κινδύνου ώστε να είναι δυνατή η πρόβλεψη ή η αναγνώρισή τους. Επίσης, οι κίνδυνοι που μπορεί να προκύψουν είναι απαραίτητο να αντιμετωπίζονται και να παρακολουθούνται μέχρι να επιλυθούν ενώ οι παρεκκλίσεις πρέπει να αναγνωρίζονται και να αναφέρονται στους διαχειριστές του έργου και να υλοποιούνται οι συμφωνημένες δράσεις απόκρισης (Μαρκάκη, 2021).

Η παρακολούθηση εξετάζει την αποτελεσματικότητα των μέτρων που έχουν σχεδιαστεί σε προηγούμενη διαδικασία ενώ, παράλληλα, εποπτεύει την ορθή τήρηση των διαδικασιών. Ακόμη, η ενσωμάτωση διαφόρων αλλαγών και τρόπων αντιμετώπισης είναι σημαντικό κομμάτι στα πλαίσια της παρακολούθησης (Κολιοφώτης, 2017). Είναι απαραίτητο, για να είναι αποτελεσματική η διαδικασία της παρακολούθησης, να είναι εμφανής σε όλα τα ενδιαφερόμενα μέρη του έργου και, συνεπώς, σε όλα τα τμήματα της επιχείρησης ή του οργανισμού.

Τέλος, οι διαδικασίες προσδιορισμού, ανάλυσης, αντιμετώπισης και παρακολούθησης των κινδύνων λαμβάνουν χώρα καθ' όλη τη διάρκεια της ζωής του έργου καθώς η μεταβολή των κινδύνων ή η εμφάνιση νέων μπορεί να διαφοροποιήσει το συνολικό κίνδυνο και, συνεπώς, την απόδοση του έργου. Η διαχείριση κινδύνου, έχει άμεση σύνδεση με την επίτευξη των στόχων για αυτό και είναι σημαντικό να συμπεριλαμβάνεται σε όλη τη διάρκεια ανάπτυξης και εκτέλεσης του έργου ώστε να ελέγχονται όσο γίνεται περισσότεροι κίνδυνοι που μπορεί να προκύπτουν σε όλη αυτή τη διαδικασία. (Νταρζάνου, 2012)

3.5 Ο ΚΙΝΔΥΝΟΣ ΩΣ ΕΥΚΑΙΡΙΑ

Όπως αναφέρθηκε παραπάνω, ένας κίνδυνος μπορεί να θεωρηθεί ως απειλή φέροντας αρνητικές συνέπειες ή ως ευκαιρία φέροντας καλοδεχούμενες επιρροές στο έργο ενώ μια στρατηγική για την απόκριση στον κίνδυνο είναι η δημιουργία ευκαιρίας. Για να αξιολογηθεί εάν ένας κίνδυνος μπορεί να επιφέρει θετικές ή αρνητικές συνέπειες, εκτελείται μια διαχείριση ευκαιριών, η οποία εκτιμά κατά πόσο ήταν επιβλαβής η εμφάνιση του κινδύνου στις δραστηριότητες του έργου.

Οι επιχειρήσεις, συνήθως, προσπαθούν είτε να αποφύγουν ένα κίνδυνο (downside- ο κίνδυνος ως απειλή) είτε να επωφεληθούν από αυτόν (upside - ο κίνδυνος ως ωφέλεια). Με τη διαχείριση του κινδύνου ως ευκαιρία προσπαθούν να εξασφαλίσουν προστιθέμενη αξία, συνεπώς ανταγωνιστικό πλεονέκτημα, άρα επιτυχία και κέρδος (Κολιοφώτης, 2017). Οι θετικοί κίνδυνοι αναφέρονται μερικές φορές ως "ευκαιρίες" επειδή αντιπροσωπεύουν καταστάσεις που, εάν πραγματοποιηθούν, μπορούν να οδηγήσουν σε ευεργετικά αποτελέσματα, όπως αύξηση εσόδων, βελτιωμένη απόδοση ή βελτιωμένη ικανοποίηση των ενδιαφερομένων μερών. Τα έργα και τα νέα εγχειρήματα συνήθως σχεδιάζονται έτσι ώστε να εκμεταλλεύονται αυτές τις ευκαιρίες -

να κατασκευαστεί κάτι καινούργιο, να εμπλουτιστεί μια υπάρχουσα εγκατάσταση ή να υπάρχει μια απάντηση στις ανάγκες της αγοράς (Μαρκάκη, 2021).

Συγκεκριμένα, τα βασικά χαρακτηριστικά των θετικών κινδύνων (ευκαιριών) περιλαμβάνουν:

- Δυνατότητα κέρδους: Οι θετικοί κίνδυνοι παρουσιάζουν τη δυνατότητα κέρδους ή οφέλους για το έργο ή τον οργανισμό. Αντιπροσωπεύουν ευκαιρίες για την επίτευξη των στόχων πιο αποτελεσματικά ή αποτελεσματικά.
- Βελτιωμένη απόδοση: Η συνειδητοποίηση θετικών κινδύνων μπορεί να οδηγήσει σε βελτιωμένη απόδοση του έργου, αυξημένη αποδοτικότητα ή επίτευξη στόχων του έργου πέρα από τις αρχικές προσδοκίες.
- Καινοτομία και δημιουργικότητα: Οι θετικοί κίνδυνοι συχνά περιλαμβάνουν καινοτόμες ιδέες ή δημιουργικές λύσεις που μπορούν να αξιοποιηθούν για να ενισχύσουν την επιτυχία του έργου.
- Στρατηγικά πλεονεκτήματα: Οι θετικοί κίνδυνοι μπορεί να παρέχουν στρατηγικά πλεονεκτήματα, όπως η ικανότητα εισόδου σε νέες αγορές, απόκτησης ανταγωνιστικού πλεονεκτήματος ή ενίσχυσης των σχέσεων με τα ενδιαφερόμενα μέρη.
- Προληπτική διαχείριση: Η διαχείριση θετικών κινδύνων περιλαμβάνει προληπτικό σχεδιασμό και εφαρμογή στρατηγικών για την εκμετάλλευση αυτών των ευκαιριών όταν προκύψουν.

Για να εντοπισθεί εάν ένας κίνδυνος μπορεί να αποτελέσει ευκαιρία, συνήθως χρησιμοποιείται η ανάλυση SWOT, κατά την οποία αναλύονται τα αδύναμα σημεία της επιχείρησης ή του οργανισμού και ταξινομούνται ως απειλές ή ευκαιρίες. Αξίζει να σημειωθεί, πως στην προσπάθεια ανάλυσης και διαχείρισης των απειλών και ευκαιριών, συνήθως θεωρείται πως θα πρέπει να υπάρχει ένα σχέδιο αντιμετώπισης ώστε να διαχωρίζονται οι απειλές και οι ευκαιρίες. Αυτό συνήθως, όμως, είναι αδύνατο καθώς σπανίως θεωρούνται ανεξάρτητοι όροι (Νταρζάνου, 2012).

Συνοπτικά, οι θετικοί κίνδυνοι στη διαχείριση κινδύνων αντιπροσωπεύουν ευκαιρίες για θετικά αποτελέσματα και κέρδη. Ο προληπτικός εντοπισμός και η διαχείριση αυτών των ευκαιριών μπορεί να συμβάλει στη συνολική επιτυχία ενός έργου ή ενός οργανισμού. Η αναγνώριση των απειλών ως ευκαιριών στη διαχείριση κινδύνου περιλαμβάνει προληπτική νοοτροπία, στρατηγικό σχεδιασμό και δέσμευση για συνεχή βελτίωση. Αντιμετωπίζοντας τους κινδύνους κατά μέτωπο, οι ομάδες μπορούν όχι μόνο να μετριάσουν τις πιθανές αρνητικές επιπτώσεις αλλά και να αξιοποιήσουν ευκαιρίες για ανάπτυξη, καινοτομία και βελτίωση.

ΚΕΦΑΛΑΙΟ 4 – ΔΙΑΧΕΙΡΙΣΗ ΚΙΝΔΥΝΟΥ ΣΕ ΕΡΓΑ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ

4.1 ΕΙΣΑΓΩΓΗ

Στον τομέα της πληροφορικής η έννοια του κινδύνου συνδέεται άμεσα με σφάλματα ή προβλήματα που μπορεί να προκύψουν κατά την λειτουργία ενός πληροφοριακού συστήματος (πρόβλημα υλικού ή λογισμικού). Στα έργα πληροφορικής δίνεται ιδιαίτερη έμφαση στους κινδύνους που προκύπτουν από τον κώδικα του λογισμικού, τις συγχωνεύσεις εταιρειών (συνένωση των πληροφοριακών συστημάτων των εταιρειών) και τις συβάσεις για την συντήρηση των πληροφοριακών συστημάτων. (Παπαδημητρίου, 2009)

Στις εταιρείες πληροφορικής πολλές δραστηριότητες μπαίνουν σε φάση υλοποίησης, ακόμα και εάν οι εταιρείες γνωρίζουν ότι αυτές οι δραστηριότητες εμπεριέχουν υψηλό κίνδυνο. Πολλές φορές οι δραστηριότητες που εμπεριέχουν υψηλούς κινδύνους είναι σημαντικές για μια επιχείρηση, για να αποκτήσει η επιχείρηση στρατηγικό πλεονέκτημα έναντι των ανταγωνιστών της. (Παπαδημητρίου, 2009)

Η διαχείριση κινδύνων στην τεχνολογία λογισμικού, και κατ' επέκταση στα έργα πληροφορικής, χρησιμοποιεί μια πολύ προσεκτική προσέγγιση λαμβάνοντας όλα τα δυνατά προληπτικά μέτρα, έτσι ώστε να γίνει η ολοκλήρωση ενός έργου μέσα στον καθορισμένο χρόνο και προϋπολογισμό. Στην πραγματικότητα στα έργα που λαμβάνονται υπόψη οι κίνδυνοι το τελικό αποτέλεσμα είναι πολύ καλύτερο τόσο από πλευράς τελικού κόστους και χρόνου υλοποίησης όσο και από την πλευρά της ποιότητας των παραδοτέων. Χωρίς την διαχείριση κινδύνων θα υπήρχε μεγάλη πιθανότητα οι εταιρείες που υλοποιούν έργα να χάσουν τόσο έσοδα όσο και την φήμη τους στους πελάτες (όπως και συμβαίνει άλλωστε), ή ακόμα χειρότερα να οδηγηθούν σε ολοκληρωτική πτώχευση οι συμμετέχοντες επιχειρήσεις / οργανισμοί σε ένα έργο. (Παπαδημητρίου, 2009)

Η διαχείριση κινδύνου σε έργα λογισμικού έχει πολλές ομοιότητες με τη διαχείριση κινδύνου σε άλλους τύπους έργων. Παρ' όλα αυτά, λόγω των ιδιαίτερων χαρακτηριστικών που έχει ένα λογισμικό ως τελικό προϊόν, η διαχείριση κινδύνου έχει επίσης κάποια ιδιαίτερα χαρακτηριστικά στα έργα λογισμικού. Αυτά τα χαρακτηριστικά

αναφέρονται κυρίως σε κινδύνους των απαιτήσεων/ προδιαγραφών του συστήματος, κινδύνους τεχνολογίας και κινδύνους προσωπικού κ.ά (Enfei, 2015)

Η Διαχείριση Κινδύνου Λογισμικού αναφέρεται σε μια σειρά βημάτων των οποίων οι στόχοι είναι να εντοπιστούν, να αντιμετωπιστούν και να εξαλειφθούν τα στοιχεία κινδύνου λογισμικού προτού γίνουν είτε απειλές για την επιτυχή λειτουργία του λογισμικού είτε βασική πηγή δαπανηρής επανεξέτασης (Arafteh & El-Ahmad, 2017)

Στα έργα πληροφορικής συνήθως χρησιμοποιούνται οι παρακάτω μέθοδοι για τη διαχείριση του κίνδυνου (Παπαδημητρίου, 2009):

- Κύκλοι μάθησης
- Προβληματισμός (Brainstorming)
- Τεχνική Delphi
- Nominal Group Technique
- Ανάλυση SWOT
- Συνεντεύξεις
- Λίστες ελέγχου
- Διαγράμματα αιτιών και αποτελέσματος
- Εμπειρία- Προηγούμενα έργα

4.2 ΠΑΡΑΓΟΝΤΕΣ ΚΙΝΔΥΝΟΥ ΕΡΓΩΝ ΛΟΓΙΣΜΙΚΟΥ - ΒΙΒΛΙΟΓΡΑΦΙΑ

Η ανάπτυξη λογισμικού είναι από μόνη της μια επικίνδυνη διαδικασία. Ο κύκλος της ανάπτυξης λογισμικού είναι ευάλωτος σε κινδύνους από την έναρξη του έργου έως και την παράδοση του έργου ως προϊόν. Κάθε φάση του κύκλου ζωής είναι ευαίσθητη σε ποικίλες απειλές που μπορεί να εμποδίσουν τη σωστή ανάπτυξη και την ολοκλήρωση της διαδικασίας ώστε να επέλθει σε ένα πλήρες προϊόν και να επιτύχει το έργο. Για τη σωστή διαχείριση αυτών των κινδύνων, οφείλεται να υπάρχει σωστή κατανόηση των προβλημάτων της διαδικασίας, των κινδύνων καθώς και οι αιτίες τους. Ως εκ τούτου, το πρώτο βήμα για τη διαχείριση αυτών, είναι η αναγνώρισή τους. Αυτοί που οδηγούν σε αποτυχία του έργου, ονομάζονται Παράγοντες Κινδύνου Λογισμικού. (Hijazi, Algrainy, Muaidi, & Khmour, RISK FACTORS IN SOFTWARE DEVELOPMENT PHASES, January 2014)

Κατά τον (Menezes, Gusmão, & Moura, 2018), οι παράγοντες κινδύνου είναι η πηγή και η αιτία κάθε κινδύνου. Οι κίνδυνοι στα έργα ανάπτυξης λογισμικού μπορούν να χωριστούν σε εσωτερικούς, εντός της επιχείρησης ή του οργανισμού, και εξωτερικούς, εκτός της επιχείρησης ή του οργανισμού, κινδύνους. Οι εσωτερικοί συνήθως χωρίζονται σε κινδύνους που έχουν να κάνουν με το προϊόν όπως τεχνικοί, τη διαδικασία όπως το αποτέλεσμα των τεχνικών κινδύνων ή το έργο ως την τελική επίδοσή του (performance). (Sankhwar & Pandey, 2014). Συνεπώς, ο κίνδυνος είναι το άθροισμα ενός συνόλου παραγόντων κινδύνου που εάν πραγματοποιηθούν, αποτελούν κίνδυνο. Επομένως, οι παράγοντες κινδύνου συμβάλλουν στη μείωση του επιπέδου αφαίρεσης ενός κινδύνου. (Menezes, Gusmão, & Moura, 2018)

Καθώς οι κίνδυνοι σε ένα έργο ανάπτυξης λογισμικού παρουσιάζονται σε κάθε φάση της ανάπτυξης (Sankhwar & Pandey, 2014) , είναι σημαντικό να συνδυάζεται ο κύκλος της διαδικασίας ανάπτυξης λογισμικού (SDLC) με τη διαχείριση κινδύνου ως ποιοτική και ποσοτική τεχνική ώστε να βοηθήσει τον διαχειριστή του έργου λογισμικού να παρακολουθεί και να μετριάσει το έργο. Παρ' όλα αυτά, εάν το έργο είναι αρκετά πολύπλοκο ή αρκετά μεγάλο σε μέγεθος, η διαχείριση του κινδύνου γίνεται πολύ δύσκολη (Abdelrafe , Burairah, & Norhaziah, 2016)

Κάθε έργο ανάπτυξης λογισμικού έχει ρίσκα σε κάθε στάδιο του κύκλου ανάπτυξης λογισμικού (SDLC) (Abdelrafe , Burairah, & Norhaziah, 2016). Σύμφωνα με τη βιβλιογραφία, οι πιθανότεροι και σοβαρότεροι παράγοντες αναφέρονται στο στάδιο του Σχεδιασμού και της Ανάλυσης ή παράγοντες που παρατηρούνται σε όλα τα στάδια της ανάπτυξης όπως η έλλειψη της επικοινωνίας (μεταξύ ομάδων ή μεταξύ επιχείρησης και πελάτη) ή έλλειψη δεξιοτήτων της ομάδας ή του διαχειριστή του έργου κ.ά.

Πιο συγκεκριμένα, η ελλιπής ή συνεχής αλλαγή των απαιτήσεων καθώς και οι γνώσεις της ομάδας για το συγκεκριμένο έργο ή την τεχνολογία που πρέπει να ακολουθηθεί, μπορεί να αποτελούν μικρότερο κίνδυνο σύμφωνα με τους Habibah et. al (NOOR HABIBAH, AZLINAH, & ZAIHA, 2007) ενώ τον μεγαλύτερο κίνδυνο έχει η κακή επικοινωνία σε έναν οργανισμό. Ανεξάρτητα από το εάν το έργο ολοκληρώθηκε πληρώντας τις απαιτήσεις ή όχι, μπορεί να προχωρήσει, και αυτό δείχνει την έλλειψη επικοινωνίας μεταξύ των χρηστών και των διαχειριστών του έργου.

