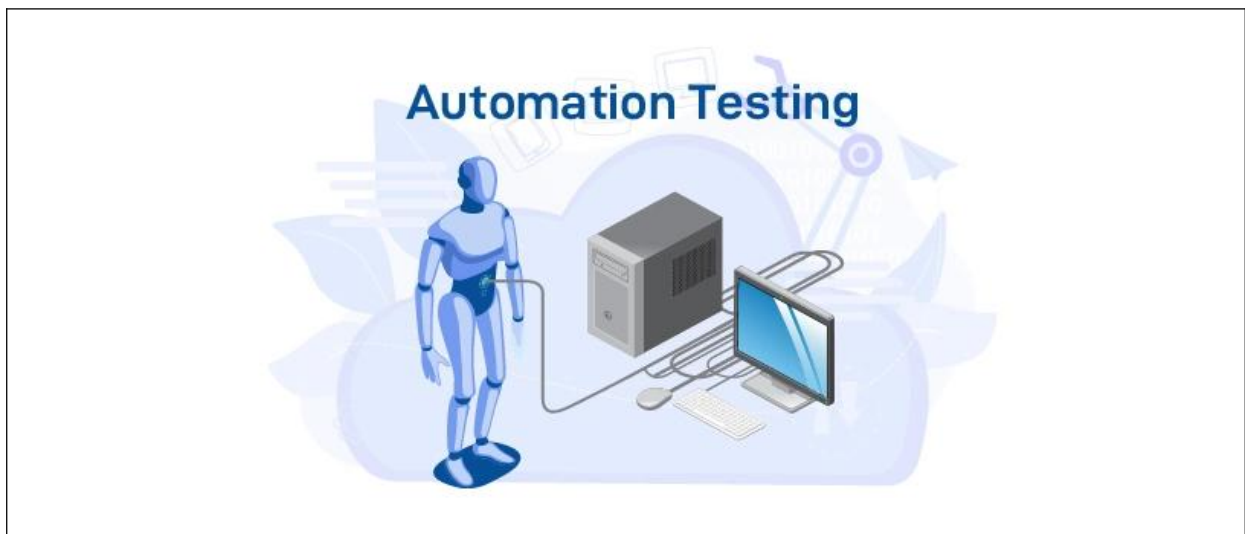




ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
«Αυτόματος Έλεγχος Κώδικα»



Του φοιτητή
Τράιτη Θεόδωρου
Αρ. Μητρώου: 174894

Επιβλέπουσα
Ελβίρα Μαρία Αρβανίτου

Θεσσαλονίκη 2023

Τίτλος Δ.Ε. Αυτόματος Έλεγχος Κώδικα

Κωδικός Δ.Ε. 22348

Όνοματεπώνυμο φοιτητή Θεόδωρος Τράιτσης

Όνοματεπώνυμο εισηγητή Ελβίρα-Μαρία Αρβανίτου

Ημερομηνία ανάληψης Δ.Ε. 29/11/2022

Ημερομηνία περάτωσης Δ.Ε. 10/09/2023

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Τράιτση Θεόδωρου που την εκπόνησε/αν. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

«Αφιερώνω αυτήν τη διπλωματική εργασία στην οικογένειά μου και σε όλους όσους με στήριξαν κατά τη διάρκεια αυτής της πορείας εκπαίδευσης.»

Πρόλογος

Η επιστήμη της πληροφορικής εξελίσσεται με αστραπιαίους ρυθμούς, με την τεχνολογία του λογισμικού να αποτελεί κινητήρια δύναμη στον ψηφιακό μας κόσμο. Καθώς η εξάρτησή μας από το λογισμικό αυξάνεται, το ζήτημα της αξιοπιστίας και της ποιότητας του καθίσταται κρίσιμο. Εμπνευσμένος από αυτήν την ανάγκη, επέλεξα να αφιερώσω τη διπλωματική μου στον "Αυτόματο Έλεγχο Κώδικα". Ο λόγος είναι απλός: ο επαναστατικός ρόλος της αυτοματοποίησης στην αξιολόγηση του λογισμικού.

Αυτή η διαδικασία δεν μόνο εξασφαλίζει τη σωστή λειτουργία του κώδικα αλλά επίσης ελαχιστοποιεί τον ανθρώπινο παράγοντα στον έλεγχο και την ανίχνευση σφαλμάτων. Στόχος μου είναι να ανακαλύψω το εύρος των οφελών που μπορεί να φέρει ο αυτόματος έλεγχος κώδικα και πώς επηρεάζει την ποιότητα του λογισμικού. Μέσα από αυτήν την έρευνα, ελπίζω να προσφέρω στην πρόοδο της πληροφορικής και να συνεισφέρω στην ανάπτυξη λογισμικού υψηλής ποιότητας που θα εξυπηρετεί τις ανάγκες του αύριο. Αυτή η διπλωματική αποτελεί μια ευκαιρία για εμένα να αναδείξω την αγάπη μου για την τεχνολογία και να εξερευνήσω τον επαναστατικό ρόλο του αυτοματοποιημένου ελέγχου στην ανάπτυξη λογισμικού.

Περίληψη

Η διπλωματική εργασία με τίτλο "Αυτοματοποιημένος Έλεγχος Κώδικα" εξετάζει τη σημασία του αυτοματοποιημένου ελέγχου λογισμικού και τη συμβολή του στην επιτυχία των έργων ανάπτυξης λογισμικού. Παρουσιάζονται οι διάφορες κατηγορίες ελέγχου λογισμικού, με έμφαση στον αυτοματοποιημένο έλεγχο, και εξετάζονται οι διαφορές μεταξύ αυτού και του χειροκίνητου ελέγχου. Στη συνέχεια, παρουσιάζεται η βιβλιογραφική ανασκόπηση σχετικά με τα πλαίσια λογισμικού και τα εργαλεία αυτοματοποιημένου ελέγχου. Ακολουθεί μια ανάλυση της εφαρμογής του αυτοματοποιημένου ελέγχου σε μια συγκεκριμένη βιομηχανία, παρουσιάζοντας τον οργανισμό, τη διαδικασία ανάπτυξης λογισμικού και την περιγραφή του εμπλεκόμενου λογισμικού. Έπειτα, εξετάζεται η αρχιτεκτονική του εργαλείου αυτοματοποιημένου ελέγχου Learwork και παρουσιάζεται η διαδικασία εκτέλεσης ελέγχων, αναφοράς αποτελεσμάτων αλλά και η διαδικασία παραγωγής αυτοματοποιημένων ελέγχων. Στόχος της συγκεκριμένης διπλωματικής εργασίας είναι να εξερευνήσουμε αν οι αυτοματοποιημένοι έλεγχοι βοηθάνε πραγματικά τους προγραμματιστές σε διάφορους τομείς όπως για παράδειγμα στη παραγωγικότητα, αποδοτικότητα και μείωση των σφαλμάτων. Για την αξιολόγηση των αυτοματοποιημένων ελέγχων, δημιουργήθηκε ένα ερωτηματολόγιο το οποίο διανεμήθηκε σε 10 υπαλλήλους του οργανισμού. Τα αποτελέσματα είναι σημαντικά για τον οργανισμό, καθώς υποστηρίζουν ότι οι αυτοματοποιημένοι έλεγχοι αποτελούν μια αποτελεσματική πρακτική για τη βελτίωση της διαδικασίας ανάπτυξης λογισμικού και την εξοικονόμηση χρόνου και πόρων, καθώς και για την ενίσχυση της ποιότητας και της εμπιστοσύνης στο προϊόν.

«Αυτόματος Έλεγχος Κώδικα»

(Automation Testing)

«Traitsis Theodoros»

Abstract

The thesis "Test Automation" examines the importance of software testing and its contribution to the success of software development projects. The various categories of software testing are presented, with an emphasis on automation testing, and the differences between it and manual testing are discussed. The literature review on software frameworks and automation testing tools is then presented. This is followed by an analysis of the application of automation tests in a specific industry, presenting the organization, the software development process and a description of the software involved. Then, the architecture of the Leapwork automation testing tool is discussed and the process of performing tests, reporting results and the process of producing automation tests is presented. Furthermore, the research question "Do automation testing helps the developer?" is presented, trying to find an answer through an industrial evaluation, for which the results and conclusions of the research are presented. Finally, future research that may result from this thesis is discussed. The aim of this thesis is to explore whether automation testing really help programmers in various areas such as productivity, efficiency and error reduction. To evaluate the automation testing, a questionnaire was created and distributed to 10 employees of the organization. The results are important to the organization as they support that automation testing is an effective practice for improving the software development process and saving time and resources, as well as enhancing quality and confidence in the product.

Ευχαριστίες

Θα ήθελα θερμά να ευχαριστήσω όλους εκείνους που συνέβαλαν στην ολοκλήρωση αυτής της διπλωματικής εργασίας. Η υποστήριξή σας, είτε επιστημονική, ηθική, ή οικονομική, ήταν κρίσιμη για την επιτυχή ολοκλήρωσή της. Πρώτα και κύρια, θέλω να ευχαριστήσω την επιβλέπουσα καθηγήτρια, κ.Ελβίρα Μαρία Αρβανίτου, για την εμπειρογνομosύνη, την υποστήριξη και την καθοδήγησή της κατά τη διάρκεια αυτής της ερευνητικής διαδρομής. Επίσης, θέλω να εκφράσω τις ειλικρινές μου ευχαριστίες προς τους φίλους και την οικογένειά μου που με υποστήριξαν και με ενέπνευσαν καθ' όλη τη διάρκεια αυτής της πορείας. Τέλος, εκφράζω την ευγνωμοσύνη μου προς όλους τους συναδέλφους και τους συνεργάτες που μοιράστηκαν τις γνώσεις, τις ιδέες και την εμπειρία τους μαζί μου. Η συμβολή σας είναι ανεκτίμητη και συνέβαλε σημαντικά στην επιτυχία αυτής της διπλωματικής εργασίας. Ευχαριστώ θερμά για όλη τη στήριξη και τη βοήθειά σας.

Περιεχόμενα

Πρόλογος.....	v
Περίληψη.....	vi
Abstract.....	vii
Ευχαριστίες.....	viii
Περιεχόμενα.....	ix
Κατάλογος Σχημάτων.....	x
Κατάλογος Πινάκων.....	xi
Συνομογραφίες.....	xii
Κεφάλαιο 1ο: Εισαγωγή.....	1
1.1 Σημαντικότητα του Ελέγχου.....	2
1.2 Η Συμβολή του Ελέγχου για την Επιτυχία.....	3
1.3 Ο έλεγχος λογισμικού αποτελεί βασική πτυχή της ποιότητας.....	3
1.4 Κατηγορίες Ελέγχου Λογισμικού.....	4
1.4.1 Χειροκίνητος Έλεγχος.....	5
1.4.2 Αυτοματοποιημένος Έλεγχος.....	10
1.5 Διαφορές Χειροκίνητου και Αυτοματοποιημένου Ελέγχου.....	13
1.6 Κίνητρο και Στόχος Διπλωματικής.....	15
1.6.1 Χρησιμότητα του Αυτοματοποιημένου Ελέγχου.....	15
Κεφάλαιο 2ο: Βιβλιογραφική Ανασκόπηση.....	18
2.1 Πλαίσια Λογισμικού και Εργαλεία Αυτοματοποιημένου Ελέγχου.....	18
2.2 Σχετική Βιβλιογραφία.....	20
Κεφάλαιο 3ο: Εφαρμογή Αυτοματοποιημένου Ελέγχου στη Βιομηχανία.....	24
3.1 Οργανόγραμμα Εμπλεκόμενης Βιομηχανίας.....	24
3.2 Διαδικασία Ανάπτυξης Λογισμικού.....	25
3.3 Περιγραφή Εμπλεκόμενου Λογισμικού.....	27
3.3.1 Λειτουργίες Εμπλεκόμενου Λογισμικού.....	30
3.4 Περιγραφή Εργαλείου.....	37
3.4.1 Αρχιτεκτονική.....	37
3.4.2 Building Blocks.....	39
3.4.3 Ροή Αυτοματοποιημένου Ελέγχου.....	40
3.4.4 Εκτέλεση Ελέγχων και Αναφορά αποτελεσμάτων.....	42
3.5 Αυτοματοποιημένοι Έλεγχοι που παρήχθησαν.....	44
Κεφάλαιο 4ο: Βιομηχανική Αξιολόγηση.....	58
4.1 Ερευνητικό Ερώτημα.....	58
4.2 Οργάνωση Βιομηχανικής Αξιολόγησης και Συλλογή Δεδομένων.....	58
Κεφάλαιο 5ο: Αποτελέσματα και Συμπεράσματα.....	60
5.1 Αποτελέσματα.....	60
5.2 Συμπεράσματα.....	69
5.3 Μελλοντική Έρευνα.....	70
Βιβλιογραφία.....	72

Κατάλογος Σχημάτων

Σχήμα 3.3α: Προθάλαμος εφαρμογής Adminkit	28
Σχήμα 3.3β: Εγγραφή χρήστη	28
Σχήμα 3.3γ: Εγγραφή εταιρείας	29
Σχήμα 3.3δ: Μήνυμα καλωσορίσματος	29
Σχήμα 3.3ε: Αρχική σελίδα Adminkit	30
Σχήμα 3.3.1.1: Σελίδα Employee	31
Σχήμα 3.3.1.2: Σελίδα Vacation & Leave	31
Σχήμα 3.3.1.3: Σελίδα Employment Contracts	32
Σχήμα 3.3.1.4: Σελίδα Shared Documents	32
Σχήμα 3.3.1.5: Σελίδα Digital signature	33
Σχήμα 3.3.1.6: Σελίδα Contracts	34
Σχήμα 3.3.1.7: Σελίδα Text Messages	34
Σχήμα 3.3.1.8: Σελίδα GDPR	35
Σχήμα 3.3.1.9: Σελίδα Access Control	36
Σχήμα 3.3.1.10: Σελίδα Lottery	36
Σχήμα 3.4.3α: Καμβάς σχεδίασης στο Leawork	41
Σχήμα 3.4.3β: Λίστα με όλες τις κατηγορίες των μπλοκ συναρμολόγησης	41
Σχήμα 3.4.4α: Run List	43
Σχήμα 3.4.4β: Reporting	44
Σχήμα 3.5α: Login and Logou	45
Σχήμα 3.5β: StartBrowserAndLogin_SF	46
Σχήμα 3.5γ: Logout_SF	46
Σχήμα 3.5δ: Employee Correct Features	47
Σχήμα 3.5ε: EmployeeFeatures_SF	47
Σχήμα 3.5στ: EmployeeFeatures_SF	48
Σχήμα 3.5ζ: Invite Non-Existing User	48
Σχήμα 3.5η: Invite Non-Existing User	49
Σχήμα 3.5θ: EmailGenerator_SF	49
Σχήμα 3.5ι: PrivateEmail_SF	50
Σχήμα 3.5ια: EmailSearchInvite_SF	50
Σχήμα 3.5ιβ: Employee can't Edit Organisation	51
Σχήμα 3.5ιγ: Το κείμενο 'adminkit-leapwork' από την σκοπιά του Chrome DevTools	51
Σχήμα 3.5ιδ: Επιλογή 'Edit Web Element' στο μπλοκ 'Get Web Text	52
Σχήμα 3.5ιε: Upload a Document	53
Σχήμα 3.5ιστ: Upload Invalid Files	54
Σχήμα 3.5ιζ: Access level	54
Σχήμα 3.5ιη: Access level	55
Σχήμα 3.5ιθ: Transfer to Next Year as Admin	56
Σχήμα 3.5κ: Βοηθητικό παράθυρο 'Transfer vacation days	56
Σχήμα 5.1α: Ειδικότητα των συμμετοχόντων της έρευνας	60
Σχήμα 5.1β: Γράφημα 1ης ερώτησης πολλαπλής επιλογής	61
Σχήμα 5.1γ: Γράφημα 2ης ερώτησης πολλαπλής επιλογής	61
Σχήμα 5.1δ: Γράφημα 3ης ερώτησης πολλαπλής επιλογής	62
Σχήμα 5.1ε: Γράφημα 4ης ερώτησης πολλαπλής επιλογής	62

Σχήμα 5.1στ: Γράφημα 5ης ερώτησης πολλαπλής επιλογής.....	63
Σχήμα 5.1ζ: Γράφημα 6ης ερώτησης πολλαπλής επιλογής.....	63
Σχήμα 5.1η: Γράφημα 7ης ερώτησης πολλαπλής επιλογής.....	63
Σχήμα 5.1θ: Γράφημα 8ης ερώτησης πολλαπλής επιλογής.....	64
Σχήμα 5.1ι: Γράφημα 9ης ερώτησης πολλαπλής επιλογής.....	65
Σχήμα 5.1ια: Γράφημα 10ης ερώτησης πολλαπλής επιλογής.....	65

Κατάλογος Πινάκων

Πίνακας 1.5: Διαφορές χειροκίνητου και αυτοματοποιημένου ελέγχου.....	13
Πίνακας 5.1α: Απαντήσεις στην 1 ^η ερώτηση ανάπτυξης.....	66
Πίνακας 5.1β: Απαντήσεις στην 2 ^η ερώτηση ανάπτυξης.....	67
Πίνακας 5.1γ: Απαντήσεις στην 3 ^η ερώτηση ανάπτυξης.....	67
Πίνακας 5.1δ: Απαντήσεις στην 4 ^η ερώτηση ανάπτυξης.....	68

Συντομογραφίες

Δ.Ε.	Διπλωματική Εργασία
ΔΙΠΙΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
Π.Ε.	Πτυχιακή Εργασία
SDLC	Software Development Life Cycle
UX	User Experience
AUT	Application Under Test
GUI	Graphical User Interface
QA	Quality Assurance
CI	Continuous Integration
CD	Continuous Delivery
CT	Continuous Testing
DevOps	Development and Operations
UI	User Interface
API	Application Programming Interface
SaaS	Software as a Service
DOM	Document Object Model
URL	Uniform Resource Locator
ID	Identifier
OCR	Optical Character Recognition
CI-CD	Continuous Integration and Delivery

Κεφάλαιο 1ο: Εισαγωγή

Τα συστήματα λογισμικού αποτελούν, πλέον, αναπόσπαστο κομμάτι της καθημερινής μας ζωής. Στους περισσότερους ανθρώπους έχει συμβεί να χρησιμοποιήσουν ένα λογισμικό το οποίο δεν λειτουργούσε όπως αναμενόταν. Το λογισμικό που δεν λειτουργεί σωστά μπορεί να οδηγήσει σε πολλά προβλήματα, συμπεριλαμβανομένης της απώλειας χρημάτων, χρόνου αλλά και επιχειρηματικής φήμης. Ο Έλεγχος Λογισμικού (Software Testing) αξιολογεί την ποιότητα του λογισμικού και συμβάλλει στη μείωση του κινδύνου αποτυχίας του λογισμικού κατά τη λειτουργία.

Ο έλεγχος λογισμικού είναι ένα σύνολο δραστηριοτήτων για την ανακάλυψη ελαττωμάτων και την αξιολόγηση της ποιότητας των λειτουργιών των λογισμικών. Μια κοινός λανθασμένη άποψη σχετικά με τον έλεγχο (testing) είναι ότι αποτελείται μόνο από την εκτέλεση δοκιμών (δηλαδή, την εκτέλεση του λογισμικού και τον έλεγχο των αποτελεσμάτων των δοκιμών). Ωστόσο, ο έλεγχος λογισμικού περιλαμβάνει και άλλες δραστηριότητες και πρέπει να ευθυγραμμίζεται με τον κύκλο ζωής ανάπτυξης λογισμικού (Software Development Lifecycle—SDLC). Ακόμη μια εσφαλμένη θεωρία είναι ότι οι δοκιμές επικεντρώνονται αποκλειστικά στην επαλήθευση του αντικειμένου δοκιμής. Ενώ η δοκιμή περιλαμβάνει την επαλήθευση, δηλαδή τον έλεγχο του κατά πόσον το σύστημα πληροί τις καθορισμένες απαιτήσεις, περιλαμβάνει επίσης την επικύρωση, δηλαδή τον έλεγχο του κατά πόσον το σύστημα ανταποκρίνεται στις ανάγκες των χρηστών και άλλων λειτουργιών του περιβάλλοντος.

Ο έλεγχος μπορεί να είναι δυναμικός (dynamic) ή στατικός (static). Ο δυναμικός έλεγχος περιλαμβάνει την εκτέλεση του λογισμικού, ενώ ο στατικός όχι. Ο στατικός έλεγχος περιλαμβάνει ανασκοπήσεις και στατική ανάλυση. Αντιθέτως, ο δυναμικός έλεγχος χρησιμοποιεί διαφορετικούς τύπους τεχνικών ελέγχου και προσεγγίσεων για την εξαγωγή περιπτώσεων ελέγχου. Ο έλεγχος δεν είναι μόνο μια τεχνική δραστηριότητα. Πρέπει επίσης να δίνεται έμφαση στον σχεδιασμό, στη διαχείριση, να γίνονται εκτιμήσεις, να παρακολουθείται και να ελέγχεται κατάλληλα. [1]

Οι μηχανικοί ελέγχου (testers) χρησιμοποιούν εργαλεία, αλλά είναι σημαντικό να θυμόμαστε ότι ο έλεγχος είναι σε μεγάλο βαθμό μια διανοητική δραστηριότητα, που απαιτεί από τους μηχανικούς ελέγχου να έχουν εξειδικευμένες γνώσεις, να κατέχουν δεξιότητες ανάλυσης και να εφαρμόζουν κριτική και συστημική σκέψη. [1]

Ο κύκλος ζωής ανάπτυξης ενός λογισμικού είναι μια διαδικασία που δημιουργεί μια δομή ανάπτυξης λογισμικού. Υπάρχουν διάφορες φάσεις στο πλαίσιο του SDLC όπου κάθε μία έχει τις δικές της διαφορετικές δραστηριότητες. Κάνει την ομάδα ανάπτυξης ικανή να σχεδιάζει, να δημιουργεί και να παραδίδει ένα προϊόν υψηλής ποιότητας. Ο SDLC περιγράφει τις διάφορες φάσεις της ανάπτυξης λογισμικού και τη σειρά εκτέλεσης των φάσεων, όπου κάθε φάση απαιτεί την διαδοχική αλληλουχία από την προηγούμενη. Οι απαιτήσεις (requirements) μεταφράζονται σε σχεδιασμό (design), ο σχεδιασμός σε ανάπτυξη (development) και η ανάπτυξη σε έλεγχο (testing), για να καταλήξει τελικά στον πελάτη.

Οι στόχοι των ελέγχων μπορεί να ποικίλουν, ανάλογα με το πλαίσιο, το οποίο περιλαμβάνει το προϊόν εργασίας που δοκιμάζεται, το επίπεδο δοκιμής, τους κινδύνους, τον κύκλο ζωής ανάπτυξης λογισμικού που ακολουθείται, και παράγοντες που σχετίζονται με το επιχειρηματικό πλαίσιο, για παράδειγμα η εταιρική δομή, οι ανταγωνιστικές εκτιμήσεις ή ο χρόνος διάθεσης στην αγορά. Οι πιο συνηθισμένες περιπτώσεις εξέτασης είναι η αξιολόγηση των προϊόντων εργασίας, όπως οι προδιαγραφές, η περιήγηση του χρήστη (User Experience—UX), ο σχεδιασμός και ο κώδικας. Επίσης, είναι καλό να παραχθεί ένα μεγάλο ποσοστό από ελέγχους του κώδικα (tests) ώστε να καλυφθούν όσο

το δυνατόν περισσότερα σενάρια χρήσης του προϊόντος, για την άμεση εντόπιση σφαλμάτων και τη μείωση των κινδύνων. Τέλος, πρέπει να εξασφαλιστεί η παροχή πληροφοριών στους αρμόδιους της εταιρείας ώστε να μπορούν να λαμβάνουν τεκμηριωμένες αποφάσεις και να χτιστεί εμπιστοσύνη στο τμήμα διασφάλισης ποιότητας. [2]

1.1 Σημαντικότητα του Ελέγχου

Σε αυτό το σημείο μπορούμε να ομολογήσουμε ότι όλοι μας κάνουμε σφάλματα, για αυτό και ο έλεγχος στον κώδικα του προϊόντος είναι απαραίτητος. Ορισμένα από αυτά τα λάθη δεν είναι σημαντικά, αλλά ορισμένα μπορεί να μας κοστίσουν ακριβά ή ακόμη και να απειλήσουν τη ζωή μας. Πρέπει να δοκιμάζουμε οτιδήποτε παράγουμε, επειδή τα πράγματα μπορεί να πάνε στραβά καθώς οι άνθρωποι μπορούν να κάνουν λάθη ανά πάσα στιγμή.

Τα ανθρώπινα λάθη μπορούν να προκαλέσουν ένα ελάττωμα ή μια αποτυχία σε οποιοδήποτε στάδιο του κύκλου ζωής της ανάπτυξης λογισμικού. Τα αποτελέσματα ταξινομούνται ως ασήμαντα ή καταστροφικά, ανάλογα με τις συνέπειες που προκαλεί το κάθε σφάλμα. [3]

Η απαίτηση των αυστηρών ελέγχων και της σχετικής τεκμηρίωσής τους κατά τη διάρκεια του κύκλου ζωής της ανάπτυξης λογισμικού προκύπτει από τρεις παράγοντες [3]. Πρωταρχικά, για τον εντοπισμό ελαττωμάτων, δεύτερον, για τη μείωση των ελαττωμάτων στις λειτουργίες ενός λογισμικού συστήματος ξεχωριστά ή και σε ολόκληρο το σύστημα και τρίτον, στην αύξηση της συνολικής ποιότητας του συστήματος.

Μπορεί επίσης να υπάρχει απαίτηση για τη διενέργεια ελέγχου λογισμικού για τη συμμόρφωση με νομικές απαιτήσεις ή ειδικά βιομηχανικά πρότυπα. Αυτά τα πρότυπα και οι κανόνες μπορούν να καθορίσουν τι είδους τεχνικές πρέπει να χρησιμοποιούμε για την ανάπτυξη προϊόντων. Για παράδειγμα, οι βιομηχανίες κινητήρων, αεροναυπηγικής, ιατρικών και φαρμακευτικών προϊόντων κ.λπ. διαθέτουν πρότυπα που καλύπτουν τους ελέγχους του προϊόντος.

Ο έλεγχος είναι σημαντικό καθώς ανακαλύπτει ελαττώματα και σφάλματα πριν από την παράδοση προς τον πελάτη, γεγονός που εγγυάται την ποιότητα του λογισμικού. Διασφαλίζει πως το λογισμικό είναι πιο αξιόπιστο και εύχρηστο. Ακόμη, ένα σχολαστικά δοκιμασμένο λογισμικό εξασφαλίζει αξιόπιστη και υψηλή απόδοση για όλες τις λειτουργίες του λογισμικού. Τα παραπάνω σημεία τονίζουν τη σημασία των ελέγχων για ένα αξιόπιστο και εύχρηστο προϊόν λογισμικού.

Ας θέσουμε ως παράδειγμα μια καθημερινή περίπτωση, ας υποθέσουμε ότι χρησιμοποιούμε μια εφαρμογή 'Net Banking' για να μεταφέρουμε ένα χρηματικό ποσό σε έναν λογαριασμό ενός φίλου μας. Έτσι, ξεκινάμε τη συναλλαγή, λαμβάνουμε ένα μήνυμα επιτυχούς συναλλαγής και το ποσό αφαιρείται επίσης από το λογαριασμό μας. Ωστόσο, ο φίλος μας μας επιβεβαιώνει ότι δεν έχει λάβει ακόμη καμία πίστωση στο λογαριασμό του. Ομοίως, και ο δικός μας λογαριασμός δεν αντικατοπτρίζει επίσης την αντίστροφη συναλλαγή. Αυτό σίγουρα θα μας αναστάτωνε και θα μας δυσαρεστούσε ως πελάτες. Τώρα, τίθεται το εξής ερώτημα, γιατί συνέβη αυτό; Οφείλεται στον ακατάλληλο έλεγχο της εφαρμογής 'Net Banking' πριν από την κυκλοφορία; Ο ενδεδειγμένος έλεγχος της ιστοσελίδας για όλες τις πιθανές λειτουργίες του χρήστη θα οδηγούσε στον έγκαιρο εντοπισμό αυτού του προβλήματος. Ως εκ τούτου, μπορεί κανείς να το διορθώσει πριν από την δημοσίευση του στο κοινό για μια ομαλότερη εμπειρία. [3]

1.2 Η Συμβολή του Ελέγχου για την Επιτυχία

Στο παραπάνω παράδειγμα, μπορούμε να παρατηρήσουμε ότι λόγω της παρουσίας σφαλμάτων, το σύστημα απέτυχε να εκτελέσει την απαιτούμενη λειτουργία και δεν ανταποκρίθηκε στις απαιτήσεις του πελάτη. Οι κατάλληλες τεχνικές ελέγχου που εφαρμόζονται σε κάθε επίπεδο ελέγχου, σε συνδυασμό με το κατάλληλο επίπεδο τεχνογνωσίας ελέγχου, εξασφαλίζουν την απόλυτη μείωση της συχνότητας τέτοιου είδους αποτυχιών λογισμικού.

Η συμμετοχή των μηχανικών ελέγχου κατά τη φάση της διαπίστωσης των απαιτήσεων, εξασφαλίζει τον εντοπισμό ορισμένων ελαττωμάτων των απαιτήσεων ακόμη και πριν από την υλοποίησή τους. Γεγονός που μειώνει σημαντικά το κόστος επιδιόρθωσης. Επίσης, ο μηχανικός ελέγχου αποκτά την ευρύτερη εικόνα του προϊόντος σε αυτό το στάδιο. Η οποία, με τη σειρά της, τον βοηθά στη φάση εκτέλεσης του προϊόντος .

Οι μηχανικοί ελέγχου, εργάζονται στενά μαζί με τους σχεδιαστές (designers) του συστήματος, έχοντας ως στόχο τη καλύτερη κατανόηση του κάθε μέρους του σχεδιασμού. Το γεγονός αυτό, θα βοηθήσει στη μείωση του κινδύνου θεμελιωδών σχεδιαστικών ελαττωμάτων και θα επιτρέψει τον εντοπισμό σφαλμάτων σε πρώιμο στάδιο. Επιπλέον, βελτιώνει την ποιότητα των σεναρίων ολοκλήρωσης και οδηγεί σε καλύτερη ποιότητα ελαττωμάτων.

Όταν οι μηχανικοί ελέγχου συνεργάζονται δίπλα στους προγραμματιστές, ενισχύεται η κατανόηση κάθε μέρους του κώδικα. Επιπλέον, δίνεται η δυνατότητα στους μηχανικούς ελέγχου να κάνουν πιο αποτελεσματικές δοκιμές σε σχέση με τις απαιτήσεις του πελάτη. Με αυτόν τον τρόπο μειώνεται ο κίνδυνος ολίσθησης ελαττωμάτων. Οι μηχανικοί ελέγχου γνωρίζουν, επίσης, τις περιοχές που θεωρούνται επικίνδυνες από τον προγραμματιστή, ώστε να μπορούν να προσαρμόσουν ανάλογα τις προτεραιότητές τους. Επιπρόσθετα, οι προγραμματιστές αποκτούν τη διορατικότητα των μηχανικών ελέγχου. Αυτό βοηθά στην αναπαραγωγή των σφαλμάτων περιστασιακά, χωρίς να περάσουν από μια χρονοβόρα διαδικασία διαχείρισης ελαττωμάτων.

Τέλος, οι μηχανικοί ελέγχου επαληθεύουν και επικυρώνουν το λογισμικό πριν από την κυκλοφορία του. Αυτό βοηθά στον εντοπισμό σφαλμάτων που διαφορετικά μπορεί να μην είχαν γίνει αντιληπτά και υποστηρίζει τη διαδικασία εξάλειψης των ελαττωμάτων που προκάλεσαν τις αποτυχίες. Η εκτέλεση ελέγχων σε διάφορα επίπεδα αυξάνει την πιθανότητα το λογισμικό να έχει λιγότερα σφάλματα και επίσης να ανταποκρίνεται στις ανάγκες του πελάτη. [3]

1.3 Ο έλεγχος λογισμικού αποτελεί βασική πτυχή της ποιότητας

Η παράδοση ενός βέλτιστου ποιοτικού προϊόντος λογισμικού, το οποίο διαθέτει μοναδικά και καινοτόμα χαρακτηριστικά, αποτελούσε πάντοτε την προτεραιότητα της βιομηχανίας λογισμικού παγκοσμίως. Ωστόσο, χωρίς την αξιολόγηση των στοιχείων του λογισμικού υπό διάφορες αναμενόμενες και απροσδόκητες συνθήκες, η ομάδα δεν μπορεί να εγγυηθεί αυτές τις πτυχές. Ως εκ τούτου, πραγματοποιούνται έλεγχοι για τη δοκιμή κάθε συστατικού λογισμικού, μικρού και μεγάλου.

Ο έλεγχος λογισμικού είναι απαραίτητος, ώστε να μην αρχίσουμε ξανά από το μηδέν κάθε φορά που προκύπτει ένας σφάλμα. Μερικές φορές, δοκιμάζουμε ένα πλήρως ανεπτυγμένο προϊόν λογισμικού σε σχέση με τις απαιτήσεις του χρήστη και διαπιστώνουμε ότι λείπει κάποια βασική λειτουργικότητα. Αυτό μπορεί να συμβεί εξαιτίας ενός λάθους κατά τη συλλογή των απαιτήσεων ή κατά τη φάση του προγραμματισμού. Τότε, για να διορθώσουμε τέτοιου είδους λάθη, μπορεί να χρειαστεί να ξεκινήσουμε την ανάπτυξη ξανά από την αρχή. Η διόρθωση τέτοιου είδους λαθών γίνεται πολύ

κουραστική, χρονοβόρα και δαπανηρή. Ως εκ τούτου, είναι πάντα επιθυμητός ο έλεγχος του λογισμικού στη φάση ανάπτυξής του.

Η αξιολόγηση της ευκολίας χρήσης του λογισμικού προσδιορίζει πόσο εύκολα μπορούν να χρησιμοποιήσουν το τελικό προϊόν οι τελικοί χρήστες. Ο έλεγχος λογισμικού εξασφαλίζει την κατασκευή του προϊόντος με τρόπο που να ανταποκρίνεται στις προσδοκίες του χρήστη όσον αφορά τη συμμόρφωση με τις απαιτήσεις με άνετο, ικανοποιητικό και απλοϊκό τρόπο.

Μπορούμε να επαληθεύσουμε όλες τις πτυχές του λογισμικού κατά τον έλεγχο του, όπως ο έλεγχος των βασικών λειτουργιών καθώς και ο έλεγχος ενός συστήματος για απροσδόκητες συνθήκες. Οι μη αναμενόμενες συνθήκες μπορεί να προέρχονται από εσφαλμένο τύπο δεδομένων ή να οφείλονται σε επίθεση πειρατείας. Ως εκ τούτου, ο έλεγχος διασφαλίζει ότι το σύστημα μπορεί να χειριστεί πολύ καλά αυτές τις καταστάσεις. Έτσι, αν διαπιστώσουμε εκ των προτέρων κάποιο σφάλμα, έχουμε τη δυνατότητα να τα διορθώσουμε. Μπορεί να αποτρέψει τα παράπονα όταν το λογισμικό ή η εφαρμογή έχει φτάσει στους πελάτες.

Ο έλεγχος λογισμικού συμβάλλουν στην επιτάχυνση της ανάπτυξης. Ο έλεγχος λογισμικού βοηθάει τους προγραμματιστές να εντοπίσουν σφάλματα και σενάρια αναπαραγωγής του σφάλματος, τα οποία με τη σειρά τους τους βοηθούν να προβούν στις απαραίτητες διορθώσεις γρήγορα. Εκτός αυτού, οι μηχανικοί ελέγχου μπορούν να εργάζονται παράλληλα με την ομάδα ανάπτυξης, κατανοώντας έτσι λεπτομερώς το σχεδιασμό, τις περιοχές κινδύνου κ.λπ. Αυτή η ανταλλαγή γνώσεων μεταξύ μηχανικών ελέγχου και προγραμματιστών επιταχύνει ολόκληρη τη διαδικασία ανάπτυξης.

Συμπερασματικά, η σημασία του ελέγχου λογισμικού είναι επιτακτική. Οι μηχανικοί ελέγχου αποτελούν κρίσιμο στοιχείο της ανάπτυξης προϊόντων λογισμικού, διότι βελτιώνουν τη συνέπεια και την απόδοση. Το κύριο όφελος των ελέγχων είναι ο εντοπισμός και η επακόλουθη εξάλειψη των σφαλμάτων. Ωστόσο, οι έλεγχοι βοηθούν επίσης τους προγραμματιστές και τους μηχανικούς ελέγχου να συγκρίνουν τα πραγματικά και τα αναμενόμενα αποτελέσματα προκειμένου να βελτιώσουν την ποιότητα. Εάν η παραγωγή λογισμικού γίνει χωρίς δοκιμή, θα μπορούσε το τελικό προϊόν να είναι άχρηστο ή μερικές φορές επικίνδυνο για τους πελάτες. Έτσι, ένας μηχανικός ελέγχου θα πρέπει να φοράει ένα μοναδικό καπέλο που προστατεύει την αξιοπιστία του λογισμικού και καθιστά ασφαλή τη χρήση του σε πραγματικές συνθήκες. [3]

1.4 Κατηγορίες Ελέγχου Λογισμικού

Σε αυτή την ενότητα, θα κατανοήσουμε τους διάφορους τύπους ελέγχων λογισμικού, οι οποίοι μπορούν να χρησιμοποιηθούν κατά τη διάρκεια του κύκλου ζωής ανάπτυξης λογισμικού. Όπως γνωρίζουμε, ο έλεγχος λογισμικού είναι μια διαδικασία ανάλυσης της λειτουργικότητας μιας εφαρμογής σύμφωνα με τις προϋποθέσεις του πελάτη. Εάν θέλουμε να διασφαλίσουμε ότι το λογισμικό μας είναι σε σταθερή κατάσταση ή απαλλαγμένο από σφάλματα, πρέπει να εκτελέσουμε τους διάφορους τύπους ελέγχων λογισμικού, διότι ο έλεγχος είναι η μόνη μέθοδος που καθιστά την εφαρμογή μας απαλλαγμένη από σφάλματα.