Η μεγαλύτερη συχνότητα κινδύνων κατά τη φάση της Ανάλυσης παρατηρείται και στην έρευνα των Menezes, Gusmão και Moura (Menezes, Gusmão, & Moura, 2018), σύμφωνα με την οποία η ασάφεια, οι αλλαγές, οι νέες τεχνολογίες, η πολυπλοκότητα και η ατελείς προδιαγραφές επισημαίνονται ως οι παράγοντες κινδύνου που χρίζουν μεγαλύτερης προσοχής στη διαχείριση έργων λογισμικού. Επιπλέον, ο πιο αναφερόμενος παράγοντας κινδύνου σχετίζεται με τις δεξιότητες της ομάδας για την ανάπτυξη των δραστηριοτήτων τους. Από την άλλη πλευρά, οι πιο σχετικοί παράγοντες κινδύνου διαχείρισης ήταν η κακή δέσμευση μεταξύ επιχείρησης και χρήστη/πελάτη, το ασταθές οργανωτικό περιβάλλον και ο μη προγραμματισμός ή ο ανεπαρκής σχεδιασμός. Οι δύο πρώτοι είναι εξωτερικοί παράγοντες ενός έργου που το επηρεάζουν άμεσα και ο τελευταίος έχει ισχυρή σχέση με την ποιότητα του σχεδιασμού του έργου, ειδικά όσον αφορά το εύρος.

Συγκεκριμένα, στην έρευνά τους χαρακτηρίζουν ως κορυφαίους παράγοντες κινδύνου τις ελλείψεις προσωπικού, την έλλειψη τεχνογνωσίας και προσόντων της ομάδας ανάπτυξης, ειδικότερα όταν τα μέλη της ομάδας δεν διαθέτουν εξειδικευμένες δεξιότητες που απαιτούνται από το έργο. Ακόμη, η ασάφεια των προδιαγραφών/απαιτήσεων, οι συχνές αλλαγές των απαιτήσεων, η εισαγωγή νέας τεχνολογίας, το ασταθές οργανωτικό περιβάλλον, η τεχνική πολυπλοκότητα, ο ανεπαρκής προγραμματισμός, και η ελλιπής ποιότητα προδιαγραφών αποτελούν εξίσου σοβαρούς κινδύνους για ένα έργο ανάπτυξης λογισμικού.

Το ίδιο συναντάται και στην έρευνα του Liu Enfei (Enfei, 2015), ο οποίος υποστηρίζει πως οι ανεπαρκείς αρμοδιότητες και η έλλειψη συνεργασίας των μελών της ομάδας, η έλλειψη αποτελεσματικής διαδικασίας/μεθοδολογίας ανάπτυξης καθώς και η μη

κατανόηση των απαιτήσεων/προδιαγραφών και η ανεπαρκής συμμετοχή των χρηστών, είναι οι κορυφαίοι παράγοντες κινδύνων. Ως παράδειγμα αναφέρει στην έρευνά του πως δεδομένου ότι οι πελάτες δεν μπορούν να οπτικοποιήσουν το λογισμικό πριν ολοκληρωθεί η ανάπτυξη, είναι πολύ δύσκολο να έχουν ακριβείς ορισμούς απαιτήσεων. Η δημιουργία ακριβών απαιτήσεων και η αποφυγή ορισμένων κινδύνων ταυτόχρονα είναι πολύ σημαντική. Θα μπορούσε να υπάρξει μεγάλος αριθμός τεχνικών προβλημάτων κατά την εφαρμογή λογισμικού και η ταχεία ανάπτυξη τεχνολογιών επιδεινώνει ακόμη περισσότερο την πιθανότητα εμφάνισης κινδύνων. Ακόμη, η εναλλαγή των εργαζομένων κατά την ανάπτυξη λογισμικού είναι αρκετά συνηθισμένη και μπορεί να επιφέρει σοβαρούς κινδύνους σε έργα λογισμικού.

Η ανεπαρκής συμμετοχή των χρηστών και οι ανεπαρκείς γνώσεις/δεξιότητες των μελών της ομάδας αναφέρονται ως παράγοντες και στην έρευνα των Abdelrafe , Burairah, και Norhazia (Abdelrafe , Burairah, & Norhazia, 2016) , τους οποίους παράγοντες κατατάσσουν στη φάση του προγραμματισμού και της υποστήριξης αντίστοιχα. Σε γενικές γραμμές, και εδώ οι περισσότεροι κίνδυνοι παρατηρούνται στο στάδιο της ανάλυσης, όπου θεωρούν τους πιο συχνούς παράγοντες να είναι οι ασαφείς, εσφαλμένες και συχνά μεταβαλλόμενες απαιτήσεις του έργου λογισμικού, οι ελλείψεις ή απώλεια λεπτομερειών στις προδιαγραφές του συστήματος, ο μη επαρκής προσδιορισμός απαιτήσεων του συστήματος του λογισμικού, η αλλαγή των απαιτήσεων μετά τη φάση του σχεδιασμού του έργου, ενώ παράλληλα θεωρούν πως η έλλειψη προγραμματισμού έργου αποτελεί και αυτή παράγοντα κινδύνου στη φάση προγραμματισμού.

Σε αντίθεση, ο Shahzad (Shahzad, 2012) αναγνωρίζοντας τους δέκα κορυφαίους παράγοντες κινδύνου στα έργα λογισμικού, θεωρεί πως η έλλειψη προγραμματισμού έργου δεν αποτελεί κίνδυνο, όπως επίσης και η έλλειψη δεξιοτήτων του διαχειριστή του έργου. Παρ' όλα αυτά συμφωνεί πως η έλλειψη δεξιοτήτων της ομάδας ανάπτυξης του έργου, η κακή δέσμευση μεταξύ επιχείρησης και χρήστη/πελάτη, οι συνεχείς μεταβολές και η μη κατανόηση των απαιτήσεων του συστήματος αποτελούν σοβαρό κίνδυνο, ενώ προσθέτει κάποιους παραπάνω παράγοντες όπως οι λανθασμένες εκτιμήσεις προϋπολογισμού και χρονοδιαγράμματος, η αδύναμη μεθοδολογία διαχείρισης έργου και η εναλλαγή προσωπικού.

Η εναλλαγή προσωπικού, η έλλειψη επικοινωνίας της ομάδας έργου, η έλλειψη τεχνογνωσίας και δεξιοτήτων της ομάδας έργου αλλά και η ανεπαρκής συμμετοχή των χρηστών παρατηρείται και στην έρευνα των (NOOR HABIBAH, AZLINAH, & ZAIHA, 2007), οι οποίοι προσθέτουν με τη σειρά τους τον παράγοντα της έλλειψης πόρων, της έλλειψης του σχεδιασμού του έργου κατά τη φάση του σχεδιασμού και τις μη ρεαλιστικές προσδοκίες κατά της φάση της ανάλυσης.

Στον αντίποδα, οι (Bilal, Bashir, & Gani, 2020) αναφέρουν παράγοντες κινδύνου και σε άλλα στάδια του κύκλου ζωής των έργων λογισμικού. Οι πιο συχνοί παράγοντες κατά τη φάση της Υλοποίησής φαίνεται να είναι η ανάπτυξη λανθασμένων λειτουργιών και ιδιοτήτων του λογισμικού, οι ελλείψεις του προσωπικού ανάπτυξης, η ανάπτυξη λανθασμένου περιβάλλοντος χρήστη και η ανάπτυξη στοιχείων του λογισμικού από την αρχή. Κατά τις φάσεις της Ανάλυσης και του Σχεδιασμού, αναγνώρισαν ως κύριους

παράγοντες τη χρήση μη ρεαλιστικών χρονοδιαγραμμάτων και προϋπολογισμών, τους μη ρεαλιστικούς στόχους, τον κακό ή ανεπαρκή σχεδιασμό, και την παράλειψη των μη λειτουργικών απαιτήσεων, τις συχνές αλλαγές, την εισαγωγή νέας τεχνολογίας, την απουσία ποιοτικών αρχιτεκτονικών και σχεδιαστικών εγγράφων καθώς επίσης και τις εκτενείς προδιαγραφές. Ακόμη, παρατηρείται πως αναγνώρισαν κύριους παράγοντες κινδύνου κατά το στάδιο των δοκιμών, όπως την έλλειψη πλήρους αυτοματοποιημένου συστήματος δοκιμών, την έλλειψη γνώσεων στην ομάδα δοκιμών, το υψηλό ποσοστό σφάλματος στα πρόσφατα σχεδιασμένα και υλοποιημένα μέρη του συστήματος.

Τέλος, κατά το στάδιο της υποστήριξης αναφέρουν ως παράγοντες κινδύνου την αποτυχία χρήσης μιας προσέγγισης σταδιακής παράδοσης, τις ελλείψεις απόδοσης σε πραγματικό χρόνο, την έλλειψη πόρων και εγκαταστάσεων αναφοράς, την ενοποίηση μεταξύ πολλών διαφορετικών τεχνολογιών αλλά και σε επίπεδο συμβάσεων αναφέρεται ότι οι αναντιστοιχίες διαδικασίας απόκτησης και σύναψης συμβάσεων μπορούν να αποτελέσουν εξίσου κίνδυνο στο έργο λογισμικού.

Συνοψίζοντας, παρατηρείται πως κίνδυνοι υπάρχουν σε όλες τις φάσεις της ανάπτυξης των έργων λογισμικού, κάποιοι περισσότερο σοβαροί που μπορεί να συμβάλουν στην αποτυχία του έργου. Από τη βιβλιογραφία αναγνωρίζουμε πως οι συχνότεροι κίνδυνοι αναφέρονται στη φάση της ανάλυσης και του σχεδιασμού, όπως για παράδειγμα ο μη προγραμματισμός ή ο ανεπαρκής σχεδιασμός, η ελλιπής ή συνεχής αλλαγή των απαιτήσεων, η χρήση μη ρεαλιστικών χρονοδιαγραμμάτων και προϋπολογισμών, η μη κατανόηση των απαιτήσεων/προδιαγραφών, ενώ μπορούμε να πούμε πως σοβαροί παράγοντες αναγνωρίζονται σε ότι αφορά όλες τις φάσεις, όπως η έλλειψη ή η εναλλαγές προσωπικού, η έλλειψη επικοινωνίας ή η ανεπαρκής συμμετοχή των χρηστών και οι ανεπαρκείς δεξιότητες των μελών της ομάδας.

4.3 ΔΙΑΧΕΙΡΙΣΗ ΚΙΝΔΥΝΟΥ ΣΕ ΕΡΓΑ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ

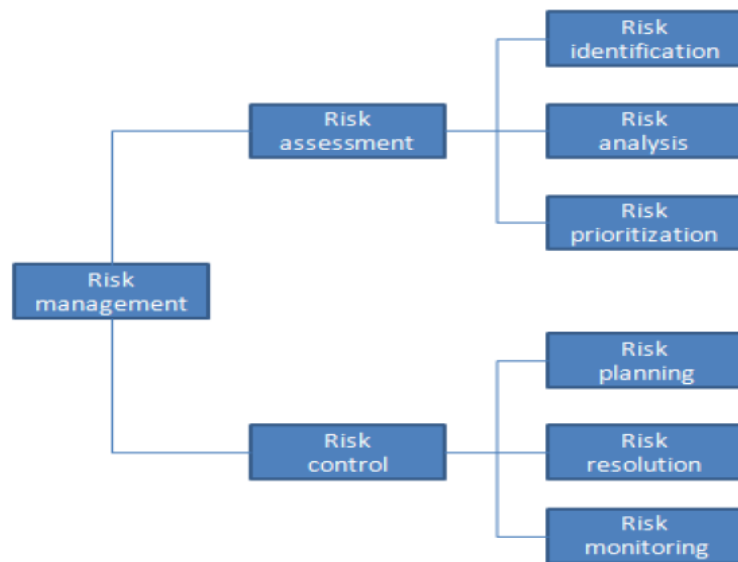
Η διαχείριση κινδύνου έργου λογισμικού ορίζεται συνήθως ως ένα σύνολο αρχών και πρακτικών που στοχεύουν στον εντοπισμό, την ανάλυση και τον χειρισμό των παραγόντων κινδύνου για την αύξηση των πιθανοτήτων να επιτευχθεί ένα επιτυχημένο αποτέλεσμα που θα αποτελέσει και ένα επιτυχημένο έργο ή/και την αποφυγή αποτυχίας του έργου (Enfei, 2015).

Στα έργα λογισμικού, ο κίνδυνος μπορεί να εμφανιστεί για πολλούς λόγους, π.χ. διαφορετικές απαιτήσεις, έλλειψη συμμετοχής των χρηστών, αποτυχίες συμβάσεων, έλλειψη προσωπικού, θέματα τεχνολογίας και περιβάλλοντος κ.λπ που αναφέρθηκαν στο παραπάνω υποκεφάλαιο. Τα έργα λογισμικού δεν μπορούν να απαλλαγούν πλήρως από τους κινδύνους, αλλά η πιθανότητα εμφάνισης κινδύνου μπορεί να μειωθεί μαζί με τον αντίκτυπό του χρησιμοποιώντας τις κατάλληλες τεχνικές διαχείρισης κινδύνου. Η διαχείριση κινδύνου είναι ένα σημαντικό συστατικό για την επιτυχία των έργων λογισμικού (Bilal, Bashir, & Gani, 2020) διότι η φύση αυτών των έργων αντιμετωπίζει ένα ευρύ φάσμα κινδύνων και όλοι οι κίνδυνοι δεν μπορούν να αντιμετωπιστούν στο ίδιο επίπεδο. Ως εκ τούτου, φαίνεται η ανάγκη για αξιολόγηση

κινδύνου για την επισήμανση και την ιεράρχηση των σοβαρών κινδύνων που αντιμετωπίζουν τα έργα λογισμικού.

Η ενσωμάτωση της επίσημης διαχείρισης κινδύνου με τη διαχείριση έργων είναι ένα νέο φαινόμενο στην κοινότητα της ανάπτυξης λογισμικού. (Abdelrafe , Burairah, & Norhaziah, 2016). Επομένως, είναι σημαντικό να συνδυαστεί ο κύκλος ζωής του λογισμικού με τη διαχείριση κινδύνου σε έργα ανάπτυξης λογισμικού, για να βοηθηθεί ο διαχειριστής λογισμικού να παρακολουθεί και να μετριάξει τον κίνδυνο στο λογισμικό ως προϊόν.

Η διαχείριση κινδύνου συνδέει τις πιθανές ανταποκρίσεις στους κρίσιμους κινδύνους, εντοπίζει τις αιτίες της αποτυχίας και μοιράζεται μια κοινή ιδέα για το έργο μεταξύ των ενδιαφερομένων μερών. Η διαχείριση κινδύνου προσπαθεί να διερευνήσει, να εντοπίσει, να αξιολογήσει και να ξεπεράσει τους κινδύνους που σχετίζονται με όλες τις φάσεις του κύκλου ζωής ανάπτυξης λογισμικού (SDLC) (Bilal, Bashir, & Gani, 2020). Περιλαμβάνει κυρίως δύο κύριες δραστηριότητες, την αξιολόγηση κινδύνου και τον έλεγχο κινδύνου, το καθένα με τρία δευτερεύοντα βήματα. Για παράδειγμα, η αξιολόγηση κινδύνου χωρίζεται στον προσδιορισμό κινδύνου, την ανάλυση κινδύνου και την ιεράρχηση κινδύνου, (Sahu, Pandey, & Kumar, 2014) όπως φαίνεται και στο παρακάτω διάγραμμα. Αυτά τα βήματα είναι ευρέως αποδεκτά.



Εικόνα 9: Βήματα διαχείρισης κινδύνου στα έργα ανάπτυξης λογισμικού (Enfei, 2015)

Επομένως, μπορούμε να ορίσουμε πως η διαχείριση κινδύνου μπορεί να είναι ο εντοπισμός, η αξιολόγηση και η ιεράρχηση των κινδύνων που ακολουθείται από συντονισμένη εφαρμογή πόρων για την ελαχιστοποίηση, παρακολούθηση και έλεγχο της πιθανότητας και/ή του αντίκτυπου ενός ατυχούς γεγονότος ή για τη μεγιστοποίηση των ευκαιριών (Sahu, Pandey, & Kumar, 2014).

4.3.1 Η ΦΥΣΗ ΤΩΝ ΕΡΓΩΝ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ ΣΤΗ ΔΙΑΧΕΙΡΙΣΗ ΚΙΝΔΥΝΩΝ

Όπως αναφέρθηκε στην εισαγωγή αυτού του κεφαλαίου, η διαχείριση κινδύνου σε έργα λογισμικού έχει πολλές ομοιότητες με τη διαχείριση κινδύνου σε άλλους τύπους έργων. Λόγω, όμως, των ιδιαίτερων χαρακτηριστικών του προϊόντος, υπάρχουν ορισμένες συγκεκριμένες εκτιμήσεις και διαφορές όσον αφορά τη διαχείριση κινδύνου σε έργα ανάπτυξης λογισμικού σε σύγκριση με άλλα έργα, οι οποίες αφορούν κυρίως τους κινδύνους των απαιτήσεων, κινδύνους τεχνολογίας και κινδύνους προσωπικού (Enfei, 2015). Πιο αναλυτικά βλέπουμε παρακάτω κάποιες από αυτές τις διαφορές:

- **Αβεβαιότητα και πολυπλοκότητα:**

Έργα ανάπτυξης λογισμικού: Τα έργα ανάπτυξης λογισμικού συχνά περιλαμβάνουν υψηλό βαθμό αβεβαιότητας και πολυπλοκότητας λόγω της δυναμικής φύσης της τεχνολογίας, των εξελισσόμενων απαιτήσεων και της πιθανότητας απρόβλεπτων τεχνικών προκλήσεων.

Άλλα έργα: Αν και άλλα έργα μπορεί επίσης να έχουν αβεβαιότητες, το επίπεδο πολυπλοκότητας στην ανάπτυξη λογισμικού είναι συχνά υψηλότερο λόγω του ταχέως μεταβαλλόμενου πεδίου της τεχνολογίας και της ανάγκης για καινοτόμες λύσεις.

- **Τεχνικοί κίνδυνοι:**

Έργα ανάπτυξης λογισμικού: Οι τεχνικοί κίνδυνοι είναι εγγενείς στην ανάπτυξη λογισμικού, συμπεριλαμβανομένων ζητημάτων που σχετίζονται με την επιλογή τεχνολογίας και την πιθανότητα απρόβλεπτων τεχνικών εμποδίων ή αλλαγών.