Η κατηγοριοποίηση των ελέγχων λογισμικού αποτελεί μέρος διαφόρων δραστηριοτήτων των ελέγχων, όπως η στρατηγική που θα ακολουθήσει ο έλεγχος, ποιο είναι το έργο της υλοποίησης του ελέγχου, ένας καθορισμένος στόχος που γίνεται ο έλεγχος κ.λπ. Ο έλεγχος λογισμικού είναι η εκτέλεση του λογισμικού για την εύρεση ελαττωμάτων. Ο σκοπός της ύπαρξης ενός τύπου ελέγχου είναι η επιβεβαίωση της εφαρμογής υπό έλεγχο (Application Under Test—AUT). Μετά το τμήμα σχεδιασμού και κωδικοποίηση του κύκλου ανάπτυξης, όταν η εφαρμογή τίθεται υπό έλεγχο, τότε εκείνη τη στιγμή η κατάσταση της εφαρμογής είναι υπό έλεγχο, οπότε εκείνη τη χρονική περίοδο η εφαρμογή

ονομάζεται ‘Εφαρμογή υπό έλεγχο’ [4]. Για να ξεκινήσουμε τον έλεγχο, θα πρέπει να έχουμε μια απαίτηση, μια εφαρμογή έτοιμη, τους απαραίτητους πόρους διαθέσιμους. Για να διατηρήσουμε την υπευθυνότητα, θα πρέπει να αναθέσουμε μια αντίστοιχη ενότητα σε διαφορετικούς μηχανικούς ελέγχου. Ο έλεγχος λογισμικού χωρίζεται κυρίως σε δύο κατηγορίες, το **Χειροκίνητο Έλεγχο** (Manual Testing) και τον **Αυτοματοποιημένο Έλεγχο** (Automation Testing) τις οποίες θα αναλύσουμε παρακάτω. [5]

1.4.1 Χειροκίνητος Έλεγχος

Ο χειροκίνητος έλεγχος (manual testing), είναι μια διαδικασία ελέγχου λογισμικού κατά την οποία οι περιπτώσεις ελέγχου εκτελούνται χειροκίνητα, χωρίς τη χρήση αυτοματοποιημένου εργαλείου. Όλες οι περιπτώσεις ελέγχου εκτελούνται χειροκίνητα από τον μηχανικό ελέγχου, σύμφωνα με την οπτική γωνία του τελικού χρήστη. Με αυτό τον τρόπο εξασφαλίζεται αν η εφαρμογή λειτουργεί σύμφωνα με τις προδιαγραφές του εγγράφου απαιτήσεων ή όχι. Οι περιπτώσεις ελέγχου σχεδιάζονται και υλοποιούνται για να ολοκληρωθεί σχεδόν το εκατό (100) τοις εκατό της εφαρμογής λογισμικού. Οι αναφορές για τα σενάρια ελέγχου (test cases) παράγονται επίσης χειροκίνητα.

Ο χειροκίνητος έλεγχος είναι μια από τις πιο θεμελιώδεις διαδικασίες ελέγχου, καθώς μπορεί να εντοπίσει τόσο ορατά όσο και κρυφά ελαττώματα του λογισμικού [6]. Η διαφορά μεταξύ της αναμενόμενης εξόδου και της εξόδου που δίνει το λογισμικό, ορίζεται ως ελάττωμα. Ο προγραμματιστής είναι υπεύθυνος στο να διορθώνει τα ελαττώματα και στη συνέχεια να τα παραδίδει στον μηχανικό ελέγχου για επανέλεγχο.

Ο χειροκίνητος έλεγχος είναι υποχρεωτικός για κάθε νέο αναπτυσσόμενο λογισμικό πριν από τον αυτοματοποιημένο έλεγχο. Αυτός ο έλεγχος απαιτεί μεγάλες προσπάθειες και χρόνο, αλλά δίνει τη βεβαιότητα ότι το λογισμικό λειτουργεί σωστά και δεν έχει σφάλματα. Ο χειροκίνητος έλεγχος απαιτεί γνώση των τεχνικών χειροκίνητου ελέγχου αλλά όχι οποιουδήποτε εργαλείου αυτοματοποιημένου ελέγχου.

Ο χειροκίνητος έλεγχος είναι απαραίτητος, καθώς κάθε φορά που μια εφαρμογή βγαίνει στην αγορά και είναι ασταθής, είναι πολύ πιθανό να εμφανιστεί κάποιο σφάλμα, είτε υπολειτουργίες είτε δημιουργούνται προβλήματα ενώ οι τελικοί χρήστες τη χρησιμοποιούν. Για να μην φτάσουμε σε αυτό το σημείο και αναγκαστούμε να αντιμετωπίσουμε τέτοιου είδους προβλήματα, πρέπει να εκτελέσουμε έναν γύρο ελέγχων για να απαλλάξουμε την εφαρμογή από σφάλματα, να βρεθεί σε σταθερή κατάσταση και να παραδοθεί ένα ποιοτικό προϊόν προς τον πελάτη, διότι εάν η εφαρμογή δεν έχει σφάλματα, ο τελικός χρήστης θα έχει μια καλύτερη εμπειρία χρήσης όσο χρησιμοποιεί την εφαρμογή.

Εάν ο μηχανικός ελέγχου κάνει χειροκίνητους ελέγχους, μπορεί να δοκιμάσει την εφαρμογή από την οπτική γωνία του τελικού χρήστη και να εξοικειωθεί περισσότερο με το προϊόν, γεγονός που τον βοηθά να γράψει στοχευμένα σενάρια ελέγχου της εφαρμογής και να δώσει γρήγορη ανατροφοδότηση για τυχόν προβλήματα. [6]

1.4.1.1 Τύποι Χειροκίνητου Ελέγχου

Υπάρχουν διάφοροι μέθοδοι που χρησιμοποιούνται για χειροκίνητο έλεγχο. Κάθε τεχνική χρησιμοποιείται ανάλογα με τα κριτήρια ελέγχου της και διακρίνονται σε τρεις (3) κύριες κατηγορίες: Έλεγχος Λευκού Κουτιού (White Box Testing), Έλεγχος Μαύρου Κουτιού (Black Box Testing) και Έλεγχος Γκριζού Κουτιού (Gray Box Testing). [6]

1.4.1.1.1 Έλεγχος λευκού κουτιού

Η προσέγγιση του λευκού κουτιού, η οποία είναι επίσης γνωστή ως έλεγχος γυάλινου κουτιού, ελέγχει την εσωτερική κωδικοποίηση και την υποδομή ενός λογισμικού και επικεντρώνεται στον έλεγχο των προκαθορισμένων εισόδων σε σχέση με τις αναμενόμενες και επιθυμητές εξόδους. Βασίζεται στην εσωτερική λειτουργία μιας εφαρμογής και περιστρέφεται γύρω από τον έλεγχο της εσωτερικής δομής. Σε αυτόν τον τύπο ελέγχου απαιτούνται δεξιότητες προγραμματισμού για τον σχεδιασμό σεναρίων. Ο πρωταρχικός στόχος των ελέγχων λευκού κουτιού είναι να επικεντρωθεί στη ροή των εισροών και των εκροών μέσω του λογισμικού και στην ενίσχυση της ασφάλειας του λογισμικού.

Ο όρος "λευκό κουτί" χρησιμοποιείται λόγω της εσωτερικής προοπτικής του συστήματος. Η ονομασία καθαρό κουτί (clear box) ή λευκό κουτί (white box) ή διαφανές κουτί (transparent box) δηλώνει τη δυνατότητα να δει κανείς μέσα από το εξωτερικό περίβλημα του λογισμικού τις εσωτερικές λειτουργίες του.

Οι προγραμματιστές εκτελώντας ελέγχους λευκού κουτιού, δοκιμάζουν κάθε γραμμή του κώδικα του προγράμματος. Οι προγραμματιστές εκτελούν τον έλεγχο λευκού κουτιού και στη συνέχεια στέλνουν την εφαρμογή ή το λογισμικό στην ομάδα των μηχανικών ελέγχου, όπου θα εκτελεστεί ο έλεγχος του μαύρου κουτιού για να επαληθευτεί η εφαρμογή μαζί με τις απαιτήσεις της, να εντοπιστούν τυχόν σφάλματα. Σε αυτή την περίπτωση, η εφαρμογή ή το λογισμικό θα σταλθεί πίσω στους προγραμματιστές μαζί με μια αναλυτική αναφορά σχετικά με τα σφάλματα που εντοπίστηκαν ώστε να διορθωθούν.

Οι προγραμματιστές διορθώνουν τα σφάλματα και εκτελούν έναν γύρο ελέγχου λευκού κουτιού και το στέλνουν στην ομάδα των μηχανικών ελέγχου. Εδώ, η διόρθωση των σφαλμάτων συνεπάγεται ότι το σφάλμα διαγράφεται και το συγκεκριμένο χαρακτηριστικό λειτουργεί κανονικά, χωρίς λάθη, στην εφαρμογή.

Οι μηχανικοί ελέγχου δεν θα συμμετέχουν στη διόρθωση των σφαλμάτων. Ως εκ τούτου, οι μηχανικοί ελέγχου θα πρέπει πάντα να βρίσκουν τα σφάλματα και οι προγραμματιστές θα πρέπει να εξακολουθούν να δουλεύουν πάνω στις διορθώσεις των σφαλμάτων. Εάν οι μηχανικοί ελέγχου ξοδεύουν τον περισσότερο χρόνο βοηθώντας στην επίλυση των σφαλμάτων, τότε μπορεί να μην ήταν σε θέση να βρουν νέα σφάλματα στην εφαρμογή. [7]

1.4.1.1.2 Έλεγχος μαύρου κουτιού

Ο έλεγχος μαύρου κουτιού είναι μια τεχνική ελέγχου λογισμικού που εξετάζει τη λειτουργικότητα του λογισμικού χωρίς να εισχωρεί στην εσωτερική δομή ή τον κώδικα. Η πρωταρχική πηγή του ελέγχου μαύρου κουτιού είναι μια προδιαγραφή απαιτήσεων που δηλώνεται από τον πελάτη.

Σε αυτή τη μέθοδο, ο μηχανικός ελέγχου επιλέγει μια συνάρτηση και δίνει μια τιμή εισόδου για να εξετάσει τη λειτουργικότητά της και ελέγχει αν η συνάρτηση δίνει την αναμενόμενη έξοδο ή όχι. Εάν η συνάρτηση παράγει σωστή έξοδο, τότε ο έλεγχος περνάει ως επιτυχημένος, διαφορετικά αποτυγχάνει. Η ομάδα των μηχανικών ελέγχου αναφέρει το αποτέλεσμα στην ομάδα ανάπτυξης και στη συνέχεια δοκιμάζει την επόμενη συνάρτηση. Μετά την ολοκλήρωση του ελέγχου όλων των λειτουργιών, εάν υπάρχουν σοβαρά προβλήματα, τότε δίνεται πίσω στην ομάδα ανάπτυξης για διόρθωση. [8]

1.4.1.1.3 Έλεγχος γκρίζου κουτιού

Ο έλεγχος γκρίζου κουτιού είναι μια μέθοδος ελέγχου λογισμικού για τον έλεγχο της εφαρμογής λογισμικού, μη έχοντας άριστη γνώση της εσωτερικής δομής του λογισμικού. Είναι ένας συνδυασμός των ελέγχων μαύρου κουτιού και λευκού κουτιού, επειδή περιλαμβάνει πρόσβαση στην εσωτερική

κωδικοποίηση για τον σχεδιασμό περιπτώσεων ελέγχου, όπως οι έλεγχοι λευκού κουτιού και οι πρακτικές ελέγχου που γίνονται σε επίπεδο λειτουργικότητας, όπως οι έλεγχοι μαύρου κουτιού.

Ο έλεγχος γκρίζου κουτιού εντοπίζει συνήθως σφάλματα που σχετίζονται με το συγκεκριμένο πλαίσιο και ανήκουν σε διαδικτυακά συστήματα. Για παράδειγμα, κατά τη διάρκεια του ελέγχου, εάν ο μηχανικός ελέγχου συναντήσει κάποιο σφάλμα, τότε κάνει αλλαγές στον κώδικα για να επιλύσει το σφάλμα και στη συνέχεια το δοκιμάζει ξανά σε πραγματικό χρόνο. Επικεντρώνεται σε όλα τα επίπεδα οποιουδήποτε πολύπλοκου συστήματος λογισμικού για να αυξήσει την κάλυψη των ελέγχων. Παρέχει τη δυνατότητα ελέγχου τόσο στο επίπεδο παρουσίασης όσο και της εσωτερικής δομής του κώδικα. Χρησιμοποιείται κυρίως στις κατηγορίες του έλεγχου ολοκλήρωσης (integration testing) και έλεγχου διείσδυσης (penetration testing).

Η συγκεκριμένη κατηγορία, παρέχει συνδυασμένα οφέλη τόσο των ελέγχων μαύρου κουτιού όσο και των ελέγχων λευκού κουτιού. Περιλαμβάνει ταυτόχρονα τις τιμές εισόδου τόσο των προγραμματιστών όσο και των μηχανικών ελέγχου για τη βελτίωση της συνολικής ποιότητας του προϊόντος, δίνοντας παράλληλα επαρκή χρόνο στους προγραμματιστές για να διορθώσουν τα ελαττώματα του προϊόντος. [9]

1.4.1.1.3.1 Λειτουργικός έλεγχος

Ο Λειτουργικός έλεγχος (Functional testing) είναι ένας τύπος ελέγχου λογισμικού που χρησιμοποιείται για να επαληθεύσει τη λειτουργικότητα μιας εφαρμογής λογισμικού, εάν οι λειτουργίες της εκτελούνται σύμφωνα με τις προδιαγραφές των απαιτήσεων. Στο λειτουργικό έλεγχο, κάθε λειτουργία ελέγχεται δίνοντας την τιμή, προσδιορίζοντας την έξοδο και επαληθεύοντας την πραγματική έξοδο με την αναμενόμενη τιμή. Ο λειτουργικός έλεγχος ανήκει στην ευρύτερη κατηγορία του ελέγχου μαύρου κουτιού η οποία παρουσιάζεται για να επιβεβαιώσει ότι η λειτουργικότητα μιας εφαρμογής ή ενός συστήματος συμπεριφέρεται όπως περιμένουμε. Εφαρμόζεται για να επαληθευτεί η λειτουργικότητα της εφαρμογής, καθώς επικεντρώνεται στις προδιαγραφές της εφαρμογής και όχι στον πραγματικό κώδικα. Ο μηχανικός έλεγχου πρέπει να ελέγξει μόνο το πρόγραμμα και όχι το σύστημα.

Ο σκοπός του λειτουργικού ελέγχου είναι να ελεγχθούν οι κύριες λειτουργίες εισόδου, μέσω του γραφικού περιβάλλοντος διεπαφής χρήστη (Graphical User Interface—GUI). Σε περίπτωση σφάλματος, θα εμφανιστεί στην οθόνη ένα αντίστοιχο μήνυμα έτσι ώστε ο χρήστης να καταλάβει ότι έχει ορίσει κάποια λάθος τιμή εισόδου και πρέπει να την αλλάξει ώστε να συνεχίσει την πλοήγησή του στην εφαρμογή.

Τα βήματα που πρέπει να ακολουθήσει ένας μηχανικός ελέγχου για να εφαρμόσει λειτουργικό έλεγχο δεν είναι τυχαία, γι' αυτό και χρειάζεται καλό σχεδιασμό. [10]

1. *Προαπαιτούμενα (Requirements)*: Πρώτα από όλα, ο μηχανικός έλεγχου οφείλει να γνωρίζει ποιες είναι οι προδιαγραφές και οι απαιτήσεις της εκάστοτε εφαρμογής λογισμικού.
2. *Πλάνο ελέγχου (Test plan)*: Μετά την ανάλυση και την ορθή κατανόηση των προδιαγραφών, ο μηχανικός έλεγχου καταρτίζει ένα σχέδιο για το πως θα ελέγξει με την σειρά όλες τις λειτουργίες της εφαρμογής.
3. *Σχεδιασμός σεναρίων ελέγχου*: Μετά το σχεδιασμό των ελέγχων, ο μηχανικός έλεγχου πρέπει να σχεδιάσει τα σενάρια χρήσης. Τα σενάρια αυτά αντιστοιχούν σε πιθανές κινήσεις που θα έκανε ένας χρήστης, καθώς περιηγείται στην εφαρμογή. Στόχος είναι να δημιουργηθούν όσο τον δυνατόν περισσότερα σενάρια τα οποία θα καλύψουν όλες τις λειτουργίες της εφαρμογής.

4. *Καταγραφή*: Έπειτα, ο μηχανικός ελέγχου πρέπει να δημιουργήσει έναν πίνακα στον οποίο θα καταγράψει όλα τα σενάρια ελέγχου σε μία στήλη με έναν σύντομο περιγραφικό τίτλο και μια δεύτερη στήλη που περιγράφει αναλυτικότερα το ποιές είναι οι κινήσεις που πρέπει να ακολουθήσει ο μηχανικός ελέγχου ώστε να εκτελέσει σωστά το εκάστοτε σενάριο.
5. *Εκτέλεση των σεναρίων ελέγχου*: Σε αυτό το βήμα ο μηχανικός ελέγχου θα εκτελέσει ένα σενάριο ελέγχου.
6. *Ανάλυση της εκτέλεσης των σεναρίων ελέγχου*: Μετά την εκτέλεση ενός σεναρίου ελέγχου πρέπει να κρατηθούν σημειώσεις για το αν ο έλεγχος έχει περάσει (pass), δηλαδή είχαμε τα αναμενόμενα αποτελέσματα, ή αν έσπασε (fail), δηλαδή κάτι δεν δούλεψε όπως αναμενόταν.
7. *Διαχείριση σφαλμάτων*: Τέλος, πρέπει να αναφερθούν τα σφάλματα που εντοπίστηκαν και η ομάδα των προγραμματιστών να προβεί άμεσα, σύμφωνα με την κρισιμότητα των σφαλμάτων, στην επίλυσή τους. [10]

Ο λειτουργικός έλεγχος μπορεί να χωριστεί σε υποκατηγορίες [10]. Μερικές από αυτές είναι οι εξής:

- *Έλεγχος Μονάδας (Unit Testing)*: Ο έλεγχος μονάδας είναι μια διαδικασία ανάπτυξης και ελέγχου του λογισμικού κατά την οποία τα μικρότερα ελέγξιμα τμήματα μιας εφαρμογής, που ονομάζονται μονάδες, ελέγχονται μεμονωμένα για την ορθή λειτουργία τους. Οι προγραμματιστές λογισμικού και μερικές φορές το προσωπικό της Διασφάλισης Ποιότητας (Quality Assurance—QA) ολοκληρώνουν τους ελέγχους μονάδων κατά τη διάρκεια της διαδικασίας ανάπτυξης.
- *Έλεγχος Καπνού (Smoke Testing)*: Ονομάζεται και ‘Έλεγχος επαλήθευσης’ (Build verification) είναι μια μέθοδος ελέγχου που χρησιμοποιείται για να διαπιστωθεί αν ένα νέο στοιχείο έχει υλοποιηθεί σε μια εφαρμογή και είναι έτοιμη για την επόμενη φάση ελέγχου. Αυτή η μέθοδος ελέγχου καθορίζει αν οι πιο κρίσιμες λειτουργίες ενός προγράμματος λειτουργούν, αλλά δεν εμβαθύνει σε λεπτότερες λεπτομέρειες. Για παράδειγμα, όλεγχος καπνού επαληθεύει την επιτυχή εκκίνηση της εφαρμογής και ελέγχει ότι το γραφικό περιβάλλον ανταποκρίνεται.
- *Έλεγχος Λογικής (Sanity Testing)*: Ο έλεγχος λογικής περιλαμβάνει ότι ολόκληρο το επιχειρηματικό σενάριο υψηλού επιπέδου λειτουργεί σωστά. Ο έλεγχος λογικής γίνεται για να ελεγχθεί η λειτουργικότητα μετά τη διόρθωση των σφαλμάτων. Αποτελεί μια πιο προχωρημένη μορφή του ελέγχου καπνού. Για παράδειγμα, αν η είσοδος στην εφαρμογή λειτουργεί κανονικά, τότε όλα τα κουμπιά πρέπει να λειτουργούν σωστά, μετά το πάτημα των κουμπιών θα πρέπει να είναι δυνατή ή όχι η πλοήγηση στη σελίδα, κτλ.
- *Έλεγχος Παλινδρόμησης (Regression Testing)*: Ο έλεγχος παλινδρόμησης είναι ένας τύπος ελέγχου που συγκεντρώνει το σύνολο των σεναρίων ελέγχου για να διασφαλίσει ότι όλες οι αλλαγές στον κώδικα δεν θα πρέπει να έχουν παρενέργειες στην υπάρχουσα λειτουργικότητα του συστήματος.
- *Έλεγχος Ενσωμάτωσης (Integration Testing)*: Ο έλεγχος ενσωμάτωσης είναι το στάδιο κατά το οποίο συνδυάζονται μεμονωμένες μονάδες λογισμικού και ελέγχονται ως ομάδα. Διεξάγεται για να αξιολογηθεί η συμμόρφωση ενός συστήματος ή στοιχείου (component) με τις καθορισμένες λειτουργικές απαιτήσεις.
- *Έλεγχος Ευχρηστίας (Usability Testing)*: Σε αυτό το στάδιο, πραγματικοί πελάτες δοκιμάζουν το προϊόν στο παραγωγικό περιβάλλον. Αυτό το στάδιο είναι απαραίτητο για να μετρηθεί πόσο

άνετα αισθάνεται ο πελάτης με τη διεπαφή. Τα σχόλιά τους λαμβάνονται υπόψη για περαιτέρω βελτιώσεων στη λειτουργικότητα της εφαρμογής. [11]

1.4.1.1.3.2 Μη-Λειτουργικός έλεγχος

Ο Μη-Λειτουργικός έλεγχος (Non-functional testing) είναι ένας τύπος ελέγχου λογισμικού που χρησιμοποιείται για την αξιολόγηση της απόδοσης, της χρηστικότητας, της αξιοπιστίας και άλλων μη λειτουργικών χαρακτηριστικών μιας εφαρμογής λογισμικού. Αποσκοπεί στον έλεγχο της ετοιμότητας ενός συστήματος σύμφωνα με μη-λειτουργικά κριτήρια τα οποία ο λειτουργικός έλεγχος δεν λαμβάνει ποτέ υπόψη. [12]

Ο μη-λειτουργικός έλεγχος είναι απαραίτητος για την επιβεβαίωση της αξιοπιστίας και της λειτουργικότητας του λογισμικού. Οι προδιαγραφές απαιτήσεων λογισμικού χρησιμεύουν ως βάση για αυτή τη μέθοδο λογισμικού ελέγχου, οι οποίες επιτρέπουν στις ομάδες διασφάλισης ποιότητας να ελέγχουν αν το σύστημα συμμορφώνεται με τις απαιτήσεις των χρηστών. Η αύξηση της χρηστικότητας, της αποτελεσματικότητας, της συντηρησιμότητας και της φορητότητας του προϊόντος είναι ο στόχος του μη-λειτουργικού ελέγχου. Βοηθά στη μείωση του κατασκευαστικού κινδύνου που συνδέεται με τα μη-λειτουργικά στοιχεία του προϊόντος. [13]

Ο μη-λειτουργικός έλεγχος μπορεί να χωριστεί σε υποκατηγορίες. Μερικές από αυτές είναι οι εξής:

- *Έλεγχος Απόδοσης (Performance Testing)*: Ο έλεγχος απόδοσης εξαλείφει την αιτία πίσω από την αργή και περιορισμένη απόδοση του λογισμικού. Η ταχύτητα ανάγνωσης του λογισμικού πρέπει να είναι όσο το δυνατόν ταχύτερη. Για τον έλεγχο απόδοσης πρέπει να καθοριστεί μια καλά δομημένη και σαφής προδιαγραφή σχετικά με την αναμενόμενη ταχύτητα. Διαφορετικά, το αποτέλεσμα του ελέγχου (επιτυχία ή αποτυχία) δεν θα είναι προφανές.
- *Έλεγχος Φορτίου (Load Testing)*: Ο έλεγχος φορτίου περιλαμβάνει τον έλεγχο της χωρητικότητας του φορτίου του συστήματος. Χωρητικότητα του φορτίου σημαίνει ότι όλο και περισσότεροι άνθρωποι μπορούν να εργάζονται στο σύστημα ταυτόχρονα.
- *Έλεγχος Ασφαλείας (Security Testing)*: Οι έλεγχοι ασφαλείας χρησιμοποιούνται για την ανίχνευση των αδυναμιών ασφάλειας της εφαρμογής λογισμικού. Ο έλεγχος γίνεται μέσω της διερεύνησης της αρχιτεκτονικής του συστήματος και τη στρατηγική σκέψη ενός επιτιθέμενου. Τα σενάρια ελέγχου διεξάγονται σε περιοχές όπου ο κώδικα είναι ευάλωτος αλλά και πιο πιθανό να μπει στο στόχαστρο μιας επίθεσης.
- *Έλεγχος Φορητότητας (Portability Testing)*: Ο έλεγχος φορητότητας του λογισμικού χρησιμοποιείται για να εξακριβωθεί αν το σύστημα μπορεί να τρέξει σε διαφορετικά λειτουργικά συστήματα χωρίς να εμφανιστεί κάποιο σφάλμα. Ο έλεγχος αυτός ελέγχει επίσης τη λειτουργικότητα του λογισμικού όταν υπάρχει σε ίδιο λειτουργικό σύστημα αλλά σε διαφορετικό ηλεκτρονικό υπολογιστή.
- *Έλεγχος Ευθυνών (Accountability Testing)*: Ο έλεγχος ευθυνών γίνεται για να ελεγχθεί αν το σύστημα λειτουργεί σωστά ή όχι. Μια λειτουργία πρέπει να δίνει το ίδιο αποτέλεσμα για το οποίο έχει δημιουργηθεί. Εάν το σύστημα δίνει την αναμενόμενη έξοδο, περνάει τον έλεγχο, διαφορετικά αποτυγχάνει.
- *Έλεγχος Αξιοπιστίας (Reliability Testing)*: Ο έλεγχος αξιοπιστίας υποθέτει ότι το σύστημα λογισμικού λειτουργεί χωρίς βλάβη υπό καθορισμένες συνθήκες ή όχι. Το σύστημα πρέπει να εκτελείται για συγκεκριμένο χρόνο και αριθμό διεργασιών. Εάν το σύστημα αποτύχει υπό αυτές τις καθορισμένες συνθήκες, ο έλεγχος αξιοπιστίας θα αποτύχει.

- *Έλεγχος Αποδοτικότητας (Efficiency Testing)*: Ο έλεγχος αποδοτικότητας εξετάζει τον αριθμό των πόρων που απαιτούνται για την ανάπτυξη ενός συστήματος λογισμικού και πόσοι από αυτούς χρησιμοποιήθηκαν. Περιλαμβάνει, επίσης, τον έλεγχο των τριών ακόλουθων σημείων. Πρώτον, οι απαιτήσεις του πελάτη πρέπει να ικανοποιούνται από το λογισμικό σύστημα. Δεύτερον, ένα λογισμικό σύστημα πρέπει να επιτυγχάνει τις προδιαγραφές του πελάτη. Τρίτον, θα πρέπει να καταβληθούν αρκετές προσπάθειες για την ανάπτυξη ενός λογισμικού συστήματος. [13]

1.4.2 Αυτοματοποιημένος Έλεγχος

Ο αυτοματοποιημένος έλεγχος είναι μια διαδικασία ελέγχου λογισμικού και άλλων τεχνολογικών προϊόντων για να διασφαλιστεί ότι πληρούν τις απαραίτητες προδιαγραφές. Ουσιαστικά, πρόκειται για μια κατηγορία ελέγχου η οποία ελέγχει δύο φορές ότι ο εξοπλισμός ή το λογισμικό κάνει ακριβώς αυτό για το οποίο σχεδιάστηκε. Ελέγχει για σφάλματα, ελαττώματα και οποιαδήποτε άλλα ζητήματα που μπορεί να προκύψουν κατά την ανάπτυξη προϊόντων, όπως συμβαίνει και στον χειροκίνητο έλεγχο.

Παρόλο που ορισμένοι τύποι ελέγχου, όπως τον έλεγχο παλινδρόμησης ή το λειτουργικό έλεγχο, μπορούν να γίνουν χειροκίνητα, υπάρχουν μεγαλύτερα οφέλη από την αυτόματη διεξαγωγή τους. Οι αυτοματοποιημένοι έλεγχοι μπορούν να εκτελούνται οποιαδήποτε στιγμή της ημέρας. Χρησιμοποιούν σεναριακές ακολουθίες για την εξέταση του εκάστοτε λογισμικού. Στο τέλος της εκτέλεσης, γίνεται αναφορά για το ό,τι έχει βρεθεί και οι πληροφορίες αυτές μπορούν να συγκριθούν με προηγούμενες εκτελέσεις των ελέγχων. Οι μηχανικοί αυτοματοποιημένων ελέγχων (automation testers) γράφουν τα σενάρια ελέγχου σε διάφορες γλώσσες προγραμματισμού, όπως C#, Java, JavaScript, Python, κτλ.

Πολλές επιχειρήσεις ανάπτυξης λογισμικού αρχίζουν και καθιερώνουν τη διάραση QA μηχανικών αυτοματοποιημένων ελέγχων. Αυτοί είναι αρμόδιοι για να σχεδιάζουν και να γράφουν τα σενάρια ελέγχου στα αρχικά στάδια. Ο QA μηχανικός αυτοματοποιημένων ελέγχων θα συνεργαστεί με τους υπόλοιπους μηχανικούς αυτοματοποιημένων ελέγχων και τους προγραμματιστές για την πραγματικό έλεγχο του λογισμικού και των προϊόντων. Τέλος, θα σχηματιστεί μια ομάδα και θα ελέγξει τις πρωτοβουλίες των αυτοματοποιημένων ελέγχων και θα χρησιμοποιήσει διαφορετικούς τύπους πλαισίων λογισμικού (frameworks) αυτοματοποιημένων ελέγχων για να καθορίσει το καλύτερο για την επιτυχία των αυτοματοποιημένων ελέγχων.

Όταν η ομάδα αρχίζει να εργάζεται με πλαίσια λογισμικού για έλεγχο μονάδων, πρέπει να γνωρίζει τα χαρακτηριστικά (attributes), τους δρομείς (runners), τους ισχυρισμούς (assertions), τα στιγμιότυπα οθόνης (screenshots), τις σουίτες επικύρωσης (test suites) και τη συνεχή ενσωμάτωση (Continuous Integration—CI). Τα δημοφιλή πλαίσια λογισμικού για ελέγχους ευχρηστίας περιλαμβάνουν το JUnit για τη Java και το Pytest για την Python.

Η τήρηση των πρωτοκόλλων ελέγχων είναι σημαντική στον κλάδο της τεχνολογίας. Είναι ζωτικής σημασίας για τη συνεχή παράδοση (Continuous Delivery—CD) και το συνεχή έλεγχο (Continuous Testing—CT). Οι ομάδες ανάπτυξης και λειτουργιών (Development and Operations—DevOps) και οι ευέλικτες ομάδες ανάπτυξης λογισμικού θα χρησιμοποιήσουν τόσο το CD όσο και το CT στις στρατηγικές ελέγχων τους.

Με την επιλογή του αυτοματοποιημένου ελέγχου, οι επιχειρήσεις είναι σε θέση να εξορθολογήσουν τις διαδικασίες ελέγχων τους για να επιτύχουν τη μέγιστη απόδοση της επένδυσης. Αυτό συμβαίνει επειδή οι αυτοματοποιημένοι έλεγχοι μπορούν να συντομεύσουν τους κύκλους ζωής της ανάπτυξης, να εξαλείψουν την πιθανότητα ανθρώπινου σφάλματος και να αυτοματοποιήσουν τετριμμένες και μονότονες εργασίες. [14]

1.4.2.1 Διαδικασία Αυτοματοποιημένου Ελέγχου

Η διαδικασία αυτοματοποιημένου ελέγχου είναι μια διαδικασία όπου σε πολλούς οργανισμούς, η ποιότητα του λογισμικού / προϊόντος αποτελεί προτεραιότητα. Η κάθε εταιρεία λογισμικού θα πρέπει να είναι καλά εξοικειωμένη με τους μηχανικούς λογισμικού, και ιδιαίτερα με τους μηχανικούς ελέγχου και τους προγραμματιστές. Αυτό θα βοηθήσει την εταιρεία να κατασκευάσει πιο ποιοτικά προϊόντα σε λιγότερο χρόνο και επίσης να είναι σε θέση να τα διαθέσει νωρίς στην αγορά. Σε αυτό το σημείο είναι σημαντικό να κατανοήσουμε ποια είναι τα βήματα για να οργανώσουμε σωστά ένα πλάνο αυτόματων ελέγχων αλλά και ποιο βήμα πρέπει να εκτελεστεί πρώτο και γιατί. Η τήρηση αυτών των βημάτων θα βοηθήσει την εταιρεία να εισαγάγει την αυτοματοποίηση με απρόσκοπτο τρόπο και θα αποφύγει τις συνήθεις παγίδες που οδηγούν σε αποτυχίες της αυτοματοποίησης. Ας περάσουμε σε μια συνοπτική παρουσίαση της διαδικασίας αυτοματοποιημένων ελέγχων, από την επιλογή του εργαλείου ελέγχων έως την εκτέλεση.

Υπάρχουν ορισμένα βήματα που ακολουθούνται κατά τη διαδικασία αυτοματοποιημένων ελέγχων [15]:

1. Το πρώτο βήμα είναι να πειστεί η διοίκηση της επιχείρησης ότι ο αυτοματοποιημένος έλεγχος είναι απαραίτητος για την παροχή ενός ποιοτικού προϊόντος. Στην επιχείρηση, ο έλεγχος δεν είναι μόνο στο χέρι του μηχανικού ελέγχων, επειδή τα εργαλεία ελέγχου αυτοματοποίησης είναι ακριβά και συνδέονται με μεγάλα τέλη χρήσης. Ο μηχανικός ελέγχου από μόνος του δεν μπορεί να τα καταφέρει να πείσει τη διοίκηση για τα οφέλη που προσφέρει η αυτοματοποίηση των ελέγχων. Όπως γνωρίζουμε, η αυτοματοποίηση μερικές φορές κοστίζει αρκετά, επομένως, αφού γίνει μια ανάλυση κόστους και ωφελειών, πρέπει να πειστεί η διοίκηση, προετοιμάζοντας μια λεπτομερή έκθεση σχετικά με τα οφέλη της αυτοματοποίησης των ελέγχων. Πρέπει να τονιστεί το γεγονός ότι τα αποτελέσματα των αυτοματοποιημένων ελέγχων δεν μπορούν να φανούν αμέσως. Χρειάζονται 2-3 μήνες για να φανούν τα αποτελέσματα. Πρέπει να κατανοήσουν ότι είναι θέμα υπομονής για να επιτευχθεί μια εφαρμογή χωρίς λάθη.
2. Το επόμενο βήμα είναι η συμβουλή ή η πρόσληψη ειδικών που γνωρίζουν τα εργαλεία για αυτοματοποιημένο έλεγχο. Οι έλεγχοι αυτοματισμού γίνονται από ειδικούς με εργαλεία αυτοματισμού που γνωρίζουν πώς να τα χρησιμοποιούν για καλύτερα αποτελέσματα. Υπάρχουν δύο ειδικότητες για αυτό το σκοπό, οι αρχιτέκτονες αυτοματισμού (automation architectures) και μηχανικοί αυτοματισμού (automation engineers). Ο αρχιτέκτονας αυτοματισμού δημιουργεί το αυτοματοποιημένο πλαίσιο λογισμικού, δημιουργεί κανόνες για τη δημιουργία και τον σχεδιασμό των σεναρίων και δίνει τα κατάλληλα ονόματα στα αρχεία. Βοηθά τη διοίκηση να αναλύσει τις εφαρμογές και τα εργαλεία που χρησιμοποιούνται στην εφαρμογή, ώστε να μπορέσει να επιλέξει το κατάλληλο εργαλείο για την αυτοματοποίηση. Έχει εμπειρία στη χρήση διαφόρων ειδών εργαλείων, καθώς τα κατανοεί καλά με τα πλεονεκτήματα και τα μειονεκτήματά τους, ώστε να μπορεί να επιλέξει την κατάλληλη περίπτωση ελέγχου που πρέπει να αυτοματοποιηθεί. Ο μηχανικός αυτοματισμού είναι ειδικός στη μετατροπή χειροκίνητων περιπτώσεων ελέγχου σε αυτοματοποιημένα σεναρία ελέγχων και εργάζεται υπό τον αρχιτέκτονα αυτοματισμού. Ο μηχανικός αυτοματισμού πρέπει να είναι καλός στον προγραμματισμό (αντικειμενοστραφής γλώσσα), γεγονός που τον βοηθάει στην κατανόηση, δημιουργία και εκτέλεση των σεναρίων. Οι εταιρείες μπορούν να τους προσλάβουν από το εξωτερικό ή να τους προετοιμάσουν εσωτερικά εκπαιδώντας τους υπάρχοντες μηχανικούς χειροκίνητου ελέγχου τους.
3. Το τρίτο βήμα είναι η επιλογή του κατάλληλου εργαλείου αυτοματοποίησης. Αυτό είναι το πιο σημαντικό και δύσκολο βήμα στη διαδικασία ελέγχων, καθώς η λάθος επιλογή εργαλείου είναι

πιθανό να επιφέρει αρνητικά αποτελέσματα που μπορεί να βλάψουν την επιχείρηση. Πριν από την επιλογή ενός εργαλείου αυτοματοποίησης, πρέπει να γνωρίζει κανείς τα πλεονεκτήματα και τα μειονεκτήματά του. Στη συνέχεια, εξετάζονται και αναλύονται οι επιχειρηματικές ανάγκες και απαιτήσεις πριν ληφθεί η τελική απόφαση. Σημεία που λαμβάνονται υπόψη κατά την επιλογή του εργαλείου είναι μερικά όπως το ότι το εργαλείο πρέπει να βρίσκεται στον προϋπολογισμό της εταιρείας, να υποστηρίζει τις τεχνολογίες που χρησιμοποιούνται στην εφαρμογή, να απαιτούνται εξειδικευμένοι πόροι και τα εργαλεία να διαθέτουν καλό μηχανισμό αναφοράς.