Άλλα έργα: Μπορεί να υπάρχουν τεχνικοί κίνδυνοι σε άλλα έργα, αλλά μπορεί να μην προκαλούν ιδιαίτερο πρόβλημα στο έργο τόσο όσο στα έργα ανάπτυξης λογισμικού.

- **Απαιτήσεις Αλλαγές:**

Έργα ανάπτυξης λογισμικού: Οι απαιτήσεις σε έργα ανάπτυξης λογισμικού μπορεί να αλλάζουν συχνά λόγω των εξελισσόμενων αναγκών των πελατών ή της καλύτερης κατανόησης του συστήματος κατά την ανάπτυξη, οδηγώντας σε αλλαγές πεδίου εφαρμογής και πρόσθετους κινδύνους.

Άλλα έργα: Ενώ αλλαγές απαιτήσεων μπορούν να συμβούν σε οποιοδήποτε έργο, μπορεί να είναι πιο σταθερές σε ορισμένους τύπους έργων, όπως τα κατασκευαστικά έργα, σε σύγκριση με τη φύση των απαιτήσεων λογισμικού.

- **Δοκιμές και Διασφάλιση Ποιότητας:**

Έργα ανάπτυξης λογισμικού: Οι δοκιμές και η διασφάλιση ποιότητας διαδραματίζουν κρίσιμο ρόλο στην ανάπτυξη λογισμικού για τον εντοπισμό και τον μετριασμό πιθανών κινδύνων που σχετίζονται με σφάλματα, ζητήματα ασφαλείας και αστοχίες συστήματος.

Άλλα έργα: Οι δοκιμές είναι σημαντικές σε διάφορα έργα, αλλά η φύση και η έκταση των δοκιμών ενδέχεται να διαφέρουν ανάλογα με τον τύπο του έργου. Για παράδειγμα, δοκιμές ασφάλειας σε κατασκευαστικά έργα.

- **Προτυποποίηση και Επαναληπτική Ανάπτυξη:**

Έργα Ανάπτυξης Λογισμικού: Οι ευέλικτες μεθοδολογίες συχνά περιλαμβάνουν ταχεία δημιουργία πρωτοτύπων και επαναληπτική ανάπτυξη, η οποία μπορεί να εισάγει κινδύνους που σχετίζονται με τη διαχείριση των μεταβαλλόμενων προτεραιοτήτων και την ενσωμάτωση αξιολόγησης.

Άλλα έργα: Ενώ επαναληπτικές προσεγγίσεις μπορούν να χρησιμοποιηθούν σε άλλα έργα, η συχνότητα και ο αντίκτυπος των αλλαγών ενδέχεται να διαφέρουν.

- **Ανθρώπινο δυναμικό:**

Έργα ανάπτυξης λογισμικού: Το εξειδικευμένο ανθρώπινο δυναμικό είναι ζωτικής σημασίας για την ανάπτυξη λογισμικού και η απώλεια βασικών μελών της ομάδας μπορεί να αποτελεί σημαντικό κίνδυνο.

Άλλα έργα: Ενώ το ανθρώπινο δυναμικό είναι σημαντικό σε όλα τα έργα, ο βαθμός εξειδίκευσης και ο αντίκτυπός του στον κίνδυνο του έργου μπορεί να διαφέρουν.

- **Συμμόρφωση σε κανονισμούς:**

Έργα ανάπτυξης λογισμικού: Η συμμόρφωση με νομικούς κανόνες, ειδικά σε κλάδους όπως η υγειονομική περίθαλψη ή η χρηματοδότηση, προσθέτει ένα επιπλέον επίπεδο πολυπλοκότητας και κινδύνου.

Άλλα έργα: Οι κανονισμοί μπορεί να διαφέρουν και η συμμόρφωση μπορεί να μην είναι τόσο αυστηρή σε ορισμένους άλλους τύπους έργων. Παρ' όλα αυτά, όλων των φύσεων τα έργα οφείλουν να συμμορφώνονται στους αντίστοιχους κανονισμούς και νόμους.

- **Πίεση χρόνου μέχρι το προϊόν να βγει στην αγορά:**

Έργα Ανάπτυξης Λογισμικού: Ο γρήγορος ρυθμός της τεχνολογικής αλλαγής και ο ανταγωνισμός της αγοράς συχνά δημιουργούν πιέσεις από την αρχή της υλοποίησης έως ότου να βγει το προϊόν στην αγορά, οι οποίες μπορεί να συμβάλουν σε κινδύνους που σχετίζονται με τα χρονοδιαγράμματα του έργου.

Άλλα έργα: Ενώ οι χρονικοί περιορισμοί είναι συνηθισμένοι σε πολλά έργα, ο επείγων χαρακτήρας μπορεί να είναι διαφορετικός, ανάλογα με τον κλάδο και τον τύπο του έργου.

Συνοπτικά, ενώ οι θεμελιώδεις αρχές της διαχείρισης κινδύνου ισχύουν σε όλα τα έργα, η ειδική φύση και η έμφαση σε ορισμένους παράγοντες κινδύνου μπορεί να διαφέρει σημαντικά μεταξύ έργων ανάπτυξης λογισμικού και άλλων τύπων έργων. Η κατανόηση αυτών των διαφορών είναι ζωτικής σημασίας για την αποτελεσματική διαχείριση των κινδύνων.

4.3.2 Η ΕΝΝΟΙΑ ΤΗΣ ΕΠΙΤΥΧΙΑΣ ΣΤΑ ΕΡΓΑ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ

Τα έργα λογισμικού είναι πιο πιθανό να αποτύχουν λόγω της περίπλοκης και μοναδικής φύσης τους. Παρά τη βελτίωση των μεθοδολογιών ανάπτυξης και των στρατηγικών διαχείρισης, το ποσοστό αποτυχίας έργων λογισμικού που αναφέρθηκε σε διάφορες έρευνες παρέμεινε το ίδιο (40 – 50 %) τις τελευταίες δεκαετίες. Για να αυξηθούν οι πιθανότητες επιτυχίας των έργων λογισμικού και να ξεπεραστεί η αποτυχία τους, χρειάζεται να ακολουθηθούν μέθοδοι διαχείρισης πιθανών κινδύνων. Η διαχείριση κινδύνων σε έργα λογισμικού θεωρείται ως αποτελεσματική προσέγγιση για την εξάλειψη αυτών. Η σημασία και η ανάγκη για την υιοθέτηση διαδικασιών διαχείρισης κινδύνου λογισμικού έχουν επίσης αναγνωριστεί από μεγάλες εταιρείες ανάπτυξης λογισμικού, όπως η Microsoft, η IBM κ.ά. (Bilal, Bashir, & Gani, 2020) .

Αν και υπάρχουν πολλές μέθοδοι διαχείρισης κινδύνων λογισμικού, τα έργα ανάπτυξης λογισμικού έχουν υψηλό ποσοστό αποτυχίας. Έτσι, εάν αυξηθεί η πολυπλοκότητα και το μέγεθος των έργων λογισμικού, η διαχείριση του κινδύνου ανάπτυξης λογισμικού γίνεται πιο δύσκολη. (Abdelrafe , Burairah, & Norhaziah, 2016)

Η διαχείριση κινδύνου παρέχει έναν δομημένο μηχανισμό για να παρέχει ορατότητα στις απειλές που προβάλλει την επιτυχία. (Sahu, Pandey, & Kumar, 2014). Οι Abdelrafe , Burairah, και Norhaziah, αναφέρουν στην έρευνά τους πως η αποτελεσματική αξιολόγηση των κινδύνων όχι μόνο βοηθά στον εντοπισμό και την ανάλυση κρίσιμων κινδύνων, αλλά βελτιώνει επίσης τις πιθανότητες να επιτύχουν τα έργα ανάπτυξης λογισμικού. Είναι σημαντικό να κατανοηθούν, επίσης, οι παράγοντες που συμβάλλουν στην επιτυχία του έργου καθώς και να αναγνωριστούν εκείνοι που αποτελούν κίνδυνο αποτυχίας ώστε να μετριαστούν. Τα έργα λογισμικού που ακολουθούν διαχείριση κινδύνων φαίνεται να είναι πιο επιτυχημένα από τα έργα που δεν περιλαμβάνουν μεθόδους διαχείρισης (Abdelrafe , Burairah, & Norhaziah, 2016).

Στη βιβλιογραφία παρατηρείται πως σε ένα έργο ανάπτυξης λογισμικού, η αποτυχία περιγράφει μια αρνητική επίδραση στο έργο, η οποία θα μπορούσε να είναι με τη μορφή μειωμένης ποιότητας του τελικού έργου, αυξημένο κόστος, αναβολή ολοκλήρωσης ή πλήρης αποτυχία του έργου. Η μέτρηση της επιτυχίας ενός έργου λογισμικού είναι ζωτικής σημασίας για την κατανόηση του αντίκτυπου του και για τη διασφάλιση ότι συμφωνεί με τους στόχους και τις προσδοκίες των ενδιαφερομένων μερών. Διάφοροι παράγοντες συμβάλλουν στην επιτυχία ενός έργου ανάπτυξης λογισμικού, και ως εκ τούτου, ένας συνδυασμός ποσοτικών και ποιοτικών μέτρων είναι συχνά απαραίτητος.

Οι Van der Westhuizen και Fitzgerald, για παράδειγμα, όπως και οι Procaccino και Verner, διαχωρίζουν την επιτυχία των έργων λογισμικού σε επιτυχία της διαχείρισης του έργου και σε επιτυχία του αποτελέσματος ως τελικό προϊόν (Van der Westhuizen & Fitzgerald, 2005) (Procaccino & Verner, 2006). Παρόλο που η επιτυχία ενός έργου λογισμικού οφείλεται και στο ένα (διαχείριση έργου) και στο άλλο (τελικό προϊόν), οι ερευνητές προκειμένου να ορίσουν και να «μετρήσουν» την επιτυχία των έργων λογισμικού, διαχώρισαν αυτές τις δύο έννοιες για την καλύτερη κατανόηση των παραγόντων που προκαλούν επιτυχία . Πιο συγκεκριμένα, Η επιτυχία που οφείλεται στην επιτυχία της διαχείρισης αφορά περισσότερο εσωτερικούς παράγοντες και έχει να

κάνει με το χρονοδιάγραμμα, το κόστος και την ποιότητα (τρίγωνο) ή και με την επικοινωνία και την παρακίνηση του διαχειριστή προς την ομάδα. Από την άλλη, η επιτυχία που οφείλεται στο έργο ως τελικό προϊόν, αφορά στην ποιότητα του προϊόντος και στην ικανοποίηση των ενδιαφερόμενων μερών.

Αν και τα αποτελέσματα της επιτυχίας της διαχείρισης έργου και της επιτυχίας του προϊόντος του έργου είναι άρρηκτα συνδεδεμένα, η αιτιώδης σχέση μεταξύ τους είναι ακόμη ασθενής. Για παράδειγμα, το να επέλθει ο χρόνος και/ή ο προϋπολογισμός, το έργο θα θεωρηθεί αποτυχημένο λόγω κακής διαχείρισης έργου, αλλά το προϊόν που προκύπτει μπορεί να είναι επιτυχές.

Επιπλέον, από την πλευρά της ομάδας ανάπτυξης, σημαντικοί παράγοντες που συμβάλουν στην επιτυχία ενός έργου ανάπτυξης λογισμικού και πρέπει να λαμβάνονται σοβαρά υπόψη από τους διαχειριστές, είναι οι άνθρωποι (η ομάδα, τα ενδιαφερόμενα μέρη κτλ), οι διαδικασίες ανάπτυξης, οι τεχνικές μέθοδοι, οι τεχνολογίες και τα εργαλεία ανάπτυξης που χρησιμοποιούνται για την υλοποίηση. Μια ομάδα ανάπτυξης λογισμικού χρειάζεται την ισχυρή συμμετοχή των πελατών, καλή διαχείριση του έργου από το διαχειριστή, καλές και σωστές τεχνικές, τεχνολογίες και εργαλεία ανάπτυξης, προκειμένου να εργαστεί στο μέγιστο δυνατό των ικανοτήτων της και να φέρει εις πέρας το έργο με επιτυχία (Nguyen, 2016).

Είναι σημαντικό να προσαρμοστούν αυτές οι μετρήσεις με τους στόχους και το πλαίσιο κάθε έργου ανάπτυξης λογισμικού. Η συχνή αναθεώρηση και προσαρμογή αυτών των μέτρων καθ' όλη τη διάρκεια του κύκλου ζωής του έργου μπορεί να προσφέρει πολύτιμες πληροφορίες και να βοηθήσει στη διασφάλιση της επιτυχίας. Επιπλέον, τα κριτήρια επιτυχίας θα πρέπει να καθοριστούν σε συνεργασία με τα ενδιαφερόμενα μέρη στην αρχή του έργου για να τεθούν σαφείς προσδοκίες από την πλευρά τους.

ΚΕΦΑΛΑΙΟ 5 – ΜΕΘΟΔΟΛΟΓΙΑ

5.1 ΣΥΛΛΟΓΗ ΤΩΝ ΔΕΔΟΜΕΝΩΝ

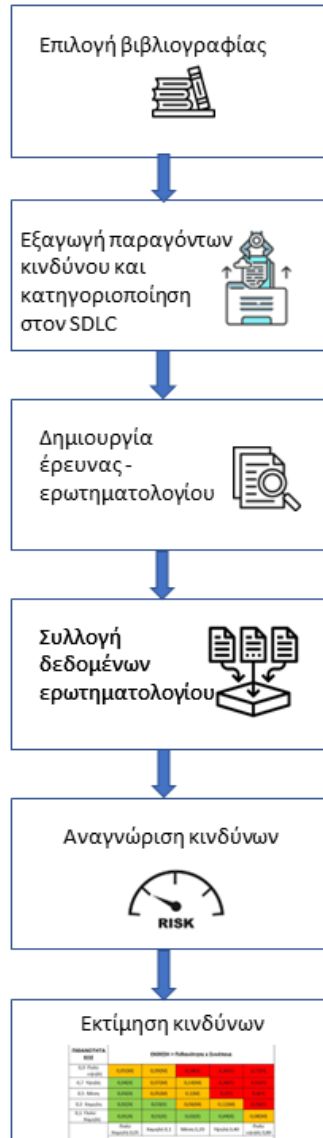
Σκοπός της παρούσας εργασίας είναι να αναγνωρισθούν οι πιο συχνοί και πιθανοί παράγοντες κινδύνων που παρουσιάζονται στα έργα ανάπτυξης λογισμικού και προκαλούν αποτυχία στο έργο, και να αξιολογηθούν. Για την έρευνα και αναγνώριση των πιο συχνών και πιθανών κινδύνων, συντάχθηκε ερωτηματολόγιο, το οποίο δόθηκε σε εργαζομένους διάφορων τίτλων εργασίας που ασχολούνται με έργα ανάπτυξης λογισμικού και διάφορων υπηκοοτήτων, προκειμένου να συλλεχθούν τα δεδομένα.

Σε αυτή την έρευνα πρώτα έγινε μια ποιοτική έρευνα για να βρεθούν οι πιο συχνοί παράγοντες κινδύνου που αναφέρονται στη βιβλιογραφία. Έπειτα, συντάχθηκε το ερωτηματολόγιο και χρησιμοποιήθηκε ποσοτική έρευνα για την ανάλυση των δεδομένων που συλλέχθηκαν από αυτό. Η αναγνώριση των κινδύνων έγινε σύμφωνα με τις απαντήσεις των ερωτηθέντων ενώ, τέλος, εκτιμήθηκαν οι κίνδυνοι και δημιουργήθηκε ο πίνακας κινδύνου.

Τα βήματα, συνεπώς, ήταν:

1. Επιλογή βιβλιογραφίας
2. Εξαγωγή κινδύνων και κατηγοριοποίηση στον SDLC
3. Δημιουργία έρευνας-ερωτηματολογίου
4. Συλλογή δεδομένων ερωτηματολογίου
5. Αναγνώριση κινδύνων (σύμφωνα με την ανάλυση των δεδομένων)
6. Εκτίμηση κινδύνων

Τα οποία παρουσιάζονται και στο παρακάτω διάγραμμα:



Εικόνα 10: Μεθοδολογία συλλογής και ανάλυσης δεδομένων

Το μέγεθος του πληθυσμού δεν είναι γνωστό, καθώς το ερωτηματολόγιο διαμοιράσθηκε σε εργαζομένους διαφορετικών κλάδων επιχειρήσεων (αυτοκινητοβιομηχανία, εφαρμογές πληροφορικής στοιχηματισμού κ.ά), οι οποίοι με τη σειρά τους ζήτησαν από φίλους και γνωστούς που εργάζονται σε αντίστοιχα έργα να το συμπληρώσουν. Το δείγμα των ερωτηθέντων ανέρχεται σε 48.

Το ερωτηματολόγιο προωθήθηκε στους υποψήφιους συμμετέχοντες μέσω διαδικτύου (e-mail και μέσα κοινωνικής δικτύωσης) καθώς ο συγκεκριμένος τρόπος διανομής είναι πιο εύκολος σε αυτή την εποχή και ειδικότερα όταν ο πληθυσμός είναι μεγάλος, βρίσκεται σε διαφορετικές χώρες και είναι άγνωστος.

Η μέθοδος που ακολουθήθηκε, συνεπώς, για τον διαμοιρασμό του ερωτηματολογίου ήταν «η μέθοδος της χιονοστιβάδας», κατά την οποία δειγματοληπτική τεχνική ο ερευνητής επιλέγει τυχαία ένα μικρό αριθμό ατόμων από ένα πληθυσμό, όπου κάθε άτομο στο δείγμα επιλέγει κάποιον άλλο αριθμό ατόμων και στη συνέχεια αυτό μεγαλώνει και επεκτείνεται σαν «χιονόμπαλα».