4. Το τέταρτο βήμα είναι η επιλογή της εφαρμογής στην οποία θα χτιστούν τα σενάρια των αυτοματοποιημένων ελέγχων. Για τον αυτοματοποιημένο έλεγχο, είναι σημαντικό να επιλεγθούν οι σωστοί έλεγχοι για αυτοματοποίηση πάνω στη σωστή εφαρμογή και αυτό εξαρτάται από ορισμένους παράγοντες όπως το γεγονός ότι η επιλεγμένη εφαρμογή πρέπει να είναι απαλλαγμένη από σφάλματα μετά από τους χειροκίνητους ελέγχους. Επίσης, η διεπαφή χρήστη (User Interface—UI) της εφαρμογής δεν πρέπει να αλλάζει συχνά και θα ήταν καλύτερα εάν είναι σταθερή, όπως ακόμη η επιλεγμένη εφαρμογή θα πρέπει να βρίσκεται στο αρχικό στάδιο της ανάπτυξής της με σταθερές ενότητες και αποτελέσματα χειροκίνητων ελέγχων.
5. Το αμέσως επόμενο βήμα είναι η εκπαίδευση της ομάδας που είναι υπεύθυνη για την αυτοματοποίηση. Με την τεχνολογική αλλαγή, οι μηχανικοί χειροκίνητου ελέγχου πρέπει να αναβαθμιστούν ώστε να γίνουν μηχανικοί αυτοματισμού. Πρέπει να εκπαιδευτούν στις έννοιες και τις ορολογίες του αυτοματισμού. Η ομάδα του αυτοματοποιημένου ελέγχου πρέπει να εκπαιδευτεί ώστε να χρησιμοποιεί το εργαλείο για ελέγχους και να δίνει τα επιθυμητά αποτελέσματα.
6. Στο έκτο βήμα συναντάμε την ανάπτυξη του πλαισίου λογισμικού για αυτοματοποιημένο έλεγχο. Μετά την επιλογή εργαλείων, εφαρμογών και εκπαίδευσης για την ομάδα αυτοματοποίησης, ήρθε η ώρα να αναπτυχθεί το πλαίσιο λογισμικού για τον αυτοματοποιημένο έλεγχο. Το πλαίσιο λογισμικού συνδυάζει στρατηγικές σχεδιασμού και κανόνες για τη συγγραφή των σεναρίων ελέγχου. Αυτό θα μπορούσε να οδηγήσει στη μικρότερη δυνατή συντήρηση της εφαρμογής και αν υπάρχει ανάγκη για οποιαδήποτε αλλαγή που είναι εύκολο να αντιμετωπιστεί. Περιλαμβάνει σπονδυλωτό (modular), με γνώμονα τα δεδομένα (data-driven), υβριδικό (hybrid), με γνώμονά τις λέξεις-κλειδιά (keyword-driven) και γραμμικό (linear) πλαίσιο λογισμικού.
7. Η προετοιμασία του σχεδίου εκτέλεσης αποτελεί το έβδομο βήμα. Ο αυτοματοποιημένος έλεγχος εκτελείται με ένα σχέδιο εκτέλεσης που περιλαμβάνει την επιλογή ενός περιβάλλοντος στο οποίο θα εκτελεστούν τα προγράμματα υπολογιστών (scripts), συμπεριλαμβανομένου του προγράμματος περιήγησης, του λειτουργικού συστήματος και άλλων διαμορφώσεων στον ηλεκτρονικό υπολογιστή (hardware). Η ομάδα αυτοματισμού εκτελεί το σχέδιο και δηλώνει ποιος θα εκτελέσει τα προγράμματα. Το σχέδιο εκτέλεσης διαφέρει από εταιρεία σε εταιρεία, καθώς ορισμένοι ζητούν από τους προγραμματιστές τους να εκτελέσουν τα σενάρια πριν από την έκδοση αλλά υπάρχουν και ορισμένες περιπτώσεις που τα εκτελούν μετά την έκδοση.
8. Το όγδοο βήμα είναι η συγγραφή των προγραμμάτων. Όταν ολοκληρωθούν οι εργασίες σε επίπεδο βάσης (συμπεριλαμβανομένου του σχεδιασμού του πλαισίου λογισμικού, της εκτέλεσης του σχεδίου και της εκπαίδευσης των μηχανικών ελέγχου στο νέο εργαλείο), έρχεται η ώρα για τη συγγραφή των σεναρίων. Τα σημεία που πρέπει να θυμάται κανείς κατά τη συγγραφή ενός σεναρίου είναι:

- i. Η απώλεια κώδικα θα πρέπει να αποτραπεί διατηρώντας τον σε έναν έλεγχο έκδοσης (source control).
 - ii. Στον έλεγχο έκδοσης θα πρέπει να διατηρείται το ιστορικό των αλλαγών.
 - iii. Κάθε σενάριο θα πρέπει να διατηρείται με διατυπωμένο τρόπο (κατάλληλες συμβάσεις ονοματοδοσίας), για αυτό και η συγγραφή ενός τεκμηρίου (documentation) με όλο τα σενάρια ελέγχου και την περιγραφή τους είναι απαραίτητο να υπάρχει και να έχουν πρόσβαση όλα τα μέλη της ομάδας αυτοματισμού.
 - iv. Ο συγγραφέας σεναρίων πρέπει να γνωρίζει γλώσσες προγραμματισμού, διότι ο έλεγχος αυτοματισμού αποτελεί μέρος της ανάπτυξης λογισμικού.
9. Το ένατο βήμα είναι και το πιο σημαντικό καθώς σχετίζεται με την αναφορά των αποτελεσμάτων μετά την εκτέλεση των σεναρίων. Η αναφορά περιλαμβάνει το αποτέλεσμα του ελέγχου που δημιουργήθηκε στο τέλος της εκτέλεσης. Αποτελείται από όλες τις λεπτομέρειες των ελέγχων και το τι απομένει. Το αποτέλεσμα γράφεται σε πίνακες και διαγράμματα σύμφωνα με τις απαιτήσεις της διοίκησης και αποστέλλεται στη διοίκηση.
10. Το τελευταίο βήμα δεν θα μπορούσε να είναι κάποιο άλλο εκτός από τη συντήρηση των προγραμμάτων. Το βήμα συντήρησης των προγραμμάτων απαιτείται όταν ζητείται μια αλλαγή στην εφαρμογή. Για να αντιμετωπιστούν οι αλλαγές, τα προγράμματα ενημερώνονται αμέσως για να εξασφαλιστεί η άσπρη εκτέλεση. Η τακτική συντήρηση προγραμμάτων είναι σημαντική για την ομαλή λειτουργία της εφαρμογής και τη λειτουργία χωρίς σφάλματα. [15]

1.5 Διαφορές Χειροκίνητου και Αυτοματοποιημένου Ελέγχου

Είναι σημαντικό να γίνεται διάκριση μεταξύ χειροκίνητων και αυτοματοποιημένων ελέγχων, καθώς έχουν δυνατά και αδύνατα σημεία έκαστος. Ο χειροκίνητος έλεγχος πραγματοποιείται αυτοπροσώπως, κάνοντας κλικ στην εφαρμογή ή αλληλοεπιδρώντας με το λογισμικό και τη διεπαφή προγραμματισμού εφαρμογών (Application Programming Interface—API) με τα κατάλληλα εργαλεία. Αυτό είναι πολύ δαπανηρό, καθώς απαιτεί από κάποιον να ρυθμίσει ένα περιβάλλον και να εκτελέσει ο ίδιος τα σενάρια ελέγχου, και μπορεί να είναι επιρρεπές σε ανθρώπινα λάθη, καθώς ο μηχανικός ελέγχου μπορεί να κάνει τυπογραφικά λάθη ή να παραλείψει βήματα στο σενάριο ελέγχου.

Ο αυτοματοποιημένος έλεγχος, από την άλλη πλευρά, εκτελείται από μια μηχανή που εκτελεί ένα σενάριο ελέγχου που έχει γραφτεί εκ των προτέρων. Αυτοί οι έλεγχοι μπορεί να ποικίλουν σε πολυπλοκότητα, από τον έλεγχο μιας μεμονωμένης μεθόδου σε μια κλάση έως τη διασφάλιση ότι η εκτέλεση μιας ακολουθίας πολύπλοκων ενεργειών στο περιβάλλον εργασίας οδηγεί στα ίδια αποτελέσματα. Είναι πολύ πιο ισχυρός και αξιόπιστος από το χειροκίνητο έλεγχο, αλλά η ποιότητα των αυτοματοποιημένων ελέγχων εξαρτάται από το πόσο καλά έχουν γραφτεί τα σενάρια ελέγχου. [16]. Στον Πίνακα 1.5 περιγράφονται οι διαφορές μεταξύ αυτοματοποιημένου και χειροκίνητου ελέγχου.

Πίνακας 1.5: Διαφορές χειροκίνητου και αυτοματοποιημένου ελέγχου.

Τομέας σύγκρισης	Αυτοματοποιημένου Ελέγχου	Χειροκίνητος Ελέγχου
Ορισμός	Όταν μια εφαρμογή ή ένα λογισμικό ελέγχεται χρησιμοποιούνται εργαλεία αυτοματοποίησης. Οι έλεγχοι μπορούν να εκτελεστούν πολλαπλές φορές και	Ο άνθρωπος είναι υπεύθυνος για την εκτέλεση ενός χειροκίνητου ελέγχου, για να ελέγξει τη λειτουργικότητα μιας εφαρμογής

	να καλύψουν μεγάλο ποσοστό των περιπτώσεων ελέγχου.	με βάση τις απαιτήσεις του πελάτη
Αξιοπιστία	Είναι αξιόπιστη επειδή δοκιμάζει την εφαρμογή με τη βοήθεια εργαλείων και βασίζεται σε σενάρια δοκιμών.	Είναι λιγότερο αξιόπιστη, επειδή υπάρχει η πιθανότητα του ανθρώπινου λάθους. Μια μικρή δόση απροσεξίας είναι αρκετή για να μην γίνει σωστή αξιολόγηση και να παραδοθεί η εφαρμογή με κάποιο σφάλματα.
Επαναχρησιμότητα	Ένα σενάριο ελέγχου μπορεί να επαναχρησιμοποιηθεί σε πολλαπλές εκτελέσεις.	Κάθε φορά ο μηχανικός ελέγχου είναι αναγκασμένος να αναπαραστήσει το σενάριο ελέγχου. Οπότε δεν υφίσταται επαναχρησιμότητα.
Ομαδοποιημένη εκτέλεση	Η εκτέλεση ομαδοποιημένων σετ σεναρίων είναι δυνατή με τη χρήση του αυτοματισμού ελέγχου, καθώς όλα τα σενάρια που έχουν γραφτεί μπορούν να εκτελεστούν παράλληλα ή ταυτόχρονα σε διαφορετικά περιβάλλοντα και μηχανήματα.	Η ομαδοποιημένη εκτέλεση δεν μπορεί να εφαρμοστεί χειροκίνητα. Ο κάθε μηχανικός ελέγχου μπορεί να εκτελέσει έναν έλεγχο ανά την φορά.
Εξοικονόμηση χρόνου	Η εκτέλεση είναι πάντα ταχύτερη από τη χειροκίνητη, γι αυτό και η διαδικασία αυτοματοποιημένων ελέγχων μπορεί να εξοικονομήσει πολύτιμο χρόνο.	Είναι χρονοβόρα λόγω της χρήσης των ανθρώπινων πόρων.
Επένδυση	Υπάρχουν εργαλεία αυτοματισμού τα οποία προσφέρονται δωρεάν στην αγορά. Όμως υπάρχουν και εργαλεία για τα οποία απαιτείται συνδρομή ώστε να τα αποκτήσει και να τα χρησιμοποιήσει μια εταιρεία.	Απαιτείται επένδυση μόνο για το ανθρώπινο δυναμικό που προσληφθεί για εκτελεί τα καθήκοντα ενός μηχανικού χειροκίνητου ελέγχου.
Έλεγχος Απόδοσης	Για να ελεγχθεί η απόδοση μιας εφαρμογής με τη βοήθεια των ελέγχων φορτίου (load) και στρεσαρίσματος (stress), ο μηχανικός αυτόματου ελέγχου πρέπει να εφαρμόσει έλεγχο απόδοσης, ο οποίος υποστηρίζεται από εργαλεία αυτόματου ελέγχου.	Στο χειροκίνητο έλεγχο, ο έλεγχος των επιδόσεων δεν είναι δυνατόν να πραγματοποιηθεί.
Γνώσεις προγραμματισμού	Παλαιότερα η γνώση μιας γλώσσας προγραμματισμού ήταν απαραίτητη για να ξεκινήσει κάποιος να γράφει σενάρια αυτόματου ελέγχου. Πλέον	Δεν είναι απαραίτητες οι γνώσεις του προγραμματισμού, αλλά θα πρέπει να κατέχει κανείς καλή γνώση του προϊόντος για να γράφει

	υπάρχουν εργαλεία τα οποία υποστηρίζουν τη δημιουργία σεναρίων χωρίς προγραμματισμό.	και να εκτελέσει σενάρια ελέγχου.
Πλαίσια Λογισμικού	Ο μηχανικός αυτοματοποιημένων ελέγχων μπορεί να χρησιμοποιήσει διαφορους τύπους πλαισίων λογισμικού όπως με γνώμονα τα δεδομένα (data driven), υβριδικά (hybrid), σπονδυλωτά (modular driven) και με γνώμονα τις λέξεις-κλειδιά (keyword-driven) για να επιταχύνει τη διαδικασία αυτοματοποίησης.	Δεν υπάρχει καμία ανάγκη για τη χρήση κάποιου πλαισίου λογισμικού κατά τη εκτέλεση χειροκίνητων ελέγχων.
Συμβατότητα λειτουργικού συστήματος	Οι αυτοματοποιημένοι έλεγχοι μπορούν να εκτελούνται σε διαφορετικά συστήματα με διαφορετικές πλατφόρμες λειτουργικών συστημάτων και σε διάφορες γλώσσες προγραμματισμού.	Η συμβατότητα των λειτουργικών συστημάτων δεν είναι δυνατή στο χειροκίνητο έλεγχο. Για να γίνουν οι έλεγχοι σε διαφορετικά συστήματα, θα πρέπει να υπάρχουν μηχανικοί έλεγχοι που θα εκτελούν τα σενάρια ελέγχου, δουλεύοντας ο καθένας σε διαφορετικό λειτουργικό σύστημα.
Έλεγχος Παλινδρόμησης	Όποτε ανανεώνεται ο κώδικα μια εφαρμογής, ο μηχανικός αυτόματων ελέγχων θέτει σε λειτουργία την εκτέλεση ενός συνόλου αυτόματων σεναρίων.	Όποτε ανανεώνεται ο κώδικα μια εφαρμογής, ο μηχανικός ελέγχου μπορεί να εκτελέσει ένα σύνολο σεναρίων, όμως είναι αρκετά χρονοβόρο.

1.6 Κίνητρο και Στόχος Διπλωματικής

Στο κεφάλαιο αυτό, εξετάζουμε τα κίνητρα που με οδήγησαν στην επιλογή αυτού του θέματος για τη διπλωματική μου εργασία. Αναδεικνύω το πώς η ανάγκη για αυτοματοποιημένους ελέγχους ανακύπτει από τη σύγχρονη ανάπτυξη της τεχνολογίας και τη σημασία του λογισμικού στην καθημερινή μας ζωή. Επίσης, περιγράφω τον στόχο της διπλωματικής εργασίας, που είναι να ανακαλύψω τα οφέλη του αυτοματοποιημένου ελέγχου και την επίδρασή του στην ποιότητα του λογισμικού. Μέσα από αυτό το κεφάλαιο, ανοίγω το παράθυρο για τον λόγο που αφιέρωσα την ενέργειά μου σε αυτήν την έρευνα.

1.6.1 Χρησιμότητα του Αυτοματοποιημένου Ελέγχου

Ορισμένες ομάδες απλά δεν έχουν το χρόνο ή τους πόρους για να δοκιμάζουν χειροκίνητα το λογισμικό. Η αυτοματοποίηση μπορεί να βοηθήσει σε αυτό. Μπορεί να **μειώσει σημαντικά τον χρόνο** που απαιτείται για τον έλεγχο του λογισμικού, επειδή εκτελείται γρήγορα και αποτελεσματικά [17]. Έτσι, οι προγραμματιστές και οι διαχειριστές παραγωγής κερδίζουν πολύτιμο χρόνο και μπορούν να στρέψουν τις προσπάθειές τους σε άλλες πτυχές του προϊόντος. Ως αποτέλεσμα, μπορεί να συμβάλει σημαντικά στην **ενίσχυση της παραγωγικότητας**. [17]

Η χρήση της τεχνολογίας του αυτοματισμού σημαίνει, επίσης, ότι οι έλεγχοι **μπορούν να γίνονται συχνότερα**, βελτιώνοντας τη συνολική λειτουργικότητα. Οι κύκλοι ανάπτυξης λογισμικού απαιτούν επαναλαμβανόμενους ελέγχους, συχνά τον ίδιο έλεγχο ξανά και ξανά. Ο αυτοματοποιημένος έλεγχος το καθιστά αυτό δυνατό, χωρίς να σπαταλάει τον χρόνο από τα μέλη της ομάδας που έχουν άλλες εργασίες. Ακόμη, μπορεί να προσφέρει **πιο ακριβή και αξιόπιστα αποτελέσματα** από ό,τι μόνο ο χειροκίνητος έλεγχος. Εξασφαλίζοντας περαιτέρω ότι το προϊόν είναι έτοιμο για την αγορά ή για να προχωρήσει στο επόμενο στάδιο ανάπτυξης. Αυτή η επικύρωση δίνει ώθηση στα μέλη της ομάδας να συνεχίσουν την δουλειά και την ανάπτυξη.

Το πιο σημαντικό είναι ότι η αυτοματοποίηση είναι ωφέλιμη για την ανάπτυξη λογισμικού. Αυτό συμβαίνει επειδή όταν ένα λογισμικό ή μια εφαρμογή ή ένα άλλο προϊόν μπορεί να σχεδιαστεί και να παραχθεί πιο αποτελεσματικά, ανοίγει το δρόμο για την εφαρμογή της συνεχής ανάπτυξης του όταν δημοσιευθεί. Ουσιαστικά, η επιχείρηση θα μπορεί να εργάζεται στην ανάπτυξη περισσότερων λογισμικών και προϊόντων, ακόμη και με τον ίδιο αριθμό μελών της ομάδας, χάρη στην αυτοματοποίηση. Αυτό δεν σημαίνει μόνο ότι τελειοποιούνται τα τελικά προϊόντα που βγάζουν, αλλά σημαίνει επίσης ότι δημιουργείται συνεχώς νέο λογισμικό.

Ο έλεγχος λογισμικού έχει πολλά **οφέλη**, γι' αυτό και οι επιχειρήσεις τύπου Λογισμικό ως Υπηρεσία (Software as a Service—SaaS) σε όλο τον κόσμο χρησιμοποιούν την τεχνολογία αυτοματοποίησης. Ακολουθούν μερικά από τα μεγαλύτερα οφέλη της χρήσης του αυτοματοποιημένου ελέγχου για την ανάπτυξη λογισμικού:

- *Δυνατότητες λεπτομερείς αναφοράς:* Οι αυτοματοποιημένοι έλεγχοι χρησιμοποιούν καλά σχεδιασμένα σενάρια ελέγχου. Αυτές οι σεναριακές ακολουθίες μπορούν να είναι απίστευτα εμπειρισταωμένες και να παρέχουν λεπτομερείς αναφορές που απλά δεν θα ήταν δυνατόν να γίνουν από έναν άνθρωπο.
- *Βελτιωμένη ανίχνευση σφαλμάτων:* Ένας από τους κύριους λόγους για τον έλεγχο ενός προϊόντος είναι η ανίχνευση σφαλμάτων και άλλων ελαττωμάτων. Ο αυτοματοποιημένος έλεγχος καθιστά τη διαδικασία αυτή ευκολότερη. Επίσης, είναι σε θέση να αναλύσει και να καλύψει ένα μεγάλο εύρος από σενάρια ελέγχου, σε σύγκριση με ό,τι μπορεί να καλύψει ένας άνθρωπος.
- *Απλοποίηση του Ελέγχου Λογισμικού:* Οι έλεγχοι αποτελούν μέρος της ρουτίνας των εργασιών των περισσότερων εταιρειών SaaS και τεχνολογίας. Το κλειδί της επιτυχίας βρίσκεται στο πόσο απλοϊκή μπορεί να γίνει η διεξαγωγή τους. Η χρήση αυτοματοποίησης είναι εξαιρετικά επωφελής, καθώς με τη χρήση εργαλείων για αυτοματοποιημένους ελέγχους, τα σενάρια ελέγχου μπορούν να επαναχρησιμοποιηθούν. Αντιθέτως, στο χειροκίνητο έλεγχο δεν ισχύει κάτι τέτοιο.
- *Επιτάχυνση στη διαδικασία ελέγχων:* Ο αυτοματοποιημένος έλεγχος λειτουργεί ταχύτερα συγκριτικά με τους ανθρώπους. Σε συνδυασμό με τη βελτιωμένη ακρίβεια, είναι οι κύριοι λόγοι για τους οποίους το χρησιμοποιούμε. Αυτό με τη σειρά του, συντομεύει τους κύκλους ανάπτυξης του λογισμικού.
- *Μείωση της ανθρώπινης παρέμβασης:* Οι έλεγχοι μπορούν να εκτελούνται οποιαδήποτε ώρα της ημέρας, ακόμη και κατά τη διάρκεια της νύχτας, χωρίς να απαιτείται ανθρώπινη επίβλεψη ή παρέμβαση. Επιπλέον, η αυτόματη διεξαγωγή, συμβάλλει στη μείωση του κινδύνου του ανθρώπινου λάθους.
- *Εξοικονόμηση χρόνου και χρήματος:* Οι έλεγχοι μπορεί να είναι χρονοβόροι. Αν και η αυτοματοποίηση μπορεί να απαιτεί μια αρχική επένδυση, μπορεί να εξοικονομήσει χρήματα μακροπρόθεσμα και να γίνει πιο αποδοτική. Τα μέλη της ομάδας χρησιμοποιούν τον χρόνο τους

σε άλλους τομείς και δεν απαιτείται πλέον να πραγματοποιούν χειροκίνητους ελέγχους σε πολλές περιπτώσεις. Αυτό βελτιώνει τη ροή εργασίας τους.

Οι επιχειρήσεις θα πρέπει να χρησιμοποιούν την αυτοματοποίηση για να βελτιώσουν τις επιχειρηματικές τους διαδικασίες και τα λειτουργικά τους συστήματα, ιδίως εκείνες που δραστηριοποιούνται στον κλάδο της τεχνολογίας. Η αυτοματοποίηση παρέχει πολύτιμα εργαλεία για τις επιχειρήσεις που μπορούν να χρησιμοποιήσουν προς όφελός τους, είτε πρόκειται για τη βελτίωση των χρόνων παράδοσης των προϊόντων είτε για την τήρηση των αυξανόμενων προτύπων ασφαλείας.

Μόλις καθοριστεί το ποιος έλεγχος θα χρησιμοποιηθεί, πρέπει να τεθούν στόχοι ως σημείο αναφοράς για να ελεγχθεί η απόδοση. Χωρίς τον καθορισμό στόχων, θα είναι δύσκολο να αξιοποιηθεί στο έπακρο το αποτέλεσμα του ελέγχου. Η προσοχή πρέπει να μείνει επικεντρωμένη σε αυτόν τον ένα στόχο και μην υπάρχει δισταγμός για την εκτέλεση ξεχωριστών ελέγχων όπου χρειάζονται. Οφείλει κανείς να σκεφτεί τι είναι αυτό που προσπαθεί να επιτύχει και πώς αυτός ο έλεγχος μπορεί να τον βοηθήσει να το πετύχει.

Οι έλεγχοι πρέπει να διαιρεθούν σε μικρότερους λογικούς ελέγχους. Οι μεγαλύτεροι, πιο πολύπλοκοι έλεγχοι είναι πιο δύσκολο να εκτελεστούν. Τα μέλη της ομάδας που δεν γράφουν κώδικα ελέγχων μπορούν να μετακινηθούν σε άλλους τομείς της διαδικασίας ανάπτυξης του προϊόντος, ώστε να αξιοποιήσουν καλύτερα τον χρόνο τους. Η αξιοποίηση της αυτοματοποίησης έχει να κάνει με την ευκολότερη πραγματοποίηση ελέγχων και τη βελτίωση των επιχειρηματικών πρακτικών. [17]

Κεφάλαιο 2ο: Βιβλιογραφική Ανασκόπηση

Στο κεφάλαιο 2.1 παρουσιάζονται εργαλεία αυτοματοποιημένου ελέγχου που υπάρχουν στη βιβλιογραφία, καθώς στο κεφάλαιο 2.2 παρουσιάζεται η σχετική βιβλιογραφία στον αυτοματοποιημένο έλεγχο.

2.1 Πλαίσια Λογισμικού και Εργαλεία Αυτοματοποιημένου Ελέγχου

Ένα Πλαίσιο Λογισμικού για Έλεγχο (Testing Framework) διαθέτει ένα σύνολο κατευθυντήριων γραμμών για τους επαγγελματίες που περιλαμβάνει πρότυπα κώδικα, διαχείριση των αποθηκευτικών αποθετηρίων (repositories) και χειρισμό των δεδομένων ελέγχου (testing data), ώστε να επιτυγχάνονται ωφέλιμα αποτελέσματα όπως η εύκολη επαναχρησιμοποίηση του κώδικα, ο μειωμένος χρόνος διαχείρισης σεναρίων και η υψηλή φορητότητα.

Οι τύποι των πλαισίων λογισμικού για έλεγχο διακρίνονται στους εξής:

- *Γραμμικό Πλαίσιο Λογισμικού (Linear Framework)*: Το γραμμικό πλαίσιο λογισμικού αποτελεί την απλούστερη κατηγορία πλαισίων λογισμικού από όλες. Σύμφωνα με αυτό, πρέπει κανείς να γράψει έναν απλό κώδικα για την εκτέλεση των περιπτώσεων ελέγχου χωρίς καμία σπονδυλωτή δομή ή διαδοχικά βήματα. Λειτουργεί ως μοντέλο εγγραφής και αναπαραγωγής (record-and-playback).
- *Πλαίσιο Λογισμικού βασισμένο σε Λέξεις-κλειδιά (Keyword-driven Framework)*: Το πλαίσιο λογισμικού που είναι βασισμένο σε λέξεις-κλειδιά, πρόκειται για μια τεχνική σεναρίων που συνθέτει λέξεις κλειδιά με ορισμένες ενέργειες, όπως το άνοιγμα ή το κλείσιμο ενός προγράμματος περιήγησης, συμβάντα κλικ του ποντικιού και άλλα. Στα σεναρία ελέγχων, μπορεί κανείς να καλέσει αυτές τις λέξεις κλειδιά για να εκτελέσει ένα συγκεκριμένο βήμα. Επίσης, θα έχει ένα αρχείο που θα αποθηκεύονται όλες οι λέξεις κλειδιά, μαζί με τις αντίστοιχες ενέργειες που εκτελούν.
- *Πλαίσιο Λογισμικού με γνώμονα τα Δεδομένα (Data-driven Framework)*: Σε ένα πλαίσιο λογισμικού με γνώμονα τα δεδομένα, όλα τα δεδομένα εισόδου των σεναρίων ελέγχου αποθηκεύονται σε πίνακα ή σε αρχεία επέκτασης, όπως για παράδειγμα .xls, .xml, .csv κ.λπ. Κατά την εκτέλεση των σεναρίων ελέγχου, θα διαβάζει τις τιμές από τον πίνακα. Με τη βοήθεια αυτού του πλαισίου λογισμικού, μπορούν να εκτελεστούν τόσο θετικά όσο και αρνητικά σεναρία ελέγχου.
- *Πλαίσιο Λογισμικού Μοντέλου Αντικειμένων Σελίδας, (Page Object Model Framework)*: Στο πλαίσιο λογισμικού μοντέλου αντικειμένων σελίδας, πρέπει να δημιουργηθεί μια αποθήκη αντικειμένων για τα στοιχεία του UI. Με τον τρόπο αυτό δίνεται η δυνατότητα στον μηχανικό ελέγχου να κλείσει αυτές τις μεθόδους αργότερα χωρίς να χρειάζεται να γράψει ξανά τον κώδικα. Έτσι, έχει ως αποτέλεσμα λιγότερη περιττολογία, δυνατότητα επαναχρησιμοποίησης του κώδικα και μειωμένη κατανάλωση χρόνου για τη συγγραφή σεναρίων ελέγχου.
- *Σπονδυλωτό Πλαίσιο Λογισμικού (Modular Framework)*: Το σπονδυλωτό πλαίσιο λογισμικού επιτρέπει να διαιρούνται τα σεναρία ελέγχου σε μικρές, ανεξάρτητες ενότητες. Αυτές οι ενότητες θα αλληλεπιδρούν μεταξύ τους με ιεραρχικό τρόπο για την εκτέλεση μεγαλύτερων σεναρίων ελέγχου. Αυτό θα βοηθήσει επίσης στο να δημιουργηθούν τα απαιτούμενα σεναρία ελέγχου και να εκτελεστούν τα σεναρία ξεχωριστά.

- *Υβριδικό Πλαίσιο Λογισμικού (Hybrid Automation Framework)*: Το υβριδικό πλαίσιο λογισμικού αυτοματισμού είναι πάντα ένας συνδυασμός μεταξύ των πλαισίων λογισμικών που είναι βασισμένα σε λέξεις-κλειδιά και με γνώμονα τα δεδομένα, στα οποία τα δεδομένα ελέγχου και οι λέξεις κλειδιά είναι εξωτερικά. Τα δεδομένα ελέγχου αποθηκεύονται σε ένα αρχείο τύπου excel, ενώ οι λέξεις κλειδιά διατηρούνται σε ένα ξεχωριστό αρχείο τύπου java. [18]

Ένα εργαλείο αυτοματισμού θα πρέπει να επιλέγεται με βάση τον τύπο των ελέγχων και τον τύπο του πλαισίου λογισμικού που πρόκειται να εφαρμοστούν. Υπάρχουν πολλά εργαλεία διαθέσιμα στην αγορά για να επιλέξουν οι εταιρείες ανάπτυξης λογισμικού, ανάλογα με τις απαιτήσεις τους. Μερικά από τα πιο ευρέως χρησιμοποιούμενα εργαλεία αυτοματοποίησης παρατίθενται παρακάτω.

- Selenium: Όταν πρόκειται για αυτοματοποίηση, το Selenium είναι ένα από τα πιο δημοφιλή εργαλεία αυτοματοποιημένων ελέγχων ανοικτού κώδικα που χρησιμοποιούνται για τον έλεγχο διαφόρων εφαρμογών ιστού. Πρόκειται για ένα προσαρμόσιμο εργαλείο που διατίθεται σε πολλές γλώσσες και προγράμματα περιήγησης παγκοσμίως. Αυτό το πρωταρχικό εργαλείο χρησιμοποιείται από αναλυτές ποιότητας (από νέους μηχανικούς ελέγχου έως διευθύνων μηχανικούς ελέγχου για τη διενέργεια αυτοματοποιημένων ελέγχων. [19]
- Appium: Ειδικά σχεδιασμένο για τη δοκιμή εφαρμογών για κινητά (πλατφόρμες Android και iOS). Πρόκειται για μια αρχιτεκτονική διακομιστή με βάση τον πελάτη που υποστηρίζει πολλαπλές γλώσσες προγραμματισμού για τη συγγραφή ελέγχων, όπως JavaScript, PHP, Java, Python κ.λπ. Παρέχει ελέγχους πολλαπλών πλατφορμών και δυνατότητα επαναχρησιμοποίησης του κώδικα στους ελέγχους και μπορεί επίσης να καταγράφει τις χειρονομίες ως κώδικα. [20]
- Katalon: Το Katalon είναι ένα δωρεάν εργαλείο προγραμμάτων περιήγησης (cross-browser) με άδεια χρήσης που αναπτύχθηκε το 2015. Χρησιμοποιείται για αυτοματοποιημένους ελέγχους διεπαφών ιστού, API και κινητών (iOS και Android). Το εργαλείο αυτό βασίζεται στα πλαίσια λογισμικού αυτοματοποίησης του Appium και του Selenium. Με το Katalon, οι ελεγκτές μπορούν να εκτελούν τοπικούς και απομακρυσμένους ελέγχους και να υποστηρίζουν διαδοχικές και παράλληλες εκτελέσεις. Ένα χαρακτηριστικό γνώρισμα είναι μια διπλή διεπαφή σεναρίων που κάνει έναν χρήστη χωρίς δεξιότητες κωδικοποίησης να χρησιμοποιεί μια απλή διεπαφή. [21]
- Cucumber: Τεχνικά, το εργαλείο αυτοματοποίησης Cucumber είναι η σωστή επιλογή για τους αναλυτές ελέγχου ποιότητας που εστιάζουν στην εμπειρία του τελικού χρήστη πάνω από όλους τους παράγοντες. Ως εκ τούτου, ονομάζεται εργαλείο ανάπτυξης με γνώμονα τη συμπεριφορά. Πρόκειται για ένα εργαλείο αυτοματισμού ανοικτού κώδικα που υποστηρίζει διάφορες γλώσσες όπως Scala, Java, groovy, Ruby κ.λπ. Στο Cucumber, ένας έλεγχος μπορεί να αποκτήσει υποστήριξη από άκρη σε άκρη του πλαισίου λογισμικού, όπου ο κώδικας ελέγχων γράφεται σε μια απλή γλώσσα γνωστή ως Gherkin, η οποία μπορεί να εκτελεστεί σε ένα διαφορετικό πλαίσιο λογισμικού. Είναι το καλύτερο εργαλείο από επιχειρηματική άποψη, καθώς υποστηρίζει την ανάπτυξη με γνώμονα τη συμπεριφορά, στην οποία οι επιχειρησιακοί αναλυτές και οι ιδιοκτήτες προϊόντων γράφουν σενάρια ελέγχου για την ενεργοποίηση της συμπεριφοράς του συστήματος. [22]
- Cypress: Αυτό το εργαλείο αυτοματοποίησης επικεντρώνεται σε ελέγχους από άκρο σε άκρο με σύγχρονα JavaScript πλαίσια λογισμικού. Επίσης, διαθέτει διάφορα χαρακτηριστικά, όπως είναι η παροχή λεπτομερής και αναλυτικής τεκμηρίωση, έχει ενσωματωμένο μεγάλο αριθμό βιβλιοθηκών, χειρισμό Μοντέλο Αντικειμένου Εγγράφου (Document Object Model—DOM) και Μοντέλο Αντικειμένου Εγγράφου σκιάς (shadow DOM). Με το DOM μπορούν να

δημιουργηθούν ισχυρά σενάρια ελέγχου από άκρο σε άκρο με γρήγορη δημιουργία και εκτέλεση ελέγχων με φωτισμό. [23]

2.2 Σχετική Βιβλιογραφία

Οι Kong, P., Li, L., Gao, J., Liu, K., Bissyandé, T. F., & Klein, J. [24] πραγματοποίησαν μια συστηματική χαρτογράφηση η οποία παρέχει μια επισκόπηση των ευρημάτων και των συζητήσεων σχετικά με τον αυτοματοποιημένο έλεγχο πάνω σε εφαρμογές Android. Η έρευνα ανέλυσε 103 εργασίες που δημοσιεύτηκαν σε σημαντικά συνέδρια, και περιοδικά στους τομείς της μηχανικής λογισμικού, των γλωσσών προγραμματισμού και της ασφάλειας. Η έρευνα διερεύνησε τις τάσεις στους τύπους ελέγχων, τα επίπεδα ελέγχων, τις μεθοδολογίες και τους στόχους. Αξιολόγησε επίσης τις συνεισφορές των συγγραφέων, συζήτησε τη χρηστικότητα των ερευνητικών αποτελεσμάτων και ανέδειξε τα ανοιχτά ζητήματα, τις προκλήσεις και τις νέες ερευνητικές κατευθύνσεις στον τομέα.

Η ανάλυση των τύπων ελέγχων αποκάλυψε ότι οι έλεγχοι μαύρου κουτιού και γκρίζου κουτιού κυριαρχούν στις ερευνητικές εργασίες, αντιπροσωπεύοντας το 90% των δημοσιεύσεων. Ο έλεγχος συστήματος χρησιμοποιήθηκε σε μεγάλο βαθμό, αντανακλώντας την πολυπλοκότητα των εφαρμογών Android και τις προκλήσεις στην απομόνωση των συστατικών στοιχείων για ελέγχους μονάδας και παλινδρόμησης. Ο έλεγχος βάσει μοντέλου αναδείχθηκε ως η κυρίαρχη μεθοδολογία, με τις περισσότερες προσεγγίσεις να ξεκινούν με μοντελοποίηση GUI ή δημιουργία γραφήματος κλήσεων. Οι γραφικές διεπαφές χρήστη και οι μηχανισμοί συμβάντων παρέμειναν πρωταρχικοί στόχοι των ερευνητικών προσεγγίσεων, με τις ανησυχίες για την ασφάλεια να προσελκύουν σημαντική προσοχή. Ωστόσο, η εστίαση στην ασφάλεια έχει κάπως μειωθεί τα τελευταία χρόνια.

Η αξιολόγηση των συγγραφέων έδειξε ότι η ερευνητική κοινότητα των ελέγχων σε Android παρουσιάζει έναν μεγάλο δείκτη κινητικότητας συγγραφέων κάθε χρόνο, παρόλα αυτά ο αριθμός των συγγραφέων που δημοσιεύουν τις εργασίες τους για το συγκεκριμένο θέμα είναι περιορισμένος. Αυτό υποδηλώνει ότι η έρευνα στον τομέα του ελέγχου εφαρμογών Android συχνά καθοδηγείται από συγκεκριμένες ευκαιρίες και προκλήσεις. Διαπιστώθηκε ότι η διαθεσιμότητα επαναχρησιμοποιήσιμων εργαλείων και λύσεων για ελέγχους ανοικτού κώδικα είναι περιορισμένη, αν και ορισμένα ερευνητικά εργαλεία έχουν αξιοποιηθεί με επιτυχία στη βιομηχανία.