Η μέθοδος της χιονοστιβάδας, είναι μία από τις δημοφιλέστερες μεθόδους δειγματοληψίας που αφορούν την ποιοτική έρευνα, λόγω των χαρακτηριστικών και της ευκολίας της, ενώ παράλληλα θεωρείται πως είναι η κατάλληλη μέθοδος για τη συλλογή πληροφοριών από συγκεκριμένους πληθυσμούς, οι οποίοι είναι δύσκολο να προσεγγιστούν. Συνεπώς, πραγματοποιείται μια σκόπιμη δειγματοληψία (Abdelrafe , Burairah, & Norhaziah, 2016).

5.2 ΣΥΝΤΑΞΗ ΤΟΥ ΕΡΩΤΗΜΑΤΟΛΟΓΙΟΥ

Υπάρχουν πολλές τεχνικές συλλογής δεδομένων στη βιβλιογραφία, όμως η πιο διαδεδομένη και εύκολη στη συλλογή των αποτελεσμάτων στις μέρες μας είναι αυτή του ερωτηματολογίου (Shahzad, 2012). Ένα ερωτηματολόγιο μπορεί να περιέχει ερωτήσεις ανοιχτού και κλειστού τύπου. Για την ευκολία των απαντήσεων και της συλλογής αυτών, χρησιμοποιήθηκαν μόνο ερωτήσεις κλειστού τύπου (βλ. Παράρτημα).

Αφού έγινε η αναγνώριση των πιο διαδεδομένων κινδύνων από τη βιβλιογραφία, οι ερωτηθέντες καλέστηκαν να απαντήσουν σύμφωνα με την εμπειρία τους στα έργα που απασχολούνται (ή απασχολήθηκαν στο παρελθόν) αξιολογώντας σε κλίμακα Likert επιπέδων από 1 έως 5 ή επιλέγοντας από απαντήσεις πολλαπλής επιλογής όπως: Όχι, Ναι, Ίσως, σε μεγάλο βαθμό κτλ.

Οι ερωτήσεις συντάχθηκαν και κατηγοριοποιήθηκαν για κάθε φάση του κύκλου ζωής των έργων λογισμικού ώστε να αναγνωρισθούν οι πιο συχνόι κίνδυνοι κάθε φάσης σύμφωνα με τις απαντήσεις των ερωτηθέντων. Κάποιο σύνολο ερωτήσεων μπορεί να αναφέρονται στην αναγνώριση ενός παράγοντα κινδύνου. Για κάθε παράγοντα το σύνολο των ερωτήσεων ήταν από 1 έως 3 ενώ για κάθε φάση οι ερωτήσεις κυμαίνονταν από 3 έως 15. Για παράδειγμα, για να αναγνωρισθεί εάν ο Παράγοντας «Μη ρεαλιστικός προϋπολογισμός» αποτελεί κίνδυνο στη φάση του σχεδιασμού του έργου, σύμφωνα με την κρίση των ερωτηθέντων, δόθηκαν δύο ερωτήσεις: «Ποια είναι η ισορροπία μεταξύ χρόνου και προσπάθειας στο έργο?» και «Πόσο καλά πιστεύετε ότι έχει υπολογισθεί το κόστος του έργου».

Το ερωτηματολόγιο περιλάμβανε συνολικά 75 ερωτήσεις, συμπεριλαμβανομένων και των ερωτήσεων για τη συλλογή των δημογραφικών στοιχείων των ερωτηθέντων, όπως το φύλο, τον κλάδο απασχόλησης, το επίπεδο εκπαίδευσης κτλ. Τέλος, συντάχθηκε στην αγγλική γλώσσα καθώς απευθύνθηκε σε εργαζομένους που απασχολούνται σε διάφορες χώρες και, συνεπώς, μπορεί να ανήκουν σε διαφορετικές υπηκοότητες που δε μιλούν την ελληνική γλώσσα.

5.3 ΑΝΑΛΥΣΗ ΤΩΝ ΔΕΔΟΜΕΝΩΝ

Λαμβάνοντας υπόψη την κλίμακα της έρευνας, είναι αδύνατο να συμπεριληφθούν όσο το δυνατόν περισσότερα άτομα που να μπορούν να επιβεβαιώσουν μια ζητούμενη απάντηση και να ληφθεί μια ισχυρή επικύρωση. Οφείλει, ωστόσο, να σημειωθεί ότι οι απαντήσεις πρέπει να είναι ακριβείς και να προέρχονται από έμπειρους χρήστες. Σκοπός είναι να αναγνωρισθούν οι πιο δημοφιλείς παράγοντες κινδύνου στα έργα λογισμικού σύμφωνα με τις απαντήσεις των ερωτηθέντων που εργάζονται σε αντίστοιχα έργα.

Για να γίνει αυτό χρησιμοποιήθηκε η μέθοδος της **μέσης σταθμισμένης απόκρισης (average weighted response)**, η οποία ερμηνεύεται ως μία εκτίμηση της μέσης πιθανότητας απόκρισης στα μέλη του πληθυσμού και χρησιμοποιείται για τη συλλογή των απαντήσεων και την ανάλυση των δεδομένων, όπως αυτή χρησιμοποιήθηκε εξίσου και στην έρευνα του Shahzad (Shahzad, 2012).

Η μέση σταθμισμένη απόκριση για την ανάλυση των απαντήσεων του ερωτηματολογίου υπολογίστηκε για κάθε ερώτηση που αντιστοιχεί σε κάποιο παράγοντα κινδύνου ξεχωριστά, πολλαπλασιάζοντας τον αριθμό των απαντήσεων με το βάρος της αποκριθείσας μονάδας (βάρος κλίμακας Likert), αθροίζοντας το σύνολο αυτών και στη συνέχεια διαιρώντας με τον αριθμό των συνολικών απαντήσεων. Για τους παράγοντες που περιλάμβαναν περισσότερες από μια ερωτήσεις, υπολογίστηκε η μέση τιμή των σταθμισμένων αποκρίσεων των αντίστοιχων ερωτήσεων. Κάποια ενδεικτικά αποτελέσματα παρουσιάζονται στους παρακάτω πίνακες:

Πίνακας 1: Υπολογισμός μέσης σταθμισμένης απόκρισης απαντήσεων κάθε ερώτησης

Question/Likert Scale	1	2	3	4	5	Average Weighted response
Is the project scope (e.g. size, goals, requirements etc) clear to the project members?	0	0	22	0	26	4.083
Do you believe that the project schedule (completion time or deadlines) is realistic?	4	0	28	0	16	3.500
How much realistic you find it? (Please scale your	0	6	12	30	0	3.500

answer from 1 as not at all to 5 as very realistic)						
Which is the balance between time and effort of the project?	18	0	30	0	0	2.250
How well is estimated the cost of the project? (Please scale your answer from 1 as very bad to 5 as very well)	2	2	20	14	0	3.211
Is the number of the project members sufficient?	16	0	0	32	0	3.000
Is the technology that is used for the project (e.g. tools) updated?	12	0	26	0	10	2.917
From 1 to 5, how sufficient is the estimation of the required resources (people, tools, technologies)?	0	6	22	16	4	3.375

Πίνακας 2: Συνοπτικός πίνακας υπολογισμού μέσης σταθμισμένης απόκρισης απαντήσεων κάθε ερώτησης

Risk Factor	Question	Average Weighted response
Unclear project scope to the manager and/or the other project members	Is the project scope (e.g. size, goals, requirements etc) clear to the project members?	4.083
Unrealistic project schedule	Do you believe that the project schedule (completion time or deadlines) is realistic?	3.500
	How much realistic you find it? (Please scale your answer from 1 as not at all to 5 as very realistic)	3.500

Unrealistic budget	Which is the balance between time and effort of the project?	2.250
	How well is estimated the cost of the project? (Please scale your answer from 1 as very bad to 5 as very well)	3.211
Insufficient resources	Is the number of the project members sufficient?	3.000
	Is the technology that is used for the project (e.g. tools) updated?	2.917
	From 1 to 5, how sufficient is the estimation of the required resources (people, tools, technologies)?	3.375

Θεωρείται ότι ένας παράγοντας κινδύνου είναι δημοφιλής, αποτελεί δηλαδή κίνδυνο, όταν οι περισσότεροι ερωτηθέντες έχουν αξιολογήσει τις απαντήσεις τους με το λιγότερο βάρος της κλίμακας. Για παράδειγμα, στην ερώτηση «Είναι οι προδιαγραφές του συστήματος κατανοητές από την ομάδα υλοποίησης;» εάν οι περισσότεροι απαντήσουν «Όχι» (βάρος κλίμακας Likert 1 ή 2), τότε θεωρούμε ότι ο παράγοντας «Μη ξεκάθαρες προδιαγραφές συστήματος» είναι δημοφιλής καθώς η μέση σταθμισμένη απόκριση για τον παράγοντα αυτό θα είναι μικρή.

Έτσι, υπολογίστηκε ένα κατώτατο όριο (Threshold) για κάθε φάση του κύκλου ζωής των έργων λογισμικού, παίρνοντας τη μέση τιμή των μεγαλύτερων και μικρότερων μέσων σταθμισμένων αποκρίσεων. Σκοπός του υπολογισμού αυτού, ήταν να αναγνωρισθεί εάν κάθε παράγοντας κινδύνου αποτελεί δημοφιλή κίνδυνο σύμφωνα με τις απαντήσεις των ερωτηθέντων. Ο κίνδυνος γίνεται αποδεκτός ως δημοφιλής, στην περίπτωση που η μέση σταθμισμένη απόκριση του αντίστοιχου παράγοντα βρίσκεται κάτω από το κατώτατο όριο, ώστε να θεωρηθεί κίνδυνος.

Ενδεικτικά αποτελέσματα αυτού του υπολογισμού παρουσιάζονται στον παρακάτω πίνακα:

Πίνακας 3: Ενδεικτικός πίνακας υπολογισμού αποδοχής ή μη αποδοχής παραγόντων κινδύνου

Risk Factor Number	Risk factor	Average Weighted response	Threshold	Accepted/rejected if < Threshold
R1	Unclear project scope to the manager and/or the other project members	4.08	3.41	Rejected
R2	Unrealistic project schedule	3.5	3.41	Rejected
R3	Unrealistic budget	2.73	3.41	Accepted
R4	Insufficient resources	3.10	3.41	Accepted

Έπειτα, σύμφωνα με τα αποτελέσματα του υπολογισμού της αποδοχής ή μη αποδοχής των παραγόντων κινδύνου ως δημοφιλείς σε κάθε φάση του κύκλου ζωής, αναγνωρίστηκαν οι παράγοντες κινδύνου που αποτελούν κίνδυνο στα έργα ανάπτυξης λογισμικού κατά τις απαντήσεις των ερωτηθέντων του ερωτηματολογίου. Στη συνέχεια κατηγοριοποιήθηκαν στον τύπο κινδύνου που αποτελούν, ενώ, παράλληλα αξιολογήθηκε η πιθανότητα και η συνέπεια του καθενός σύμφωνα με του πίνακες αξιολόγησης του PMI (Project Management Institute):

Πίνακας 4: Πίνακας υπολογισμού πιθανότητας κινδύνου κατά PMI (Project Management Institute)

Πιθανότητα	Επεξήγηση
Πολύ χαμηλή (0,1)	Συμβαίνει 1 φορά στα 30 χρόνια
Χαμηλή (0,3)	Συμβαίνει 1 φορά στα 10 χρόνια
Μέση (0,5)	Συμβαίνει 1 φορά το χρόνο
Υψηλή (0,7)	Συμβαίνει 1 φορά το μήνα
Πολύ υψηλή (0,9)	Συμβαίνει 10 φορές το μήνα

Πίνακας 5: Πίνακας υπολογισμού συνέπειας κινδύνου κατά PMI (Project Management Institute)

Συνέπεια	Επεξήγηση
Πολύ χαμηλή (0,05)	Αμελητέα συνέπεια στο σύστημα ή τους χρήστες (π.χ. ενόχληση)
Χαμηλή (0,1)	Χαμηλή συνέπεια στο σύστημα ή τους χρήστες (π.χ. μικροτραυματισμός)
Μέση (0,2)	Μέση συνέπεια στο σύστημα ή τους χρήστες (π.χ. τραυματισμός, ανάγκη επισκευής συστήματος)
Υψηλή (0,4)	Σημαντική συνέπεια στο σύστημα ή τους χρήστες (π.χ. σοβαρός τραυματισμός, μικρής διάρκειας διακοπή λειτουργίας του συστήματος)
Πολύ υψηλή (0,8)	Σοβαρή συνέπεια στο σύστημα ή τους χρήστες (π.χ. θάνατος, μερική καταστροφή του συστήματος)

Τέλος, δημιουργήθηκε ο πίνακας κινδύνου υπολογίζοντας την έκθεση του κινδύνου σύμφωνα, εξίσου, με τους πίνακες του PMI (Project Management Institute) και ένα σχέδιο απόκρισης για κάθε έναν.

Πίνακας 6: Πίνακας υπολογισμού έκθεσης κινδύνου κατά PMI (Project Management Institute)

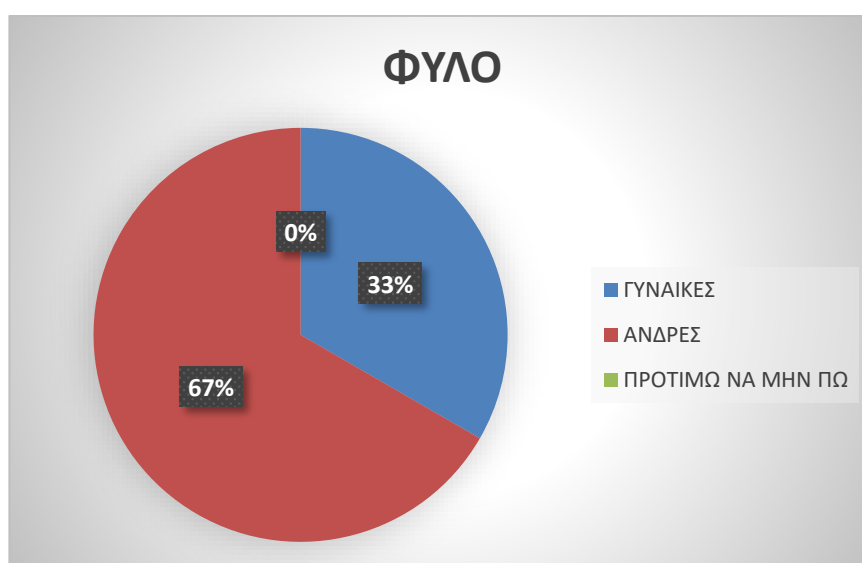
ΕΚΘΕΣΗ ΈΡΓΟΥ ΣΕ ΚΙΝΔΥΝΟ (κατά PMI)					
ΠΙΘΑΝΟΤΗΤΑ ΕΩΣ	ΕΚΘΕΣΗ = Πιθανότητα x Συνέπεια				
0,9 Πολύ υψηλή	0,05(M)	0,09(M)	0,18(Y)	0,36(Y)	0,72(Y)
0,7 Υψηλή	0,04(X)	0,07(M)	0,14(M)	0,28(Y)	0,56(Y)
0,5 Μέση	0,03(X)	0,05(M)	0,1(M)	0,2(Y)	0,4(Y)
0,3 Χαμηλή	0,02(X)	0,03(X)	0,06(M)	0,12(M)	0,24(Y)
0,1 Πολύ Χαμηλή	0,01(X)	0,01(X)	0,02(X)	0,04(X)	0,08(M)
	Πολύ Χαμηλή 0,05	Χαμηλή 0,1	Μέση 0,20	Υψηλή 0,40	Πολύ υψηλή 0,80
	ΣΥΝΕΠΕΙΑ				
Y: Υψηλός Κίνδυνος – μη αποδεκτός, χρειάζεται άμεση αντίδραση M: Μέσος Κίνδυνος, μπορεί να χρειάζεται αντίδραση X: Χαμηλός Κίνδυνος, απλή παρακολούθηση					

Τα αποτελέσματα παρουσιάζονται αναλυτικά στο παρακάτω κεφάλαιο.

ΚΕΦΑΛΑΙΟ 6 – ΑΠΟΤΕΛΕΣΜΑΤΑ

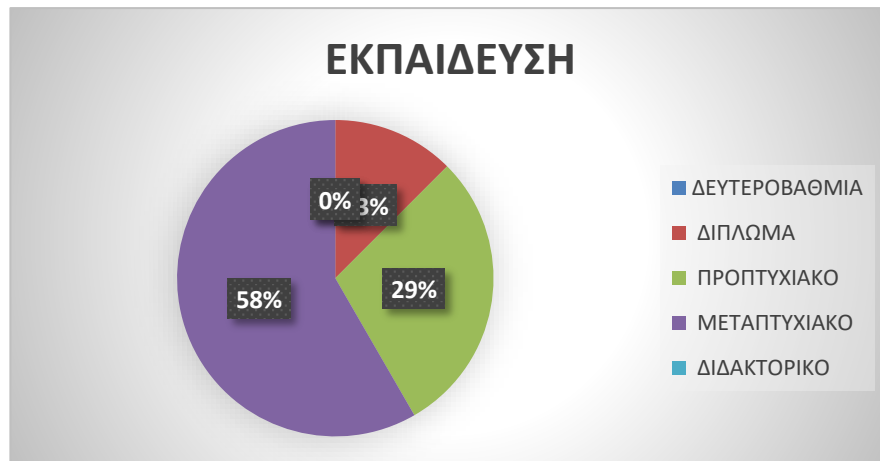
6.1 ΔΗΜΟΓΡΑΦΙΚΑ ΣΤΟΙΧΕΙΑ

Όπως προαναφέρθηκε σε προηγούμενο κεφάλαιο, το δείγμα του πληθυσμού των ερωτηθέντων ανέρχεται σε 48. Από αυτούς οι 16 ήταν γυναίκες και οι 32 άντρες.