Εντοπίστηκαν διάφορα ανοιχτά ζητήματα και προκλήσεις, συμπεριλαμβανομένης της ανάγκης δημιουργίας σεναρίων ελέγχου που ικανοποιούν απαιτητικές προϋποθέσεις και μοντελοποιούν αποτελεσματικά πολύπλοκα συμβάντα. Επισημάνθηκαν, επίσης η γεφύρωση του χάσματος μεταξύ ασύμβατων συνόλων εντολών και η ανάπτυξη αξιόπιστων εκτιμητών κάλυψης για δίκαιη σύγκριση μεταξύ των προσεγγίσεων ελέγχου. Η αντιμετώπιση των ελαττωμάτων ευχρηστίας και η εκτίμηση της εμπειρίας του χρήστη στους ελέγχους προσδιορίστηκαν ως σημαντικοί τομείς που χρειάζονται βελτίωση.

Το άρθρο πρότεινε νέες ερευνητικές κατευθύνσεις για την προώθηση των ελέγχων εφαρμογών Android. Αυτές περιλαμβάνουν την επικύρωση των ενημερώσεων μιας εφαρμογής για τον εντοπισμό των κινδύνων για ελαττώματα ή μη ασφαλών ενημερώσεων, την εκτίμηση του κατακερματισμού του οικοσυστήματος στις προσεγγίσεις ελέγχων και την ιεράρχηση στην επιλογή του κώδικα και των σεναρίων ελέγχου για τη βελτίωση της αποδοτικότητας και της αποτελεσματικότητας. Αυτές οι ερευνητικές κατευθύνσεις αποσκοπούν στη βελτίωση της ποιότητας, της αξιοπιστίας και της εμπειρίας των χρηστών των εφαρμογών Android. [24]

Οι Rafi, D. M., Moses, K. R. K., Petersen, K., & Mäntylä, M. V. [25] παρουσίασαν μια ολοκληρωμένη έρευνα σχετικά με τα οφέλη και τους περιορισμούς των αυτοματοποιημένων ελέγχων

λογισμικού τόσο από την ακαδημαϊκή όσο και από την βιομηχανική κοινότητα. Οι συγγραφείς διεξήγαγαν συστηματική βιβλιογραφική ανασκόπηση για να αναλύσουν τις ακαδημαϊκές απόψεις και μια έρευνα με 115 επαγγελματίες στον τομέα της ανάπτυξης λογισμικού για να αξιολογήσουν τις απόψεις τους.

Η ανασκόπηση της βιβλιογραφίας αποκάλυψε ότι τα στοιχεία σχετικά με τα οφέλη και τους περιορισμούς των αυτοματοποιημένων ελέγχων λογισμικού είναι επιφανειακά, καθώς μόνο 25 ερευνητικές εργασίες παρέχουν σχετικές πληροφορίες. Παρατηρήθηκε ότι τα οφέλη υποστηρίζονται συχνά από ισχυρότερες πηγές αποδείξεων, όπως πειράματα και μελέτες περιπτώσεων, ενώ οι περιορισμοί αναφέρονται συχνότερα μέσω εμπειρικών αναφορών. Αυτό υποδηλώνει μια πιθανή μεροληψία δημοσίευσης προς τα θετικά αποτελέσματα. Οι συγγραφείς προτείνουν περαιτέρω εμπειρικές μελέτες, συμπεριλαμβανομένων περιπτώσιολογικών μελετών και πειραμάτων, για να διερευνήσουν σε μεγαλύτερο βάθος τους περιορισμούς του αυτοματοποιημένου ελέγχου.

Η έρευνα για τους προγραμματιστές αποσκοπούσε στον προσδιορισμό της συνάφειας των αναγνωρισμένων πλεονεκτημάτων και περιορισμών στη βιομηχανία της ανάπτυξης λογισμικού. Η έρευνα έλαβε 115 έγκυρες απαντήσεις από συμμετέχοντες με βιομηχανική εμπειρία. Τα αποτελέσματα υποστήριξαν σθεναρά τα οφέλη της αυτοματοποίησης ελέγχων, συμπεριλαμβανομένης της υψηλής επαναληψιμοποίησης περιπτώσεων ελέγχου, της επαναληψιμότητας, της βελτιωμένης κάλυψης των ελέγχων και της εξοικονόμησης προσπάθειας κατά την εκτέλεση των ελέγχων. Ωστόσο, οι επαγγελματίες αναγνώρισαν επίσης περιορισμούς όπως η υψηλή αρχική επένδυση, τα ζητήματα προσαρμογής των εργαλείων, οι προκλήσεις συντήρησης και η αδυναμία εύρεσης σύνθετων ελαττωμάτων. Τα ευρήματα της έρευνας υποδεικνύουν διάφορες σημαντικές ερευνητικές κατευθύνσεις, όπως η ανάπτυξη εργαλείων με εύκολη καμπύλη εκμάθησης, η βελτίωση της συντηρησιμότητας και της ανθεκτικότητας των σεναρίων ελέγχου και η διασφάλιση της συμβατότητας σε διάφορα περιβάλλοντα ανάπτυξης λογισμικού.

Οι κίνδυνοι της εγκυρότητας εξετάστηκαν τόσο στη βιβλιογραφική ανασκόπηση όσο και στην έρευνα. Ελήφθησαν μέτρα για την ελαχιστοποίηση της ερμηνευτικής προκατάληψης στη βιβλιογραφική ανασκόπηση μέσω στοχευμένων κριτηρίων για την επιλογή μελετών και συζητήσεων από αναλυτές. Η έρευνα δοκιμάστηκε για την κατανόησή της και καταβλήθηκαν προσπάθειες να συγκεντρωθούν απαντήσεις από διάφορους τομείς, αν και ενδέχεται να εξακολουθεί να υπάρχει κάποια μεροληψία προς ορισμένους πληθυσμούς. Σημειώθηκε επίσης ότι ο σκοπός της αυτοματοποίησης (εκτέλεση των ελέγχων, δημιουργία και επιλογή περιπτώσεων ελέγχου, ανάλυση των ελέγχων) δεν διαφοροποιήθηκε, γεγονός που θα μπορούσε να επηρεάσει τα αποτελέσματα.

Συμπερασματικά, η παρούσα μελέτη συμβάλλει στη γεφύρωση του χάσματος μεταξύ της ακαδημαϊκής έρευνας και της βιομηχανικής εμπειρίας στον τομέα των αυτοματοποιημένων ελέγχων λογισμικού. Τα αποτελέσματα αναδεικνύουν τα οφέλη και τους περιορισμούς του αυτοματοποιημένου ελέγχου, με τους επαγγελματίες να υποστηρίζουν σε μεγάλο βαθμό τα οφέλη αλλά και να αναγνωρίζουν τις προκλήσεις. Παρατηρήθηκαν διαφορές μεταξύ των απόψεων των ακαδημαϊκών και των επαγγελματιών, ιδίως όσον αφορά την ανίχνευση σφαλμάτων και το όραμα των πλήρως αυτοματοποιημένων ελέγχων. Τα ευρήματα υπογραμμίζουν τη σημασία του πλαισίου και του σκοπού στην αυτοματοποίηση και προτείνουν τομείς για μελλοντική έρευνα ώστε να αντιμετωπιστούν οι περιορισμοί που εντοπίστηκαν από τους επαγγελματίες. [25]

Οι MACAKOĞLU, Ş. S., & Peker, S. [26] διεξήγαγαν μια συστηματική βιβλιογραφική ανασκόπηση σχετικά με τη χρήση αυτοματοποιημένων εργαλείων στην έρευνα προσβασιμότητας στο διαδίκτυο. Οι ερευνητές εξέτασαν εβδομήντα δύο (72) άρθρα που δημοσιεύτηκαν μεταξύ 2002 και 2021, τα οποία

ελήφθησαν από διάφορες βάσεις δεδομένων. Τα άρθρα κατηγοριοποιήθηκαν με βάση τον τύπο δημοσίευσης, το έτος δημοσίευσης, τον εκδότη, το εργαλείο αυτοματοποιημένου ελέγχου, τον τύπο ιστότοπου και τον οδηγό προσβασιμότητας. Τα ευρήματα αποκάλυψαν μια αυξανόμενη τάση στη χρήση εργαλείων αυτοματοποιημένων ελέγχων για αξιολογήσεις προσβασιμότητας στο διαδίκτυο με την πάροδο του χρόνου. Η μελέτη είχε ως στόχο να παρέχει μια επισκόπηση της έρευνας σχετικά με την ανάλυση προσβασιμότητας στο διαδίκτυο χρησιμοποιώντας αυτοματοποιημένα εργαλεία ελέγχων και να χρησιμεύσει ως οδικός χάρτης για μελλοντική έρευνα σε αυτόν τον τομέα. Οι ερευνητές τόνισαν τη σημασία της προσβασιμότητας στο διαδίκτυο για την παροχή ίσης πρόσβασης και ικανοποίησης των χρηστών για όλα τα άτομα, συμπεριλαμβανομένων εκείνων με αναπηρίες.

Τα ευρήματα της μελέτης παρέχουν αρκετές σημαντικές πληροφορίες, όπως το γεγονός ότι ο αριθμός των δημοσιεύσεων σχετικά με την προσβασιμότητα στο διαδίκτυο αυξάνεται με την πάροδο των ετών, γεγονός που υποδηλώνει το αυξανόμενο ενδιαφέρον για τον τομέα. Η τάση αυτή υποδηλώνει ότι μελλοντικές έρευνες στην ανάλυση προσβασιμότητας ιστού με τη χρήση εργαλείων αυτοματοποιημένου ελέγχου είναι πιθανό να συνεχίσουν να επεκτείνονται.

Η πλειονότητα των μελετών της ανασκόπησης επικεντρώθηκε σε αξιολογήσεις προσβασιμότητας ιστοτόπων στον εκπαιδευτικό τομέα. Αυτό υπογραμμίζει την ανάγκη για περισσότερη έρευνα σε άλλους τομείς και ενθαρρύνει τους ερευνητές να διερευνήσουν τις αξιολογήσεις προσβασιμότητας ιστού σε διαφορετικούς τομείς.

Τα εργαλεία αυτοματοποιημένων ελέγχων που χρησιμοποιούνται ευρέως στις αξιολογήσεις προσβασιμότητας στο διαδίκτυο, με τα εργαλεία AChecker και WAVE να είναι τα πλέον προτιμώμενα. Οι πληροφορίες αυτές μπορούν να καθοδηγήσουν τους επαγγελματίες και τους ερευνητές στην επιλογή των κατάλληλων εργαλείων αυτοματοποιημένων ελέγχων για τις αναλύσεις τους.

Η ανάλυση αποκάλυψε ότι οι ιστότοποι στην Ασία και τη Βόρεια Αμερική αξιολογούνται συχνά ως προς την προσβασιμότητα, γεγονός που υποδηλώνει ότι οι μελλοντικές μελέτες θα πρέπει να εξετάσουν την τους ιστότοπους και από άλλες γεωγραφικές τοποθεσίες, όπως η Αφρική και η Ευρώπη.

Ο μέσος αριθμός των ιστότοπων που αναλύθηκαν στα άρθρα που εξετάστηκαν ήταν περίπου 72, με διάμεση τιμή 41. Αυτό υποδηλώνει ότι τα μεγέθη δείγματος περίπου 40 έως 70 ιστοτόπων είναι επαρκή για τη διεξαγωγή αναλύσεων προσβασιμότητας στον ιστό σε συγκεκριμένους τομείς.

Τέλος, τα ευρήματα αυτά προσφέρουν πολύτιμες γνώσεις στους επαγγελματίες και τους ερευνητές που εργάζονται στον τομέα της ανάλυσης προσβασιμότητας στον ιστό, παρέχοντας καθοδήγηση σχετικά με τα εργαλεία αυτοματοποιημένων ελέγχων και επισημαίνοντας τομείς για περαιτέρω έρευνα. Ωστόσο, είναι σημαντικό να σημειωθεί ότι η μελέτη αυτή επικεντρώθηκε σε αγγλόφωνα άρθρα που δημοσιεύτηκαν μεταξύ 2002 και 2021 και χρησιμοποίησε συγκεκριμένες λέξεις-κλειδιά. Μελλοντικές μελέτες θα πρέπει να εξετάσουν το ενδεχόμενο να συμπεριλάβουν μη αγγλικά άρθρα και να χρησιμοποιήσουν ένα ευρύτερο φάσμα λέξεων-κλειδιών για μια πιο ολοκληρωμένη βιβλιογραφική ανασκόπηση. [26]

Οι Méndez, A., Quesada, L. C. U., & Jenkins, C. M. [27] ασχολήθηκαν με τις προκλήσεις και τις προσεγγίσεις στην αυτοματοποίηση των ελέγχων κινητών εφαρμογών. Τονίζουν ότι οι αυτοματοποιημένοι έλεγχοι είναι πολύπλοκοι και δύσκολοι λόγω παραγόντων όπως η ποικιλία των καταχωρήσεων (inputs) που απαιτούν οι κινητές εφαρμογές και η ετερογένεια των τεχνολογιών που χρησιμοποιούνται. Η έρευνα παρουσιάζει μια συστηματική χαρτογράφηση και βιβλιογραφική

ανασκόπηση, εντοπίζοντας, χαρτογραφώντας και αναλύοντας προσεγγίσεις, τεχνικές και εμπειρικές αξιολογήσεις αυτοματοποιημένων ελέγχων πάνω σε εφαρμογές κινητών τηλεφώνων.

Τα αποτελέσματα της χαρτογράφησης αποκαλύπτουν ότι από τις 248 μελέτες που εντοπίστηκαν, οι 83 πρωτογενείς μελέτες συμπεριλήφθηκαν στην ανάλυση. Οι συγγραφείς παρέχουν πληροφορίες σχετικά με τις τάσεις δημοσίευσης, επισημαίνοντας τους κύριους συγγραφείς και τα φόρουμ για τον αυτοματοποιημένο έλεγχο κινητών εφαρμογών. Συζητούν επίσης τον αυξανόμενο αριθμό δημοσιεύσεων τα τελευταία χρόνια, με το έλεγχο βάση μοντέλου να επικρατεί ως η πιο συχνά εφαρμοσμένη προσέγγιση.

Τα αποτελέσματα της ανασκόπησης παρουσιάζουν διαφορετικές προσεγγίσεις για αυτοματοποιημένο έλεγχο και συζητούν για διάφορα εργαλεία και πλαίσια λογισμικών που είναι διαθέσιμα για τον έλεγχο των εφαρμογών κινητής τηλεφωνίας, αν και η διαθεσιμότητα και η χρηστικότητα ποικίλλουν.

Επίσης, γίνεται αναφορά στις προκλήσεις που αντιμετωπίζει ο αυτοματοποιημένος έλεγχος, όπως η ποικιλία των πλαισίων συμβάντων, ο κατακερματισμός στο λογισμικό και τον ηλεκτρονικό υπολογιστή και οι περιορισμοί πόρων. Επισημαίνεται η ανάγκη για περαιτέρω έρευνα όσον αφορά τη δημιουργία αποτελεσματικών και αποδοτικών μοντέλων για αυτοματοποιημένους ελέγχους και την αξιοποίηση του υπολογιστικού νέφους με σκοπό τον έλεγχο.

Ακόμη, συζητείται η εμπειρική επικύρωση των προσεγγίσεων αυτοματοποιημένων ελέγχων, με τις μελέτες περιπτώσεων να αποτελούν την πιο συχνά χρησιμοποιούμενη μέθοδο. Ωστόσο, οι συγγραφείς σημειώνουν διαφορές ως προς την αυστηρότητα και την ποιότητα μεταξύ αυτών των μελετών περίπτωσης.

Συμπερασματικά, η έρευνα παρέχει πολύτιμες πληροφορίες σχετικά με τις προκλήσεις και τις προσεγγίσεις στον αυτοματοποιημένο έλεγχο των κινητών εφαρμογών. Επισημαίνει την ανάγκη για περαιτέρω έρευνα, διαθεσιμότητα εργαλείων και εμπειρική επικύρωση για τη βελτίωση της αποτελεσματικότητας και της αποδοτικότητας των αυτοματοποιημένων δοκιμών. Τα ευρήματα αυτής της μελέτης μπορούν να καθοδηγήσουν τους επαγγελματίες στη λήψη τεκμηριωμένων αποφάσεων σχετικά με τις προσεγγίσεις για ελέγχους πάνω σε κινητές εφαρμογές. [27]

Κεφάλαιο 3ο: Εφαρμογή Αυτοματοποιημένου Ελέγχου στη Βιομηχανία

3.1 Οργανόγραμμα Εμπλεκόμενης Βιομηχανίας

Η Adminkit είναι μια startup εταιρεία και εδρεύει στο Σάντεφιρντ της Νορβηγίας¹. Αυτή την χρονική περίοδο, η ομάδα της Adminkit αποτελείται από είκοσι άτομα. Η ομάδα της ανάπτυξης λογισμικού αποτελείται από δεκατρία άτομα:

- *ένας Διευθύνων Σύμβουλος Τεχνολογίας (Chief Technology Officer)*: Ο Διευθύνων Σύμβουλος Τεχνολογίας αποτελεί ένα από τα σημαντικότερα πρόσωπα σε οποιαδήποτε εταιρεία τεχνολογίας, καθώς λειτουργεί ως επικεφαλής τεχνολόγος. Έχει σημαντικό ρόλο στον καθορισμό των επιλογών που κάνει μια εταιρεία. Αυτές είναι επιλογές για μελλοντικά προϊόντα, υπηρεσίες, ροές εργασίας, τεχνολογίες και καινοτομίες.
- *ένας DevOps*: Ένας μηχανικός DevOps είναι υπεύθυνος για ένα σύνολο πρακτικών που συνδυάζουν την ανάπτυξη λογισμικού (Dev) και τις λειτουργίες πληροφορικής (Ops) με σκοπό τη συντόμηση του κύκλου ζωής του συστήματος και την παροχή συνεχούς παράδοσης με υψηλή ποιότητα λογισμικού.
- *επτά Προγραμματιστές Λογισμικού (Software Developers)*: Οι προγραμματιστές λογισμικού δημιουργούν λογισμικό που ανταποκρίνεται στις ανάγκες των χρηστών, χρησιμοποιώντας διαγράμματα και μοντέλα, γράφοντας κώδικα και εξασφαλίζοντας τη συνολική λειτουργικότητα.
- *τρεις Μηχανικούς Διασφάλισης Ποιότητας (QA Engineers)*: Οι μηχανικοί διασφάλισης ποιότητας είναι αυτοί που εργάζονται για να διασφαλίσουν τη σωστή λειτουργία του λογισμικού. Προσπαθούν να αποτρέψουν ελαττωματικές συμπεριφορές του λογισμικού πριν από το να φτάσουν στον πελάτη. Το αποτέλεσμα αυτό επιτυγχάνεται εκτελώντας μια ποικιλία χειροκίνητων και αυτοματοποιημένων ελέγχων ενώ το προϊόν βρίσκεται ακόμη σε εξέλιξη.
- *ένας UX/UI Σχεδιαστής (Designer)*: Ο Σχεδιαστής UX επικεντρώνεται στην εμπειρία του χρήστη με το προϊόν, ενώ ο Σχεδιαστής UI ασχολείται περισσότερο με την εμφάνιση και την αίσθηση του προϊόντος. Αυτές οι δύο ικανότητες μπορούν και εφαρμόζονται από το ίδιο άτομο.