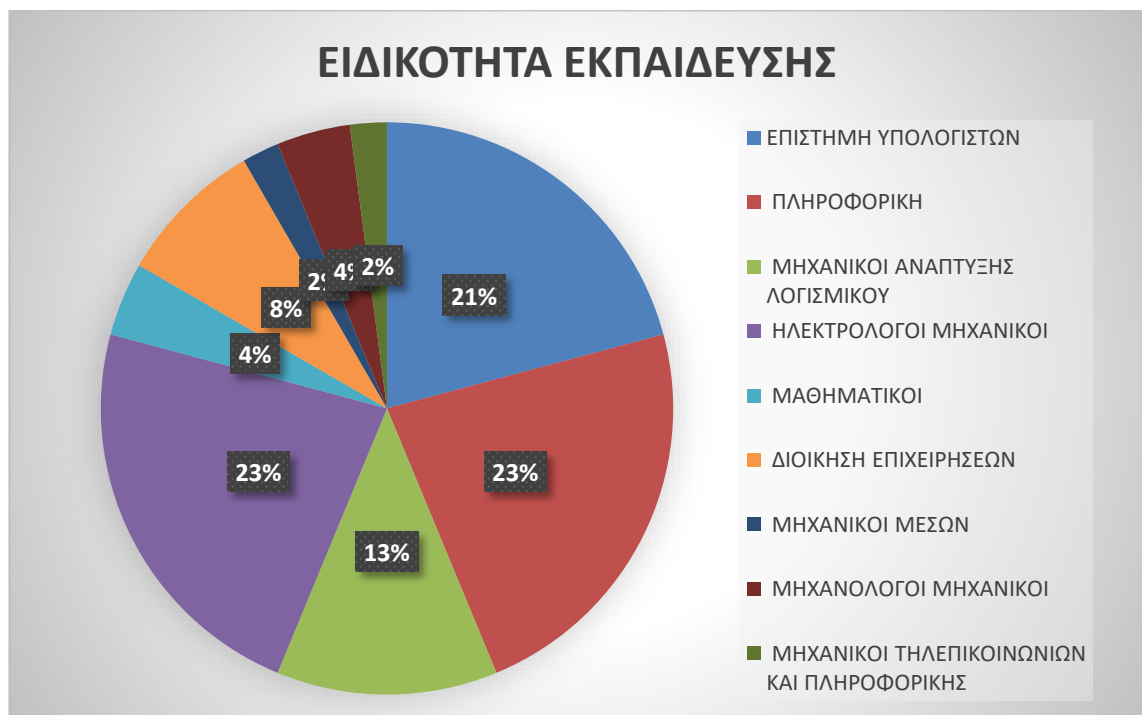
Εικόνα 11: Διάγραμμα φύλου των ερωτηθέντων

Η επόμενη ενότητα για τη συλλογή των δημογραφικών στοιχείων αναφέρεται στην εκπαίδευση των εργαζομένων σε έργα λογισμικού. Το 26% έχει στην κατοχή του Δίπλωμα, το 29% Προπτυχιακό (Bachelor) ενώ το 58% έχει κάνει Μεταπτυχιακές σπουδές (Master).



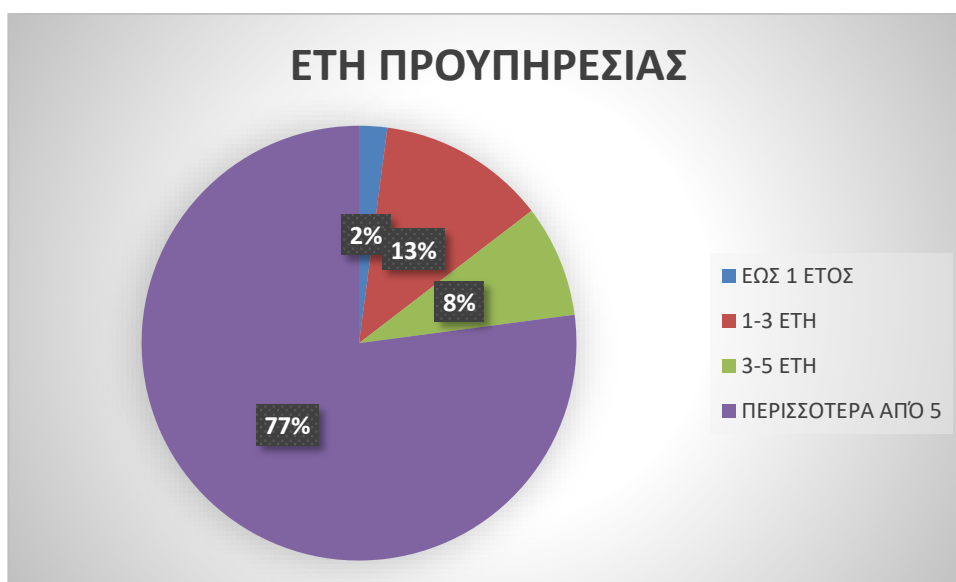
Εικόνα 12: Διάγραμμα εκπαίδευσης των ερωτηθέντων

Οι περισσότεροι σπούδασαν Πληροφορική, Επιστήμη Υπολογιστών ή Ηλεκτρολόγοι Μηχανικοί, ενώ αμέσως μετά έρχεται σε μεγάλο ποσοστό και οι Μηχανικοί Ανάπτυξης Λογισμικού καθώς και εκείνοι που έχουν σπουδές στη Διοίκηση Επιχειρήσεων. Το ερωτηματολόγιο φαίνεται να απαντήθηκε σε πολύ μικρό ποσοστό (2% - 4%) από Μαθηματικούς, Μηχανολόγους Μηχανικούς, Μηχανικούς Μέσων ή Μηχανικούς Τηλεπικοινωνιών και Πληροφορικής.

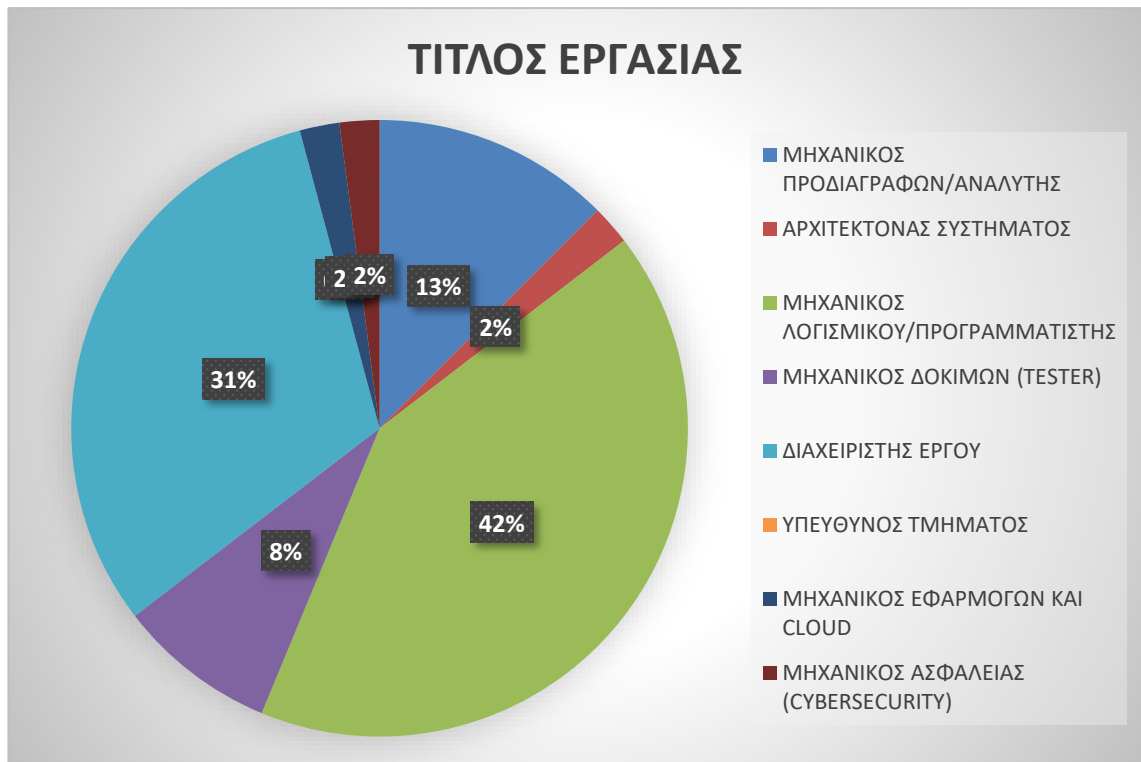


Εικόνα 13: Διάγραμμα εκπαίδευσης των ερωτηθέντων

Η τελευταία ενότητα για τη συλλογή των δημογραφικών στοιχείων αφορά στην απασχόληση των εργαζομένων. Φαίνεται πως η πλειοψηφία αφορά έμπειρους εργαζομένους (πάνω από 5 έτη προϋπηρεσίας). Το 42% εργάζονται ως Μηχανικοί Λογισμικού ή Προγραμματιστές, το 31% ως Διαχειριστές έργου (Project Managers), ενώ μόλις 13% των ερωτηθέντων εργάζονται ως Μηχανικοί Προδιαγραφών ή Αναλυτές και το 8% ως Μηχανικοί Δοκιμών (Testers). Λίγοι είναι εκείνοι που εργάζονται ως Αρχιτέκτονες Συστήματος (2%), Μηχανικοί εφαρμογών και Cloud (2%) ή Μηχανικοί Ασφαλείας δικτύου (Cybersecurity) (2%).



Εικόνα 14: Έτη προϋπηρεσίας των ερωτηθέντων



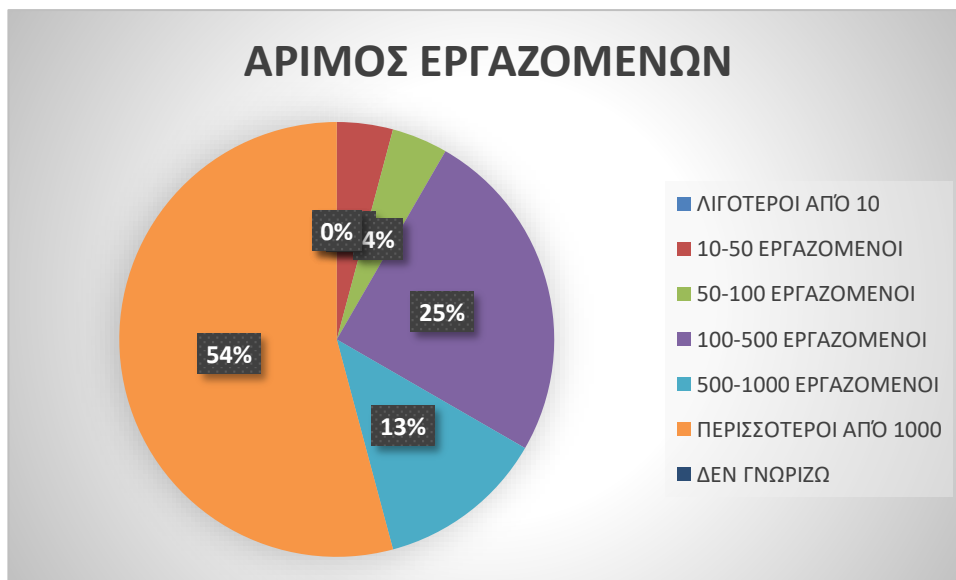
Εικόνα 15: Τίτλοι εργασίας των ερωτηθέντων

Οι εργαζόμενοι απασχολούνται σε έργα ανάπτυξης λογισμικού αλλά σε διαφορετικούς κλάδους. Όπως παρουσιάζεται και στο παρακάτω διάγραμμα, 26 από τους 48 απασχολούνται σε έργα λογισμικού του τομέα της Αυτοκινητοβιομηχανίας, 10 ερωτηθέντες στο ΙΟΤ (Internet of Things), 8 σε εφαρμογές ανάπτυξης που αφορούν το δημόσιο τομέα, ενώ μόλις 2 σε εφαρμογές στοιχηματισμού ή συμβουλευτικής.



Εικόνα 16: Κλάδος απασχόλησης των ερωτηθέντων

Τέλος, το μέγεθος των επιχειρήσεων στις οποίες απασχολούνται οι εργαζόμενοι φαίνεται να είναι αρκετά μεγάλο στις περισσότερες περιπτώσεις (πάνω από 1000), αμέσως μετά στο 25% οι επιχειρήσεις που εργάζονται οι ερωτηθέντες απασχολούν 100-500 εργαζομένους ενώ 13% απασχολούν 500-1000. Σε μικρό ποσοστό (2%) οι ερωτηθέντες απασχολούνται σε μικρές επιχειρήσεις (10-100 εργαζομένους).



Εικόνα 17: Μέγεθος των επιχειρήσεων που απασχολούνται οι ερωτηθέντες

6.2 ΑΠΟΤΕΛΕΣΜΑΤΑ ΑΝΑΛΥΣΗΣ ΔΕΔΟΜΕΝΩΝ

Παρακάτω αναλύονται τα αποτελέσματα που προκύπτουν από την ανάλυση των δεδομένων. Όπως αναφέρθηκε για την αναγνώριση των παραγόντων κινδύνου που αποτελούν κίνδυνο σύμφωνα με τις απαντήσεις των ερωτηθέντων, υπολογίσθηκε ένα κατώτατο όριο (Threshold) για κάθε φάση του κύκλου ζωής των έργων λογισμικού, παίρνοντας τη μέση τιμή των μεγαλύτερων και μικρότερων μέσων σταθμισμένων αποκρίσεων. Ο κίνδυνος, λοιπόν, γίνεται αποδεκτός ως δημοφιλής, στην περίπτωση που η μέση σταθμισμένη απόκριση του αντίστοιχου παράγοντα βρίσκεται κάτω από το κατώτατο όριο, ώστε να θεωρηθεί κίνδυνος.

Από την ανάλυση των δεδομένων προέκυψαν τα εξής αποτελέσματα:

- **Φάση Σχεδιασμού του έργου (Προγραμματισμός)**

Κατά τον σχεδιασμό των έργων λογισμικού θεωρείται ότι οι πιο δημοφιλείς παράγοντες είναι ο «**Μη ρεαλιστικός προϋπολογισμός (R3)**» των έργων και οι «**Ανεπαρκείς πόροι (R4)**». Ως ανεπαρκείς πόρους θεωρούμε ότι το ανθρώπινο δυναμικό δεν επαρκεί για να ολοκληρωθεί το έργο χωρίς προβλήματα ή ότι τα εργαλεία ή οι τεχνολογίες που χρησιμοποιούνται για την εκτέλεση του έργου δεν αρκούν ή είναι απαρχαιωμένα. Οι παράγοντες «**Ασαφές στόχος ή σκοπός του έργου για τον διαχειριστή ή/και τα μέλη του έργου (R1)**» και «**Μη ρεαλιστικό χρονοδιάγραμμα έργου (R2)**» απορρίφθηκαν ως κίνδυνοι.

Πίνακας 7: Δημοφιλείς παράγοντες κινδύνου κατά τη φάση του Σχεδιασμού του έργου

Risk factor		Average Weighted response	Threshold	Accepted/rejected if < Threshold
R1	Unclear project scope to the manager and/or the other project members	4.08	3.41	Rejected
R2	Unrealistic project schedule	3.5	3.41	Rejected
R3	Unrealistic budget	2.73	3.41	Accepted
R4	Insufficient resources	3.10	3.41	Accepted

- **Φάση Ανάλυσης/Δημιουργίας προδιαγραφών συστήματος**

Σε αυτή τη φάση του έργου παρατηρήθηκαν οι περισσότεροι κίνδυνοι. Συγκεκριμένα, οι κίνδυνοι «**Ελλιπείς προδιαγραφές συστήματος (R6)**», «**Άγνοια των μη λειτουργικών προδιαγραφών (R7)**», «**Ανακριβείς απαιτήσεις/προδιαγραφές (R8)**», «**Μη τροποποιήσιμο έγγραφο προδιαγραφών (R12)**» και «**Το έγγραφο των προδιαγραφών δεν είναι σαφές από την ομάδα υλοποίησης (R13)**» έγιναν αποδεκτοί ως κίνδυνοι, ενώ οι «**Ασαφείς απαιτήσεις (R5)**», «**Μη επαληθεύσιμες απαιτήσεις (R9)**», «**Ασυνεπείς προδιαγραφές συστήματος (R10)**» και «**Ανέφικτες απαιτήσεις (R12)**» απορρίφθηκαν. Οι παράγοντες «**Ελλιπείς προδιαγραφές συστήματος**» και «**Ανακριβείς απαιτήσεις/προδιαγραφές**» θα μπορούσαν να συγχωνευθούν ως έναν κίνδυνο «**Μη ρεαλιστικές προδιαγραφές συστήματος**», αλλά για τις ανάγκες της έρευνας θεωρήθηκαν ως ξεχωριστοί παράγοντες κινδύνου.

Πίνακας 8: Δημοφιλείς παράγοντες κινδύνου κατά τη φάση της Ανάλυσης/Δημιουργίας προδιαγραφών συστήματος

Risk factor		Average Weighted response	Threshold	Accepted/rejected if < Threshold
R5	Unclear Requirements	4.18	3.58	Rejected
R6	Incomplete Requirements	2.88	3.58	Accepted
R7	Ignoring the Non-functional requirements	3.00	3.58	Accepted
R8	Inaccurate Requirements	3.50	3.58	Accepted

R9	Non-verifiable Requirements	3.67	3.58	Rejected
R10	Inconsistent Requirements	3.92	3.58	Rejected
R11	Infeasible Requirements	4.29	3.58	Rejected
R12	Non-modifiable Requirement Document	3.36	3.58	Accepted
R13	Requirements Document is not clear for developers	3.29	3.58	Accepted

- **Φάση Σχεδιασμού συστήματος**

Και σε αυτή τη φάση του κύκλου ζωής των έργων λογισμικού παρατηρείται εξίσου ο μεγάλος αριθμός κινδύνων. Φαίνεται πως όταν το έγγραφο του σχεδιασμού του συστήματος είναι ελλιπές, μεγάλο ή ασαφές μπορεί να αποτελέσει κίνδυνο για το έργο. Εδώ οι κίνδυνοι αναφέρονται ως «**Ελλιπές Σχεδιαστικό Έγγραφο (R18)**», «**Μεγάλο Σχεδιαστικό Έγγραφο (R19)**» και «**Ασαφές Σχεδιαστικό Έγγραφο (R20)**». Αντιθέτως, το περίπλοκο σύστημα ή σχέδιο, η μη διαθέσιμη τεχνογνωσία για την επαναχρησιμοποίηση του συστήματος, του κώδικα ή της αρχιτεκτονικής του συστήματος γενικότερα, καθώς και ο λανθασμένος σχεδιασμός/αρχιτεκτονική δεν αποτελούν κίνδυνο σύμφωνα με τις απαντήσεις των ερωτηθέντων. Συγκεκριμένα, οι παράγοντες «Πολύ περίπλοκο σύστημα (R14)», «Πολύπλοκο σχέδιο/αρχιτεκτονική συστήματος (R15)», «Μη διαθέσιμη τεχνογνωσία για επαναχρησιμοποίηση του κώδικα ή της αρχιτεκτονικής του συστήματος (R16)» και «Λανθασμένος σχεδιασμός/αρχιτεκτονική (R17)» απορρίφθηκαν ως κίνδυνοι.

Πίνακας 9: Δημοφιλείς παράγοντες κινδύνου κατά τη φάση του Σχεδιασμού του συστήματος

Risk factor		Average Weighted response	Threshold	Accepted/rejected if < Threshold
R14	Too complex system	3.57	3.54	Rejected
R15	Complicated Design	3.56	3.54	Rejected
R16	Unavailable expertise for reusability	3.67	3.54	Rejected
R17	Incorrect Design	4.13	3.54	Rejected
R18	Incomplete Design Document	2.94	3.54	Accepted
R19	Large Design Document	3.17	3.54	Accepted

R20	Unclear Document	Design	3.05	3.54	Accepted
-----	------------------	--------	------	------	----------

- **Φάση Υλοποίησης**

Σε αυτή τη φάση αναγνωρίστηκε μόνο ένας παράγοντας ως κίνδυνος. Αυτός είναι η «**Μεγάλη ποσότητα επαναλαμβανόμενου κώδικα (R22)**», ο οποίος ερμηνεύεται πως όταν ο κώδικας υλοποίησης επαναλαμβάνεται σε πολλά σημεία και για διαφορετικές λειτουργίες του συστήματος, μπορεί να προκαλέσει σφάλματα που να είναι δύσκολο να αναγνωρισθούν. Ο κίνδυνος αυτός έγινε αποδεκτός, ενώ οι «Ανάπτυξη κώδικα από την αρχή (R21)», «Μεγάλος και πολύπλοκος κώδικας υλοποίησης (R23)», «Πολλά συντακτικά σφάλματα/έλλειψη εμπειρίας ή τεχνογνωσίας ομάδας υλοποίησης (R24)», «Αλλαγές της τεχνολογίας (R25)» και «Κακός προγραμματισμός, δυσκολίες ή/και λάθη στην ενσωμάτωση του κώδικα (R26)» απορρίφθηκαν.

Πίνακας 10: Δημοφιλείς παράγοντες κινδύνου κατά τη φάση της Υλοποίησης

Risk factor		Average Weighted response	Threshold	Accepted/rejected if < Threshold
R21	Developing components from scratch	3.50	3.44	Rejected
R22	Large amount of repetitive code	2.94	3.44	Accepted
R23	Large and complex code	3.83	3.44	Rejected
R24	Too many syntax errors □ inexperienced developers/lack of technical knowledge	3.71	3.44	Rejected
R25	Technology changes	3.95	3.44	Rejected
R26	Poor integration plan/difficulties in integration/unclear integration plan	3.56	3.44	Rejected

- **Φάση Δοκιμών**

Όταν αλλαγές του κώδικα ενσωματώνονται στα ήδη υπάρχοντα κομμάτια κώδικα υλοποίησης μπορεί να προκύψουν σφάλματα. Εάν τα σφάλματα είναι πολλά μπορεί να προκαλέσουν κίνδυνο αποτυχίας του έργου και δείχνει έλλειψη ποιότητας των παραδοτέων. Αντίστοιχα, εάν υπάρχουν δυσκολίες στην επίλυση

σφαλμάτων λόγω έλλειψης τεχνογνωσίας της ομάδας υλοποίησης ή λόγω πολυπλοκότητας, αποτελεί εξίσου κίνδυνο για το έργο. Έτσι, οι παράγοντες «**Εμφάνιση σφαλμάτων κατά την ενσωμάτωση (R32)**» και «**Δυσκολίες στην επιδιόρθωση σφαλμάτων (R33)**» γίνονται αποδεκτοί ως κίνδυνοι. Εν αντιθέσει, οι «Οι δοκιμές δεν μπορούν να ανταποκριθούν στους στόχους ή τις απαιτήσεις του έργου (R27)», «Ο κώδικας υλοποίησης δεν είναι κατανοητός (R28)», «Έλλειψη αυτοματισμού δοκιμών (R29)», «Κακή συγγραφή των δοκιμαστικών περιπτώσεων (R30)», «Ανεπαρκείς δοκιμές αποδοχής (R31)», «Αδυναμία δοκιμών στο λειτουργικό περιβάλλον /Η τεχνολογία εργαλείων δοκιμών (προσομοίωση) δεν είναι διαθέσιμη (R34)» και «Το σύστημα δεν μπορεί να δοκιμαστεί επαρκώς (R35)» απορρίφθηκαν.

Πίνακας 11: Δημοφιλείς παράγοντες κινδύνου κατά τη φάση των Δοκιμών

Risk factor		Average Weighted response	Threshold	Accepted/rejected if < Threshold
R27	Testing cannot cope with project goals or requirements	4.33	3.23	Rejected
R28	Code is not understandable	4.37	3.23	Rejected
R29	Lack of testing automation□ boring and monotonous testing	3.42	3.23	Rejected
R30	Poor documentation of test cases	3.70	3.23	Rejected
R31	Inadequate acceptance testing	3.81	3.23	Rejected
R32	A lot of bugs emerged during the integration	2.79	3.23	Accepted
R33	Difficulties in repairing errors	2.05	3.23	Accepted
R34	Inability to test in the operational environment /Unit testing (simulation) technology unavailable	4.40	3.23	Rejected
R35	System is not enough testable	4.11	3.23	Rejected

- **Φάση Υποστήριξης**

Σε αυτή τη φάση δεν έχουν καταγραφεί πολλοί κίνδυνοι στη βιβλιογραφία. Οι ερωτηθέντες καλέστηκαν να επιλέξουν ανάμεσα σε δύο και αναγνωρίστηκε πως κατά την υποστήριξη, όταν προκύπτουν προβλήματα που καλούνται οι μηχανικοί

ανάπτυξης λογισμικού να επιλύσουν, όταν η περιγραφή του προβλήματος από τον πελάτη είναι ασαφής, μπορεί να προκαλέσει δυσκολίες στην ομάδα υλοποίησης ή και καμιά φορά αδυναμία επίλυσης. Για το λόγο αυτό γίνεται αποδεκτός ως κίνδυνος «**Ασαφής περιγραφή προβλημάτων (R36)**». Οι «Δυσκολίες στη συντήρηση του συστήματος (R37)» απορρίπτονται ως κίνδυνος.

Πίνακας 12: Δημοφιλείς παράγοντες κινδύνου κατά τη φάση της Υποστήριξης

Risk factor		Average Weighted response	Threshold	Accepted/rejected if < Threshold
R36	Software engineer cannot reproduce the problem/unclear problem description	2.61	3.04	Accepted
R37	Difficulties in maintainability	3.47	3.04	Rejected

- **Κίνδυνοι σε όλες τις φάσεις**

Οι παράγοντες κινδύνου που παρατηρούνται σε όλες τις φάσεις χωρίστηκαν σε κινδύνους που αφορούν τον άνθρωπο, τον πελάτη, την επιχείρηση και τη διαχείριση του έργου. Έτσι, αναγνωρίστηκαν ως κίνδυνοι που αφορούν τον άνθρωπο η «**Έλλειψη κινήτρων από την ομάδα (R38)**» και «**Η συχνή εναλλαγή των μελών της ομάδας (R41)**», καθώς η «**Έλλειψη γνώσης της ομάδας υλοποίησης (R39)**» και η «**Έλλειψη συνεργασίας (R40)**» απορρίφθηκαν.

Πίνακας 13: Δημοφιλείς παράγοντες κινδύνου σε όλες τις φάσεις που αφορούν τον άνθρωπο

Risk factor		Average Weighted response	Threshold	Accepted/rejected if < Threshold
R38	Lack of motivation of the team members	3.58	3.79	Accepted
R39	Lack of knowledge of the development team	4.11	3.79	Rejected
R40	Lack of cooperation	4.21	3.79	Rejected
R41	Team turnover	3.38	3.79	Accepted

Σε ότι αφορά τον πελάτη φαίνεται να αποτελεί κίνδυνο η «**Μεγάλη εμπλοκή του πελάτη (R42)**» στο έργο ενώ οι «Μη ρεαλιστικές προσδοκίες του πελάτη (R43)» απορρίφθηκαν. Ότι αφορά στην επιχείρηση, οι «**Δυσκολίες προϋπολογισμού από την επιχείρηση (R47)**», δείχνουν να αποτελούν κίνδυνο για τα έργα ανάπτυξης λογισμικού, ενώ αντίθετα, το γεγονός ότι «Οι πολιτικές πελατών-επιχειρήσεων δεν έχουν συμφωνηθεί/δεν κοινοποιούνται πλήρως εντός των οργανισμών (R48)» και οι «Αλλαγές στις κυβερνητικές πολιτικές (R49)» απορρίφθηκαν. Τέλος, για τη διαχείριση του έργου οι παράγοντες «Άπειρη διαχείριση έργου (R44)» και «Η πρόοδος του έργου δεν παρακολουθείται αρκετά στενά (R46)» απορρίφθηκαν, ενώ η «**Έλλειψη διαχείρισης και επικοινωνίας στην ομάδα από το διαχειριστή (R45)**» γίνεται αποδεκτή ως κίνδυνος.

Πίνακας 14: Δημοφιλείς παράγοντες κινδύνου σε όλες τις φάσεις που αφορούν τον πελάτη, την επιχείρηση και τη διαχείριση του έργου

Risk factor		Average Weighted response	Threshold	Accepted/rejected if < Threshold
R42	Too much user/customer involvement	2.97	3.40	Accepted
R43	Unrealistic customer/user expectations	3.83	3.40	Rejected
R44	Inexperienced project management	4.58	4.17	Rejected
R45	Lack of management and team communication	3.75	4.17	Accepted
R46	Project progress not monitored closely enough	4.25	4.17	Rejected
R47	Budget difficulties from business line	2.63	2.98	Accepted
R48	Customer-Business policies not agreed/not completely communicated within organizations	3.33	2.98	Rejected
R49	Change in government policies	3.30	2.98	Rejected

Συνεπώς, η έρευνα καταλήγει στους 19 πιο δημοφιλείς/κορυφαίους κινδύνους που παρατηρούνται στα έργα ανάπτυξης λογισμικού όπως παρουσιάζονται στον παρακάτω πίνακα. Παράλληλα, οι κίνδυνοι αυτοί κατηγοριοποιήθηκαν ανάλογα με τον τύπο κινδύνου που αντιπροσωπεύουν. Σημειώνεται ότι οι κίνδυνοι εμφανίζονται στον πίνακα με σειρά προτεραιότητας έπειτα από τον υπολογισμό της έκθεσης του έργου σε αυτούς:

Πίνακας 15: Κορυφαίοι 19 κίνδυνοι στα έργα ανάπτυξης λογισμικού σύμφωνα με την έρευνα

Παράγοντας κινδύνου	Φάση κύκλου του έργου (SDLC)	Τύπος Κινδύνου	
R8	Ανακριβείς απαιτήσεις/προδιαγραφές	Ανάλυση/Δημιουργία προδιαγραφών συστήματος	Ανάλυσης
R6	Ελλιπείς προδιαγραφές συστήματος	Ανάλυση/Δημιουργία προδιαγραφών συστήματος	Ανάλυσης
R41	Η συχνή εναλλαγή των μελών της ομάδας	Όλες οι φάσεις	Ανθρωπίνων πόρων
R3	Μη ρεαλιστικός προϋπολογισμός	Σχεδιασμός του έργου (Προγραμματισμός)	Οικονομικοί/Εκτίμησης έργου
R45	Έλλειψη διαχείρισης και επικοινωνίας στην ομάδα από το διαχειριστή	Όλες οι φάσεις	Διαχείρισης
R13	Το έγγραφο των προδιαγραφών δεν είναι σαφές από την ομάδα υλοποίησης	Ανάλυση/Δημιουργία προδιαγραφών	Καταγραφών (Documentation)
R20	Ασαφές Σχεδιαστικό Έγγραφο	Σχεδιασμός συστήματος	Καταγραφών (Documentation)
R18	Ελλιπές Σχεδιαστικό Έγγραφο	Σχεδιασμός συστήματος	Καταγραφών (Documentation)
R38	Έλλειψη κινήτρων από την ομάδα	Όλες οι φάσεις	Ανθρωπίνων πόρων
R22	Μεγάλη ποσότητα επαναλαμβανόμενου κώδικα	Υλοποίηση	Υλοποίησης
R33	Δυσκολίες στην επιδιόρθωση σφαλμάτων	Δοκιμές	Ποιότητας
R32	Εμφάνιση σφαλμάτων κατά την ενσωμάτωση	Δοκιμές	Ποιότητας
R4	Ανεπαρκείς πόροι	Σχεδιασμός του έργου	Λειτουργικοί/Οργανωτικοί

		(Προγραμματισμός)	
R47	Δυσκολίες προϋπολογισμού από την επιχείρηση	Όλες οι φάσεις	Στρατηγικοί
R36	Ασαφής περιγραφή προβλημάτων	Υποστήριξη	Ενσωμάτωσης
R7	Άγνοια των μη λειτουργικών προδιαγραφών	Ανάλυση/Δημιουργία προδιαγραφών	Ανάλυσης
R19	Μεγάλο Σχεδιαστικό Έγγραφο	Σχεδιασμός συστήματος	Καταγραφών (Documentation)
R12	Μη τροποποιήσιμο έγγραφο προδιαγραφών	Ανάλυση/Δημιουργία προδιαγραφών	Καταγραφών (Documentation)
R42	Μεγάλη εμπλοκή του πελάτη	Όλες οι φάσεις	Λειτουργικοί/Οργανωτικοί

6.3 ΠΙΝΑΚΑΣ ΚΙΝΔΥΝΟΥ

Αφού ολοκληρώθηκε η έρευνα και αναγνωρίστηκαν οι κορυφαίοι κίνδυνοι, δημιουργήθηκε ο πίνακας κινδύνου υπολογίζοντας την έκθεση καθενός από τους 19 κινδύνους ως το γινόμενο της πιθανότητας και της συνέπειας αυτών.

$$\text{Έκθεση} = \text{πιθανότητα} \times \text{συνέπεια}$$

Παράλληλα, αναγνωρίστηκε η σημαντικότητα αυτών, δηλαδή, αν η έκθεση είναι υψηλή, χαμηλή ή απλώς σημαντική. Τα αποτελέσματα του υπολογισμού καθώς και ο πίνακας κινδύνου παρουσιάζονται παρακάτω:

Πίνακας 16: Έκθεση των 19 κορυφαίων κινδύνων

Παράγοντας κινδύνου	Πιθανότητα	Συνέπεια	Έκθεση	Σημαντικότητα	
R8	Ανακριβείς απαιτήσεις/προδιαγραφές	0.7	0.8	0.56	Υψηλή
R6	Ελλιπείς προδιαγραφές συστήματος	0.7	0.8	0.56	Υψηλή
R4 1	Η συχνή εναλλαγή των μελών της ομάδας	0.5	0.8	0.4	Υψηλή
R3	Μη ρεαλιστικός	0.5	0.8	0.4	Υψηλή

	προϋπολογισμός				
R4 5	Έλλειψη διαχείρισης και επικοινωνίας στην ομάδα από το διαχειριστή	0.5	0.8	0.4	Υψηλή
R1 3	Το έγγραφο των προδιαγραφών δεν είναι σαφές από την ομάδα υλοποίησης	0.5	0.8	0.4	Υψηλή
R2 0	Ασαφές Σχεδιαστικό Έγγραφο	0.5	0.8	0.4	Υψηλή
R1 8	Ελλιπές Σχεδιαστικό Έγγραφο	0.5	0.8	0.4	Υψηλή
R3 8	Έλλειψη κινήτρων από την ομάδα	0.7	0.4	0.28	Υψηλή
R2 2	Μεγάλη ποσότητα επαναλαμβανόμενου κώδικα	0.5	0.4	0.2	Υψηλή
R3 3	Δυσκολίες στην επιδιόρθωση σφαλμάτων	0.5	0.4	0.2	Υψηλή
R3 2	Εμφάνιση σφαλμάτων κατά την ενσωμάτωση	0.7	0.2	0.14	Σημαντική
R4	Ανεπαρκείς πόροι	0.3	0.4	0.12	Σημαντική
R4 7	Δυσκολίες προϋπολογισμού από την επιχείρηση	0.3	0.4	0.12	Σημαντική
R3 6	Ασαφής περιγραφή προβλημάτων	0.5	0.2	0.1	Σημαντική
R7	Άγνοια των μη λειτουργικών προδιαγραφών	0.5	0.2	0.1	Σημαντική
R1 9	Μεγάλο Σχεδιαστικό Έγγραφο	0.5	0.2	0.1	Σημαντική
R1 2	Μη τροποποιήσιμο έγγραφο προδιαγραφών	0.5	0.2	0.1	Σημαντική
R4 2	Μεγάλη εμπλοκή του πελάτη	0.9	0.1	0.09	Σημαντική

Πίνακας 17: Πίνακας κινδύνου των 19 κορυφαίων κινδύνων

ΕΚΘΕΣΗ ΕΡΓΩΝ ΛΟΓΙΣΜΙΚΟΥ ΣΕ ΚΙΝΔΥΝΟ				
Πιθανότητα	R42			
			R32	R38
			R7,R12,R19, R36	R22,R33
		R49		R4
Συνέπεια				
	R8, R6	R3,R13,R18, R20,R41,R45		

Τέλος, για κάθε ένα κίνδυνο αναγνωρίσθηκε η στρατηγική του σχεδίου απόκρισης (αποφυγή, μετριασμός ή αποδοχή) που είναι απαραίτητο να ακολουθηθεί για την αντιμετώπισή του.

Πίνακας 18: Σχέδιο απόκρισης των 19 κορυφαίων κινδύνων

Παράγοντας κινδύνου	Σημαντικότητα	Σχέδιο Απόκρισης
R8 Ανακριβείς απαιτήσεις/προδιαγραφές	Υψηλή	Αποφυγή
R6 Ελλιπείς προδιαγραφές συστήματος	Υψηλή	Αποφυγή
R41 Η συχνή εναλλαγή των μελών της ομάδας	Υψηλή	Μετριασμός
R3 Μη ρεαλιστικός προϋπολογισμός	Υψηλή	Αποφυγή
R45 Έλλειψη διαχείρισης και επικοινωνίας στην ομάδα από το διαχειριστή	Υψηλή	Μετριασμός
R13 Το έγγραφο των προδιαγραφών δεν είναι σαφές από την ομάδα υλοποίησης	Υψηλή	Μετριασμός
R20 Ασαφές Σχεδιαστικό Έγγραφο	Υψηλή	Μετριασμός

R18	Ελλιπές Σχεδιαστικό Έγγραφο	Υψηλή	Αποφυγή
R38	Έλλειψη κινήτρων από την ομάδα	Υψηλή	Μετριασμός
R22	Μεγάλη ποσότητα επαναλαμβανόμενου κώδικα	Υψηλή	Μετριασμός
R33	Δυσκολίες στην επιδιόρθωση σφαλμάτων	Υψηλή	Αποφυγή
R32	Εμφάνιση σφαλμάτων κατά την ενσωμάτωση	Σημαντική	Μετριασμός
R4	Ανεπαρκείς πόροι	Σημαντική	Αποφυγή
R47	Δυσκολίες προϋπολογισμού από την επιχείρηση	Σημαντική	Αποφυγή
R36	Ασαφής περιγραφή προβλημάτων	Σημαντική	Αποδοχή
R7	Άγνοια των μη λειτουργικών προδιαγραφών	Σημαντική	Μετριασμός
R19	Μεγάλο Σχεδιαστικό Έγγραφο	Σημαντική	Μετριασμός
R12	Μη τροποποιήσιμο έγγραφο προδιαγραφών	Σημαντική	Αποδοχή
R42	Μεγάλη εμπλοκή του πελάτη	Σημαντική	Αποδοχή

ΚΕΦΑΛΑΙΟ 7 – ΣΥΜΠΕΡΑΣΜΑΤΑ

7.1 ΣΥΜΠΕΡΑΣΜΑΤΑ ΕΡΕΥΝΑΣ ΚΑΙ ΣΥΖΗΤΗΣΗ

Από την έρευνα που διεξήχθη προκύπτει πως ο μεγαλύτερος αριθμός κινδύνων παρατηρείται κατά τη φάση της ανάλυσης και δημιουργίας προδιαγραφών του κύκλου ζωής των έργων ανάπτυξης λογισμικού καθώς επίσης και κατά τη φάση του σχεδιασμού του συστήματος. Είναι εμφανές πως οι περισσότεροι τύποι κινδύνων σε αυτά τα στάδια του κύκλου έχουν να κάνουν με την καταγραφή (Documentation). Όταν μιλάμε για καταγραφή εννοούμε την καταγραφή των προδιαγραφών σε έγγραφα ή εργαλεία που χρησιμοποιεί η εκάστοτε επιχείρηση ή τα σχεδιαστικά έγγραφα. Όσο αφορά την ανάλυση, όταν το έγγραφο δεν είναι σαφές, καθαρώς γραμμένο, ελλιπές ή δύσκολο να τροποποιηθεί (συνήθως έγγραφο που παραλήφθηκε από προηγούμενη ομάδα για τη συνέχεια υλοποίησης του συστήματος), τότε η ομάδα υλοποίησης είναι δύσκολο να το ακολουθήσει και να οδηγηθεί σε σωστή υλοποίηση. Το ίδιο ισχύει και για τα σχεδιαστικά έγγραφα. Τα μεγάλα, ελλιπή ή ασαφή έγγραφα μπορεί να δημιουργήσουν σύγχυση και δύσκολη κατανόηση.

Είναι απαραίτητο, λοιπόν, η καταγραφές που γίνονται κατά τη διάρκεια του έργου να είναι ποιοτικές και να εμπεριέχουν όλα τα απαραίτητα στοιχεία ώστε η ομάδα υλοποίησης να κατανοεί πλήρως τι πρέπει να υλοποιηθεί. Έτσι, μπορούν να αποφευχθούν λάθη που θα επιφέρουν πολλά σφάλματα το παραδοτέο προϊόν και το έργο μπορεί να οδηγηθεί σε αποτυχία.

Ακόμη, ο ανθρώπινος παράγοντας και η επικοινωνία μεταξύ διαχειριστών και ομάδας χρήζουν προσοχής για την αποφυγή της αποτυχίας. Η συχνή εναλλαγή των μελών της ομάδας, η έλλειψη κινήτρων αλλά και η έλλειψη διαχείρισης και επικοινωνίας μεταξύ της ομάδας και του διαχειριστή βλέπουμε ότι βρίσκονται στους πιο σημαντικούς παράγοντες κινδύνου.

Ο βασικός ρόλος του διαχειριστή έργου είναι να καθοδηγήσει την ομάδα του και μαζί να οδηγήσουν το έργο στην επίτευξη των στόχων του. Είναι ο άνθρωπος που εμπνέει την ομάδα, που προωθεί την ομαδική εργασία και την συνεργασία. Οφείλει να εμπνέει σεβασμό και να κερδίζει με αυτόν τον τρόπο την αφοσίωση και το φιλότιμο της ομάδας. Πρέπει να υπάρχει διαφάνεια και ισονομία στις διαδικασίες που εκτελεί. Ένα ακόμη χαρακτηριστικό του διαχειριστή είναι να είναι ηθικός και να μην αμεροληπτεί. Επίσης, η ομάδα έργου που θα αναλάβει την περάτωση του, η βέλτιστη επικοινωνία-συνεργασία μεταξύ των μελών της και η σωστή στελέχωση της ομάδας, από ανθρώπους που έχουν τις απαραίτητες γνώσεις και εμπειρία πάνω στο αντικείμενο του έργου παίζουν πολύ σημαντικό ρόλο για την επιτυχία του έργου (Ζαχαροπούλου, 2021).

Μιλώντας για τις απαραίτητες γνώσεις που οφείλουν να διαθέτουν τα μέλη της ομάδας του έργου, ερχόμαστε αντιμέτωποι με τους αμέσως επόμενους σημαντικούς κινδύνους σύμφωνα με τα αποτελέσματα της έρευνας και του πίνακα κινδύνου. Αυτοί είναι η «μεγάλη ποσότητα επαναλαμβανόμενου κώδικα» και οι «δυσκολίες στην επιδιόρθωση σφαλμάτων». Και τα δύο είναι αποτέλεσμα της έλλειψης τεχνογνωσίας της ομάδας υλοποίησης. Συνεπώς, θα μπορούσαμε να συγχωνεύσουμε αυτούς τους δύο παράγοντες ως «Έλλειψη τεχνογνωσίας της ομάδας υλοποίησης», όμως συμβαίνουν σε διαφορετικό στάδιο του κύκλου ζωής του έργου λογισμικού ο καθένας. Άλλωστε, οι δυσκολίες στην επίλυση σφαλμάτων μπορεί να μην οφείλονται πάντα στην αυτού καθ' αυτού τεχνογνωσία της ομάδας αλλά και στην πολυπλοκότητα του ίδιου του συστήματος. Το ίδιο εξίσου μπορεί να συμβαίνει και με την ποσότητα του επαναλαμβανόμενου κώδικα, καθώς ναι μεν συμβαίνει από έλλειψη τεχνικής ή/και γνώσης του ατόμου που υλοποιεί τον κώδικα, αλλά επίσης μπορεί να οφείλεται στην πολυπλοκότητα. Συζητώντας, θα μπορούσαμε να πούμε ότι ο παράγοντας «Πολυπλοκότητα του συστήματος» θα μπορούσε να αποτελέσει κίνδυνο στα έργα λογισμικού.

Οι κίνδυνοι που αφορούν τον προϋπολογισμό του έργου παρατηρείται να είναι επίσης πολύ σημαντικοί. Το γεγονός ότι μπορεί να υπάρχουν δυσκολίες στον υπολογισμό του προϋπολογισμού ή οι υπολογισμοί μπορεί να καταλήξουν σε ένα μη ρεαλιστικό προϋπολογισμό του έργου, μπορεί να συμβαίνει λόγω εσφαλμένης εκτίμησης ή λανθασμένης στρατηγικής που ακολούθησε η επιχείρηση. Σε κάθε περίπτωση, ένα τέτοιο σφάλμα θα μπορούσε να προκαλέσει την αποτυχία του έργου.

Στη συνέχεια, παρατηρούμε πως κάθε κίνδυνος, ο οποίος δεν αφορά εσωτερικούς παράγοντες (όπως ομάδα υλοποίησης, ομάδα δημιουργίας προδιαγραφών, διαχειριστή, επιχείρηση κτλ.) αλλά εξωτερικούς (πελάτης, σφάλματα ή έγγραφα ή προδιαγραφές που έρχονται στην ομάδα με περιγραφή του πελάτη, λήψη ήδη υπάρχοντος εσφαλμένου κώδικα υλοποίησης ή εγγράφων από άλλη ομάδα κ.ά.) είναι δύσκολο να φέρουν κάποιο άλλο σχέδιο απόκρισης πέραν της αποδοχής. Εμπειρικά μπορούμε να πούμε πως αυτός δεν είναι ο κανόνας, αλλά η εξαίρεση λόγω των αποτελεσμάτων της συγκεκριμένης έρευνας.

Όσο αφορά το σχέδιο απόκρισης, για τους περισσότερους κινδύνους που αναγνωρίστηκαν στην έρευνα, το πιο σύνθηρες σχέδιο είναι αυτό του μετριασμού ενώ αμέσως μετά έρχεται η αποφυγή. Άλλωστε, ένας κίνδυνος στα έργα ανάπτυξης λογισμικού είναι δύσκολο να αποφευχθεί γι' αυτό η διαχείριση αυτού υπάρχει για να τον μετριάσει (Abdelrafe , Burairah, & Norhaziah, 2016).

7.2 ΣΥΓΚΡΙΣΗ ΕΡΕΥΝΑΣ ΜΕ ΤΗ ΒΙΒΛΙΟΓΡΑΦΙΑ

Σύμφωνα με την θεωρία που έχουμε μελετήσει, η μεθοδολογία και η υιοθέτηση της έρευνας συγκλίνει με τα αποτελέσματα των άλλων ερευνητών. Συγκεκριμένα, παρατηρείται πως οι παράγοντες που αφορούν τις ελλείψεις και ασαφείς προδιαγραφές, καθώς επίσης και την ασαφή καταγραφή του εγγράφου αυτών, εμφανίζονται ως κίνδυνοι από τους περισσότερους ερευνητές. Το ίδιο συμβαίνει και στον παράγοντα κινδύνου «Η συχνή εναλλαγή των μελών της ομάδας» και «Δυσκολίες στην επιδιόρθωση σφαλμάτων». Πολλοί ερευνητές στη βιβλιογραφία μιλούν για τους «Ανεπαρκείς πόρους» ως κίνδυνο στα έργα ανάπτυξης λογισμικού. Στη δική μας έρευνα ο συγκεκριμένος παράγοντας ναι μεν αναγνωρίστηκε ως κίνδυνος, όμως δεν κατατάσσεται στους πολύ σημαντικούς.

Εν αντιθέσει, οι κίνδυνοι «Έλλειψη κινήτρων από την ομάδα», «Ασαφής περιγραφή προβλημάτων», «Μη τροποποιήσιμο έγγραφο προδιαγραφών» και «Μεγάλη εμπλοκή του πελάτη» που αναγνωρίστηκαν στην έρευνα μας ως κορυφαίοι κίνδυνοι, δεν εμφανίζονται στη βιβλιογραφία ως κορυφαίοι από κανένα ερευνητή. Οι ερευνητές μιλούν πως είναι απαραίτητο να υπάρχει μεγάλη εμπλοκή του πελάτη στο έργο και ότι αν δεν εμπλέκεται συχνά ή καθόλου αυτό αποτελεί κίνδυνο. Ενώ, αντίθετα, στην έρευνά μας αναγνωρίστηκε πως η μεγάλη εμπλοκή του πελάτη μπορεί να δημιουργήσει κίνδυνο στο έργο.

Παρακάτω παρουσιάζεται ένας συγκριτικός πίνακας για το πόσοι και ποιοι από τους κινδύνους που αναγνωρίστηκαν σε αυτή την έρευνα συγκλίνουν ή αποκλίνουν από τη βιβλιογραφία:

Πίνακας 19: Συγκριτικός πίνακας κινδύνων έρευνας με τη βιβλιογραφία

Παράγοντας κινδύνου έρευνας	Βιβλιογραφία
R8 Ανακριβείς απαιτήσεις/προδιαγραφές	Menezes, Gusmão, & Moura, 2018 Abdelrafe , Burairah, & Norhaziah, 2016 Shahzad, 2012
R6 Ελλείψεις συστήματος προδιαγραφές	NOOR HABIBAH, AZLINAH, & ZAIHA, 2007 Menezes, Gusmão, & Moura, 2018 Abdelrafe , Burairah, & Norhaziah, 2016 Shahzad, 2012
R4 1 Η συχνή εναλλαγή των μελών της ομάδας	Enfei, 2015 Shahzad, 2012 NOOR HABIBAH, AZLINAH, & ZAIHA,

		2007 Bilal, Bashir, & Gani, 2020
R3	Μη ρεαλιστικός προϋπολογισμός	Menezes, Gusmão, & Moura, 2018
R4 5	Έλλειψη διαχείρισης και επικοινωνίας στην ομάδα από το διαχειριστή	NOOR HABIBAH, AZLINAH, & ZAIHA, 2007
R1 3	Το έγγραφο των προδιαγραφών δεν είναι σαφές από την ομάδα υλοποίησης	Menezes, Gusmão, & Moura, 2018 Enfei, 2015 Abdelrafe , Burairah, & Norhaziah, 2016 Shahzad, 2012
R2 0	Ασαφές Σχεδιαστικό Έγγραφο	Bilal, Bashir, & Gani, 2020
R1 8	Ελλιπές Σχεδιαστικό Έγγραφο	Bilal, Bashir, & Gani, 2020
R3 8	Έλλειψη κινήτρων από την ομάδα	-
R2 2	Μεγάλη ποσότητα επαναλαμβανόμενου κώδικα	Bilal, Bashir, & Gani, 2020
R3 3	Δυσκολίες στην επιδιόρθωση σφαλμάτων	Menezes, Gusmão, & Moura, 2018 Abdelrafe , Burairah, & Norhaziah, 2016 NOOR HABIBAH, AZLINAH, & ZAIHA, 2007 Bilal, Bashir, & Gani, 2020
R3 2	Εμφάνιση σφαλμάτων κατά την ενσωμάτωση	Abdelrafe , Burairah, & Norhaziah, 2016 Bilal, Bashir, & Gani, 2020
R4	Ανεπαρκείς πόροι	Menezes, Gusmão, & Moura, 2018 NOOR HABIBAH, AZLINAH, & ZAIHA, 2007 Bilal, Bashir, & Gani, 2020
R4 7	Δυσκολίες προϋπολογισμού από την επιχείρηση	Menezes, Gusmão, & Moura, 2018 Shahzad, 2012

R3 6	Άσαφής περιγραφή προβλημάτων	-
R7	Άγνοια των μη λειτουργικών προδιαγραφών	Bilal, Bashir, & Gani, 2020
R1 9	Μεγάλο Σχεδιαστικό Έγγραφο	Bilal, Bashir, & Gani, 2020
R1 2	Μη τροποποιήσιμο έγγραφο προδιαγραφών	-
R4 2	Μεγάλη εμπλοκή του πελάτη	-

Εν κατακλείδι και συνοψίζοντας όλα τα παραπάνω, είναι εμφανές πως τα αποτελέσματα αυτής της έρευνας είναι παρόμοια με τα αποτελέσματα των υπολοίπων ερευνών στη βιβλιογραφία, ενώ, κυρίως, συμπίπτουν στο γεγονός ότι και η συγκεκριμένη έρευνα αλλά και η βιβλιογραφία αποδεικνύουν πως οι περισσότεροι κίνδυνοι παρατηρούνται στη φάση της ανάλυσης και δημιουργίας προδιαγραφών του συστήματος κατά τον κύκλο ζωής ενός έργου ανάπτυξης λογισμικού.

BIBLIOΓΡΑΦΙΑ

- Radujkovića, M., & Sjekavicab, M. (2017). Project Management Success Factors. *Creative Construction Conference 2017*. Primosten, Croatia: Procedia Engineering 196 (2017) 607 – 615.
- Abdelrafe , E., Burairah, H., & Norhaziah, S. M. (2016). Top Fifty Software Risk Factors and the Best Thirty Risk Management Techniques in Software Development Lifecycle for Successful Software Projects. *International Journal of Hybrid Information Technology*, 9, 11-32.
- Arafeh, H., & El-Ahmad, A. (2017). The Influence of Software Risk Management on Software Project Success.
- Balachandra, R. (1984). Critical Signals for Making Go/NoGo Decisions in New Product Development. *Journal of Product Innovation Management*, Vol.2, pp. 92-100.
- Bilal, M., Bashir, N., & Gani, A. (2020). Risk Assessment Across Life Cycle Phases for Small and Medium Software Projects. *Journal of Engineering Science and Technology*, Vol. 15(No. 1 (2020)), 572 - 588.
- Enfei, L. (2015). Risk Factors of Software Development Projects in Chinese IT Small and Medium Sized Enterprises. Stockholm, Sweden.
- Hijazi, H., Alrainy, S., Muaidi, H., & Khmour, T. (2014). A Framework for Integrating Risk Management into the Software Development Process. *Research Journal of Applied Sciences, Engineering and Technology* 8(8), 919-928.
- Hijazi, H., Alrainy, S., Muaidi, H., & Khmour, T. (January 2014). RISK FACTORS IN SOFTWARE DEVELOPMENT PHASES. *European Scientific Journal*, vol.10.
- Hossain, M. I. (2023). Software Development Life Cycle (SDLC) Methodologies for Information Systems Project Management. *International Journal for Multidisciplinary Research (IJFMR)*, Volume 5(Issue 5).
- Jurison, J. (September, 1999). SOFTWARE PROJECT MANAGEMENT: THE MANAGER'S VIEW. *Communications of the Assosiation for Information Systems*, Vol. 2, Article 17.
- Lech, P. (2013). Time, budget, and functionality? IT project success criteria revised. *Information System Management*, 30 (3), pp 263–275.
- Lewis, J. P. (2010). Project planning, scheduling, and control: A hands-on guide to bringing projects in on time and on budget, Fifth Edition. New York: McGraw-Hill Education.
- Menezes, J. J., Gusmão, C., & Moura, H. (2018). Risk factors in software development projects: a systematic literature review. Springer Science+Business Media.

- Nguyen, D. S. (2016). Success Factors That Influence Agile Software Development Project Success. *American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)*, Volume 17, No 1, pp 172-222.
- NOOR HABIBAH, A., AZLINAH, M., & ZAIHA, M. (2007). Risk Factors in Software Development Projects. *Proceedings of the 6th WSEAS Int. Conf. on Software Engineering, Parallel and Distributed Systems*. Corfu Island, Greece.
- PMBOK. (2008). *A GUIDE TO THE PROJECT MANAGEMENT BODY OF LANGUAGE (PMBOK GUIDE) Fourth Edition*. Project Management Institute.
- Procaccino, J., & Verner, J. (2006). Software project managers and project success: An exploratory study. *The Journal of Systems and Software* 79 (2006), 1541–1551.
- Sahu, K., Pandey, R., & Kumar, R. (2014). Risk Management Perspective in SDLC. *International Journal of Advanced Research in Computer Science and Software Engineering* 4(3), pp. 1247-1251.
- Sankhwar, S., & Pandey, D. (2014). SOFTWARE PROJECT RISK ANALYSIS AND ASSESSMENT: A SURVEY. India: *Global Journal of Multidisciplinary Studies*, Volume 3, Issue 5, April 2014.
- Shahzad, B. (2012). Identification of Software Risk Factors in Large Scale Projects: A Quantitative Study. *Archives Des Sciences*, Vol 65(No. 12).
- Sujit Kumar, D., & Pushkar, D. (2013). SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC) ANALYTICAL COMPARISON AND SURVEY ON TRADITIONAL AND AGILE METHODOLOGY. *NATIONAL MONTHLY REFEREED JOURNAL OF RESEARCH IN SCIENCE & TECHNOLOGY, VOLUME NO.2(ISSUE NO.8)*.
- Turner, J. (1993). *THE HANDBOOK OF PROJECT-BASED MANAGEMENT*. The McGraw-Hill Companies, Inc., Third Edition.
- Van der Westhuizen, D., & Fitzgerald, E. (2005). Defining and measuring project success. *European Conference on IS Management, Leadership and Governance (ECMLG)*, (σσ. pp.1-17).
- Wideman, R. M. (2009). *First Principles of Project Management – Part 2*. Vancouver: AEW Services.
- Βάκκας, Β. (2020). Διπλωματική εργασία "Κριτήρια αξιολόγησης της Επιτυχίας έργων της Τεχνολογίας της Πληροφορίας". Πάτρα.
- Βικιπαίδεια. (χ.χ.). Ανάκτηση από https://el.wikipedia.org/wiki/Επτά_βασικά_εργαλεία_της_ποιότητας

- Βίτσιος, Α. (2018). ΔΙΑΧΕΙΡΙΣΗ ΚΙΝΔΥΝΟΥ ΣΕ ΚΑΤΑΣΚΕΥΑΣΤΙΚΑ ΕΡΓΑ ΔΗΜΟΣΙΟΥ ΚΑΙ ΙΔΙΩΤΙΚΟΥ ΔΙΚΑΙΟΥ. Αθήνα.
- Ζαχαροπούλου, Χ. (2021, Φεβρουάριος). Η ΔΙΟΙΚΗΣΗ ΕΡΓΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΣΤΟ ΔΗΜΟΣΙΟ ΤΟΜΕΑ ΚΑΙ ΣΤΗΝ ΑΝΕΞΑΡΤΗΤΗ ΑΡΧΗ ΔΗΜΟΣΙΩΝ ΕΣΟΔΩΝ. Αθήνα.
- Κολιοφώτης, Γ. (2017). ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ "ΕΡΓΑΛΕΙΑ ΒΕΛΤΙΩΣΗΣ ΠΟΙΟΤΗΤΑΣ ΚΑΙ ΔΙΑΧΕΙΡΙΣΗΣ ΚΙΝΔΥΝΟΥ ΣΕ ΒΙΟΜΗΧΑΝΙΚΗ ΚΑΙ ΕΜΠΟΡΙΚΗ ΕΠΙΧΕΙΡΗΣΗ". Πειραιάς.
- Μαρκάκη, Δ. Μ. (2021). Σημειώσεις μαθήματος "Διοίκηση Έργου", MBA για Μηχανικούς. Ελληνικό Μεσογειακό Πανεπιστήμιο.
- Νταρζάνου, Ν. (2012). Διπλωματική εργασία, «Μελέτη παραγόντων Risk Management για Ανάπτυξη και Εφαρμογή ενός Πληροφοριακού Συστήματος».
- Παπαδημητρίου, Α. (2009). Μεταπτυχιακή Διατριβή, "Διαχείριση Κινδύνων σε έργα Πληροφορικής". Πειραιάς.
- Σιώκου, Α. (2021). Διπλωματική εργασία, Η ΔΙΑΧΕΙΡΙΣΗ ΚΙΝΔΥΝΩΝ ΣΤΟ ΡΜΒΟΚ. Αθήνα.
- Τσαλαμάνδρης, Α. (2014). Risk management. *Διπλωματική εργασία*. Πειραιάς.
- Φιτσιλής, Π. (2015). Εισαγωγή στη διαχείριση έργων. *Σύγχρονα πληροφοριακά συστήματα επιχειρήσεων*. Κάλλιπος, Ανοικτές Ακαδημαϊκές Εκδόσεις.

ΠΑΡΑΡΤΗΜΑ

Στο παράρτημα αυτό παρουσιάζεται το ερωτηματολόγιο που χρησιμοποιήθηκε για την έρευνα. Αρχικά εξηγεί στους χρήστες πως θα διεξαχθεί η έρευνα και δίνει οδηγίες για τη συμπλήρωσή του. Καθώς δεν είναι δυνατό όλες οι ειδικότητες και οι ρόλοι να απαντήσουν σε όλες τις ερωτήσεις, δεν ήταν υποχρεωτικές οι απαντήσεις σε όλες τις ερωτήσεις για να προχωρήσει η έρευνα. Το ερωτηματολόγιο δημιουργήθηκε και πρωθθήθηκε στην αγγλική γλώσσα:

Risk Management in Software Development Projects

Your answers to this questionnaire will help us to investigate the main factors that cause Software Development Projects to fail. It will only take 10 minutes and it is anonymous. You have to answer to 80 questions.

The questions are multiple choice, small description or ranking questions. You will first be asked to fill in your demographic information. After that, the questionnaire is structured into sections. Each section represents a stage of the Software Development Life Cycle (SDLC). You have to push the "Next" button to go to the next section. You will not be asked technical questions, just answer through your experience. Your answers may be based on a project you are currently working on or have worked on in the past. Please be honest and answer all questions to complete the entire questionnaire.

Thank you very much for your contribution!

Demographic Data:

Q1: Which is your gender?

-Female -Male -Prefer not to say

Q2: Which is your educational status?

(Select one of the answers or write another educational status in the last bullet)

-High school or below -Diploma -Bachelor -Master -PhD -
Other...

Q3: Which is your education background?

(Select one of the answers or write another education background in the last bullet)

-Computer Science -Information Technology/Informatics -Software engineering
 -Electrical Engineering -Mathematics -Business Administration
-Other...

Q4: How many years of work experience do you have?

-Less than one year -1-3 years -3-5 years -More than 5 years

Q5: Which is your job title?

(Select one of the answers or write another job title in the last bullet)

- Requirements Engineer
- System architect
- Software Engineer/Developer
- Test Engineer
- Project Manager
- Department Leader or above
- Other...

Q6: What industry does the company you work for belong to?

(Write or describe in one or two words e.g. Financial Industry)

Q7: How many employees does the company you work for have?

- Less than 10
- 10-50
- 50-100
- 100-500
- 500-1000
- more than 1000
- I don't know

Project Planning:

Q1: Is the project scope (e.g. size, goals, requirements etc) clear to the project members?

- No
- To some extent
- Yes

Q2: Do you believe that the project schedule (completion time or deadlines) is realistic?

- No
- To some extent
- Yes

Q3: How much realistic you find it? (Please scale your answer from 1 as not at all to 5 as very realistic)

1 not at all 5 very realistic

Q4: Which is the balance between time and effort of the project?

- Very little time and too much effort
- Time and effort are balanced
- Enough effort and very much time

Q5: How well is estimated the cost of the project? (Please scale your answer from 1 as very bad to 5 as very well)

1 very bad 5 very well

Q6: Is the number of the project members sufficient?

- No, more members are required
- Yes
- Too many members within the project

Q7: Is the technology that is used for the project (e.g. tools) updated?

- New technology is required
- It is always updated
- Enough for the project needs

Q8: From 1 to 5, how sufficient is the estimation of the required resources (people, tools, technologies)?

1 Not sufficient 5 Sufficient

Analysis (Requirements):

Q9: Do the development teams understand the requirements?

-No -To some extent -Yes

Q10: Do the requirements change during the Software Development Lifecycle (SDLC)?

-No -Partially -Yes

Q11: Until now, how much percent of the requirements changed over a given period of the time?

-High, e.g., $\geq 70\%$ -Medium, e.g., $\geq 30\%$ - Low, e.g., $\leq 30\%$

Q12: Are the system requirements (e.g. functional, quality, architectural and operational) completely identified?

-Partially -Yes, but not all -Completely

Q13: Do the requirements reflect the real user needs?

-No -To some extent -Yes

Q14: Are the user requirements (e.g. use cases, scenarios & features) properly specified?

-No -Partially -Yes

Q15: Is there a finite cost effective process that verifies the requirements (e.g. testing, inspection, demonstration or analysis)?

-No -Yes -Maybe

Q16: Are requirements within the project scope?

-No -To some extent -Yes

Q17: Can the requirements be implemented within the constraints of the project (e.g. using the existing tools, technologies etc)?

-No -To some extent -Yes

Q18: Is there any requirement which is not possible to implement within existing technology?

-Yes, many -Some -None

Q19: Is the Requirements Documentation structured in order to be easy modified?

-No -Partially -Yes

Design:

Q20: Is the Requirements Documentation clear to the development team?

-No -To some extent -Yes

Q21: Based on the architecture, how large is the software system to be developed?
(Please scale your answer from 1 as not large to 5 as very large)

1 not large 5 very large

Q22: Based on the architecture, how complex is the software system to get developed? (Please scale your answer from 1 as not complex to 5 as very complex)

1 not complex 5 very complex

Q23: Based on the UI/UX Design, how complex is the software system to get developed? (Please answer the question only if your project has to do with UI/UX)
(Please scale your answer from 1 as not complex to 5 as very complex)

1 not complex 5 very complex

Q24: Is the development team experienced in maintaining old components of code in order to reuse them?

-No -To some extent -Yes

Q25: Does the architectural design meet the system requirements?

-No -Partially -Yes

Q26: Does the UI/UX design meet the system requirements? (Please answer the question only if your project has to do with UI/UX)

-No -Partially -Yes

Q27: Does the architectural design document contain all the details to be implemented?

-No -Partially -Yes

Q28: Does the UI/UX design document contain all the details to be implemented?
(Please answer the question only if your project has to do with UI/UX)

-No -Partially -Yes

Q29: How large is the architectural design document? (Please scale your answer from 1 as not large to 5 as very large)

1 not large 5 very large

Q30: Is the architectural design document clearly specified? (Please scale your answer from 1 as no to 5 as completely)

1 no 5 completely

Q31: Is the UI/UX documentation clearly specified? (Please answer the question only if your project has to do with UI/UX) (Please scale your answer from 1 as no to 5 as completely)

1 no 5 completely

Implementation:

Q32: How much percent of the code components are reusable?

-Many, e.g., $\geq 70\%$ -Some, e.g., $\geq 30\%$ - Few, e.g., $\leq 30\%$

Q33: If there are repetitive parts of the code, is the process of the repetition automatic?

-No -To some extent -Yes

Q34: How much percent of the code parts are repetitive?

-Many, e.g., $\geq 70\%$ -Some, e.g., $\geq 30\%$ - Few, e.g., $\leq 30\%$

Q35: How large is the code to be developed? (Please scale your answer from 1 as not large to 5 as very large)

1 not large 5 very large

Q36: How complex is the code to be developed? (Please scale your answer from 1 as not complex to 5 as very complex)

1 not complex 5 very complex

Q37: How much percent of the code size may contain syntax errors?

-Large, e.g., $\geq 70\%$ -Some parts, e.g., $\geq 30\%$ - Small, e.g., $\leq 30\%$

Q38: Does the development team easily adapt to the technology changes?

-No -To some extent -Yes

Q39: Is the integration ordering clear to the development team?

-No -To some extent -Yes

Testing:

Q40: Does the test plan consider project goals and requirements?

-No -Partially -Yes

Q41: Is the code understandable by the reviewers?

-No -To some extent -Yes

Q42: Are there automated tools for the unit testing?

-No -To some extent -Yes

Q43: Are the test cases specified?

-No -To some extent -Yes

Q44: How clearly are the test cases specified? (Please scale your answer from 1 as not clearly specified to 5 as very clearly specified)

1 not clearly specified 5 very clearly specified

Q45: Does the project consider adequate acceptance testing before the delivery of the product?

-No -To some extent -Yes

Q46: How much percent of bugs are emerged during the integration?

-Many, e.g., $\geq 70\%$ -Some, e.g., $\geq 30\%$ - Few, e.g., $\leq 30\%$

Q47: How often are other modules affected by the fixed bugs?

-Rarely -Often -Always

Q48: How much percent of the fixed bugs affect other modules?

-Many, e.g., $\geq 70\%$ -Some, e.g., $\geq 30\%$ - Few, e.g., $\leq 30\%$

Q49: Is a real testing environment available?

-No -Simulation is enough -Yes

Q50: Do the quality assurance principles applied properly?

-No -Partially -Yes

Maintenance:

Q51: How much percent of the bugs reported by customers are not reproducible (if any)?

-Many, e.g., $\geq 70\%$ -Some, e.g., $\geq 30\%$ - Few, e.g., $\leq 30\%$

Q52: How clear is the problem description of the bugs reported by customers? (Please scale your answer from 1 as not clear to 5 as very clear)

1 not clear 5 very clear

Q53: Is there any emergency release necessary during the maintenance?

-No -Partially -Yes

Risks at all phases:

Human

Q54: Are the project members motivated and committed to the project?

-Some of them -Almost all -Yes

Q55: How is the overall relevant domain knowledge of the development team?

-Not much -Less than adequate -Adequate

Q56: Are the project members already trained and experienced with tools, language and development environment?

-Some of them -Most of them -All

Q57: How would you evaluate the overall development team performance?

-Poor with frequent unhealthy arguments -Average and healthy arguments
-Satisfactory -Very good

Q58: How often a team member within your team leaves its job because of better vacancies?

-Rarely -Often -Very often

Customer/User

Q59: What is the level of involvement of customer / user until now?

-Passive -Occasional -Active

Q60: Does the development team face problem due to large number of users' involvement?

-No -To some extent -Yes

Q61: Do you think customers/ users have realistic expectations about the project?

- No -To some extent -Yes

Management/Business

Q62: How much capable is the project manager to lead the project successfully?

-Unreliable -Partially reliable -Reliable

Q63: How is the confidence of development team on project management?

-Low -Medium -More than medium

Q64: Does the project manager communicate critical problems with the development team?

-No -Sometimes -Yes

Q65: Is project management monitoring the project closely enough?

- No -To some extent -Yes

Q66: Does the business management agree to trade additional budget and adequate support for handling project risks?

-No -Not willingly -Yes

Q67: Are the policies and procedure approved and implemented within the organizations?

-Some policies exist but not implemented
implemented -Yes, completely

-Policies exist & partially

Q68: Do the government policies affect the project?

- No -To some extent -Yes

THANK YOU FOR YOUR PARTICIPATION!! 😊