Οι υπόλοιπες θέσεις καλύπτονται από:

- *έναν Διευθύνων Σύμβουλος (Chief Executive Officer)*: Ο Διευθύνων Σύμβουλος, βρίσκεται στην ανώτατη βαθμίδα ιεράρχησης στη διοικητική δομή της εταιρείας. Συνήθως λαμβάνει κρίσιμες στρατηγικές αποφάσεις για την επιχείρηση μαζί με το διοικητικό συμβούλιο της εταιρείας.
- *έναν Επικεφαλής Μάρκετινγκ (Chief Marketing Officer)*: Ο Επικεφαλής Μάρκετινγκ είναι το εταιρικό στέλεχος που είναι υπεύθυνο για τις δραστηριότητες μάρκετινγκ ενός οργανισμού.
- *έναν Ελεγκτή (Controller)*: Ο Ελεγκτής είναι ένα άτομο που έχει την ευθύνη για όλες τις δραστηριότητες που σχετίζονται με τη λογιστική, συμπεριλαμβανομένης της λογιστικής υψηλού επιπέδου, της διοικητικής λογιστικής και των χρηματοοικονομικών δραστηριοτήτων, σε μια εταιρεία.

¹ <https://adminkit.no/>

- έναν Υπεύθυνο Προστασίας Δεδομένων (*Data Protection Officer*): Ο Υπεύθυνος Προστασίας Δεδομένων είναι ένας ρόλος εντός μιας εταιρείας, η ευθύνη του οποίου είναι να διασφαλίζει ότι η εταιρεία ή ο οργανισμός προστατεύει σωστά τα προσωπικά δεδομένα των ατόμων σύμφωνα με την ισχύουσα νομοθεσία.
- έναν Διευθυντή Μάρκετινγκ (*Marketing Director*): Ο Διευθυντής Μάρκετινγκ είναι ένας επαγγελματίας που είναι υπεύθυνος για τη διαχείριση όλων των πτυχών που σχετίζονται με την παραγωγή και την υλοποίηση οποιασδήποτε διαφημιστικής καμπάνιας.
- δύο Διαχειριστές Επιτυχίας Πελατών (*Customer Success Managers*): Ο Διαχειριστής Επιτυχίας Πελατών υποστηρίζει τους πελάτες της εταιρείας κατά τη μετάβασή τους από υποψήφιους πωλητές σε ενεργούς χρήστες των προϊόντων της εταιρείας.

Οι δεκαπέντε από αυτούς έχουν φυσική παρουσία στα γραφεία της Adminkit, ενώ οι υπόλοιποι πέντε εργάζονται εξ αποστάσεως ως εξωτερικοί συνεργάτες σε συνεργασία με την ελληνική εταιρία Dataverse Ltd². Το προσωπικό της εταιρείας αποτελείται κυρίως από Νορβηγούς και Έλληνες, αλλά και από άλλες εθνικότητες, οπότε η επικοινωνία μεταξύ των εργαζομένων γίνεται μιλώντας την αγγλική γλώσσα.

3.2 Διαδικασία Ανάπτυξης Λογισμικού

Η ομάδα της Adminkit ακολουθεί την ευέλικτη (*agile*) μεθοδολογία για την ανάπτυξη του λογισμικού. Η ευέλικτη μεθοδολογία είναι μια προσέγγιση διαχείρισης έργου (*project management*) που βασίζεται στην ευελιξία και τη συνεργασία. Δίνει έμφαση στην επαναλαμβανόμενη παράδοση μικρότερων εργασιών (*tasks*) ενός έργου (*project*) και ενθαρρύνει τη συνεχή βελτίωση μέσω τακτικής ανατροφοδότησης και προσαρμογής. Η ευέλικτη μεθοδολογία εκτιμά τα άτομα και τις αλληλεπιδράσεις μεταξύ τους έναντι των διαδικασιών και των εργαλείων παραγωγής, το λογισμικό εργασίας έναντι των εμπειριστατωμένων τεκμηρίων, τη συνεργασία με τον πελάτη έναντι της διαπραγματεύσεως συμβολαίων και την ανταπόκριση σε αλλαγές κατά την διάρκεια εφαρμογής ενός σχεδίου. Χρησιμοποιείται συνήθως στην ανάπτυξη λογισμικού και σε έργα πληροφορικής, όπως συμβαίνει και στην εταιρεία όπου εργαζόμαστε, την Adminkit.

Το εργαλείο που χρησιμοποιούμε για να εφαρμόσουμε την ευέλικτη μεθοδολογία είναι το Jira. Το Jira είναι ένα εργαλείο λογισμικού που αναπτύχθηκε από την Atlassian και χρησιμοποιείται για την παρακολούθηση έργων και ζητημάτων. Έχει σχεδιαστεί για να βοηθά τις ομάδες να σχεδιάζουν, να παρακολουθούν και να κυκλοφορούν νέες εκδόσεις λογισμικού και χρησιμοποιείται συνήθως στην ανάπτυξη λογισμικού, στη διαχείριση υπηρεσιών πληροφορικής και στη διαχείριση προϊόντων. Το Jira παρέχει λειτουργίες όπως ευέλικτη διαχείριση έργου, παρακολούθηση σφαλμάτων, προσαρμοσμένους πίνακες εργαλείων και αναφοράς.

Όλα τα μέλη της ομάδας, ανεξαρτήτως ειδικότητας, έχουν από έναν προσωπικό λογαριασμό ώστε να μπορούν να παρακολουθούν και να συμβάλλουν στην πρόοδο της εργασίας. Κάθε χρήστης μπορεί να δημιουργήσει μία καινούργια εργασία, δίνοντας έναν τίτλο, μια περιγραφή και το είδος της κατηγορίας της εργασίας. Υπάρχουν πέντε κατηγορίες εργασιών:

- *Task*: Το Task αντιπροσωπεύει μια δουλειά που πρέπει να γίνει.
- *Sub-task*: Το Sub-task είναι στην ουσία μια δευτερεύουσα εργασία, ένα κομμάτι εργασίας που απαιτείται για την ολοκλήρωση ενός Task.

² <https://www.dataverse.gr/>

- *Bug*: Το Bug αντιπροσωπεύει ένα σφάλμα, είναι ένα πρόβλημα που επηρεάζει ή αποτρέπει τις λειτουργίες ενός προϊόντος.
- *Story*: Μια ιστορία χρήστη είναι η μικρότερη ενότητα εργασίας που πρέπει να γίνει.
- *Epic*: Μια μεγάλη ιστορία χρήστη που πρέπει να αναλυθεί. Το Epic ομαδοποιεί σφάλματα, ιστορίες και εργασίες για να δείξει την πρόοδο μιας μεγαλύτερης πρωτοβουλίας. Στην ευέλικτη ανάπτυξη, τα Epics αντιπροσωπεύουν συνήθως ένα σημαντικό παραδοτέο, όπως ένα νέο χαρακτηριστικό ή την εμπειρία χρήσης του λογισμικού που αναπτύσσει η ομάδα.

Όλες αυτές οι εργασίες στοιβάζονται σε μία λίστα, με το όνομα 'Issues'. Μια φορά την εβδομάδα, υπάρχει μια προγραμματισμένη συνάντηση μεταξύ του διοικητικών μελών της εταιρείας, ώστε να αποφασίσουν ποιες από τις υπάρχουσες εργασίες θα τεθούν σε προτεραιότητα και θα μπουν στο Kanban Board.

Ένας πίνακας Kanban στο Jira είναι ένα οπτικό εργαλείο διαχείρισης έργου που παρέχει έναν τρόπο οπτικοποίησης και διαχείρισης της εργασίας σε εξέλιξη. Βασίζεται στη μέθοδο Kanban, μια μέθοδο Lean παραγωγής για τη διαχείριση και τη βελτίωση της εργασίας σε ανθρώπινα συστήματα. Στο Jira, ένας πίνακας Kanban είναι ένας τύπος πίνακα έργου που επιτρέπει στις ομάδες να βλέπουν, να διαχειρίζονται και να ιεραρχούν εργασίες σε πραγματικό χρόνο. Οι εργασίες αναπαρίστανται ως κάρτες στον πίνακα και η πρόοδος απεικονίζεται καθώς οι κάρτες μετακινούνται από τη μια στήλη στην άλλη. Οι στήλες μπορούν να αντιπροσωπεύουν διαφορετικά στάδια εργασίας, όπως "To do", "In progress" και "Done". Ο πίνακας Kanban στο Jira είναι εξαιρετικά προσαρμόσιμος και μπορεί να προσαρμοστεί για να ταιριάζει στις συγκεκριμένες ανάγκες μιας ομάδας. Χρησιμοποιείται για τη βελτίωση της ροής της εργασίας, τη μείωση του χρόνου παράδοσης και την αύξηση της διαφάνειας και της συνεργασίας μεταξύ των μελών της ομάδας.

Αξίζει να αναφέρουμε πως στις κάρτες αναγράφεται πάντα αυτός που δημιούργησε την εργασία καθώς και το όνομα αυτού που ασχολείται με την εργασία τη συγκεκριμένη χρονική στιγμή. Στην ομάδα της AdminKit, αυτή την χρονική περίοδο, ο πίνακας Kanban αποτελείται από οκτώ στήλες:

1. *Selected for Development*: Εδώ τοποθετούνται όλες οι εργασίες που έχουν θέσει σε προτεραιότητα τα διοικητικά μέλη της εταιρείας για να υλοποιηθούν την τρέχων εβδομάδα.
2. *In Progress*: Σε αυτή την στήλη συναντάμε τις εργασίες οι οποίες είναι υπό ανάπτυξη. Οι προγραμματιστές, έπειτα από μεταξύ τους συνεννόηση επιλέγουν τις εργασίες με τις οποίες θα ασχοληθούν, μεταφέρουν την κάρτα από την προηγούμενη στήλη στην παρούσα και ξεκινάνε την υλοποίηση.
3. *Ready for Code Review*: Σε αυτή την στήλη μεταφέρονται οι κάρτες οι οποίες έχουν ολοκληρωθεί και περιμένουν για έναν έλεγχο του κώδικα από έναν από τους προγραμματιστές που δεν συνέβαλε στην υλοποίηση της εργασίας.
4. *Code Reviewing*: Εδώ, ένας προγραμματιστής που βρίσκει λίγο χρόνο από την κύρια εργασία του, αξιολογεί την εργασία ενός άλλου προγραμματιστή ώστε να γίνει ένας επιπλέον έλεγχος. Σε περίπτωση που παρατηρηθεί κάποιο λάθος, τότε υποδεικνύει ο προγραμματιστής που κάνει τον έλεγχο το λάθος στον προγραμματιστή που το υλοποίησε ώστε να το διορθώσει, οπότε η κάρτα γυρίζει στην προηγούμενη στήλη. Όμως, υπάρχει και η περίπτωση ο προγραμματιστής που κάνει την αξιολόγηση να διορθώσει ο ίδιος τυχόν λάθη ή να κάνει διορθώσεις στον κώδικα ώστε να βελτιστοποιήσει τον αλγόριθμο.

5. *Ready for QA on DEV*: Εφόσον ολοκληρωθεί ο έλεγχος του κώδικα, ο προγραμματιστής τοποθετεί την κάρτα σε αυτή τη στήλη, περιμένοντας έναν από την ομάδα διασφάλισης ποιότητας.
6. *Testing on DEV*: Σε αυτή την στήλη μεταφέρεται μια κάρτα από έναν μηχανικό ελέγχου ώστε να ξεκινήσει ο έλεγχος για το αν έχει υλοποιηθεί σωστά η εργασία. Ο έλεγχος γίνεται στο ‘development’ περιβάλλον, καθώς σε αυτή την βάση δεδομένων και περιβάλλον γίνονται οι νέες υλοποιήσεις. Σε περίπτωση που παρατηρηθεί κάποιο λάθος, ο μηχανικός ελέγχου ενημερώνει τον προγραμματιστή που είναι υπεύθυνος ώστε να το διορθώσει και γυρίζει την κάρτα πίσω στην στήλη ‘In Progress’.
7. *Ready for Staging*: Εφόσον τελειώσει ο μηχανικός ελέγχου τον έλεγχο, τοποθετεί την κάρτα στη συγκεκριμένη στήλη, η οποία δηλώνει τον έλεγχο στο ‘staging’ περιβάλλον. Εδώ ο μηχανικός ελέγχου θα πρέπει να μεταφέρει τις αλλαγές του κώδικα στην δεύτερη δοκιμαστική βάση δεδομένων και περιβάλλον. Έπειτα, να ελέγξει ξανά ότι όλα δουλεύουν όπως ήταν αναμενόμενο.
8. *Done*: Εφόσον ολοκληρωθεί και ο έλεγχος στο ‘staging’ περιβάλλον, ο μηχανικός μεταφέρει την κάρτα στην τελευταία στήλη. Εδώ συγκεντρώνονται όλες οι εργασίες που έχουν ολοκληρωθεί και είναι έτοιμες για την επερχόμενη μέρα που θα γίνει η μεταφορά στο παραγωγικό περιβάλλον.

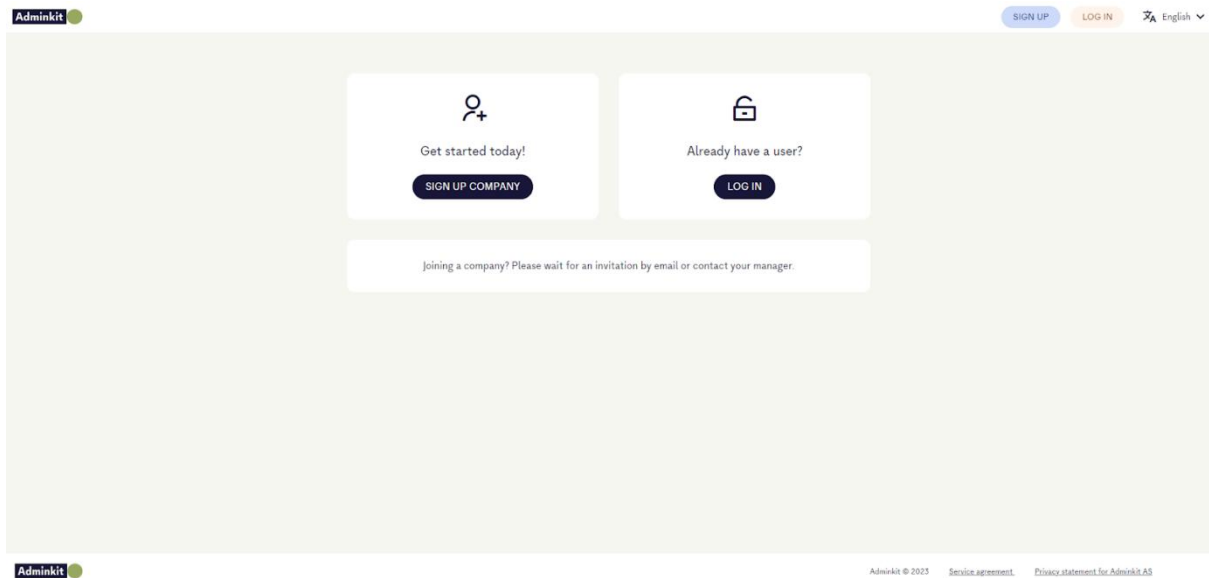
Εφόσον ολοκληρωθεί η ανάπτυξη (deployment) στην παραγωγική βάση, συνήθως γίνεται μια φορά την εβδομάδα, η στήλη ‘Done’ αδειάζει και όλες οι εργασίες εξακολουθούν να παραμένουν στο ιστορικό των εργασιών.

3.3 Περιγραφή Εμπλεκόμενου Λογισμικού

Το προϊόν της Adminkit είναι μία εφαρμογή ιστού με στόχο την καλύτερη διαχείριση και λειτουργία των μικρομεσαίων επιχειρήσεων. Προσφέρει υπηρεσίες και εργαλεία τα οποία βοηθούν στην οργάνωση μια επιχείρησης, όσον αφορά την αποθήκευση αρχείων, την συγχρότιση των υπαλλήλων, την πρόσληψη υπαλλήλων με χρήση ψηφιακών υπογραφών και πολλές άλλες που θα αναπτύξουμε στην πορεία. Η εφαρμογή υποστηρίζεται στα νορβηγικά, καθώς όλοι οι πελάτες που χρησιμοποιούν την εφαρμογή δραστηριοποιούνται στη Νορβηγία. Όμως, είναι ήδη διαθέσιμη και η αγγλική μετάφραση διότι υπάρχουν βλέψεις για την επέκταση του προϊόντος στην παγκόσμια αγορά. Για να χρησιμοποιήσει κανείς την εφαρμογή, το μόνο που χρειάζεται είναι να έχει πρόσβαση στο διαδίκτυο και να επισκεφτεί τον ιστότοπο της εφαρμογής³.

Συγκεκριμένα, στο Σχήμα 3.3α παρουσιάζεται ο προθάλαμος της εφαρμογής. Ως νέος χρήστης, ο οποίος δεν διαθέτει ενεργό λογαριασμό, θα πρέπει να πατήσει το κουμπί ‘SIGN UP COMPANY’ ώστε να εγγραφεί στην εταιρεία ως νέος χρήστης.

³ <https://app.adminkit.com/en/lobby>



Σχήμα 3.3α: Προθάλαμος εφαρμογής Adminkit.

Έπειτα θα μεταφερθείτε στην σελίδα εγγραφής του χρήστη που απεικονίζεται στο Σχήμα 3.3β και θα συμπληρώσετε τα στοιχεία σας για να δημιουργήσετε τον λογαριασμό σας. Θα χρειαστεί το ονοματεπώνυμο σας (First name και Last name), ο αριθμός του τηλεφώνου σας (Phone number), το E-mail σας και ένας κωδικός (Password).

Σχήμα 3.3β: Εγγραφή χρήστη.

Πατώντας το κουμπί του 'Register', έχετε πλέον δημιουργήσει τον προσωπικό σας λογαριασμό και μεταφερόσαστε στο επόμενο βήμα που είναι να κάνετε την εγγραφή της εταιρείας σας στην εφαρμογή της Adminkit, όπως φαίνεται στο Σχήμα 3.3γ. Απαιτείται να συμπληρώσετε τα πεδία με το όνομα της εταιρίας σας (Company name), την οδό της εταιρίας (Street Address), τον ταχυδρομικό κώδικα (Postal code), την πόλη και την χώρα όπου εδρεύει (City και Country), των αριθμό των υπαλλήλων που απασχολεί η εταιρεία (Number of employees), να συμφωνείτε με τη δήλωση απορρήτου για την Adminkit AS και τη συμφωνία παροχής υπηρεσιών (I accept the Privacy statement for Adminkit AS & Service agreement) και τέλος να πατήσετε το κουμπί 'REGISTER'.

Home / Signup

User Company

About the company

Please provide some details about your company.

Company name: Thesis ✓ Organization number: [input]

Street address: International Hellenic University ✓ Street address 2: Sindos Campus

Postal code: 57400 ✓ City: Thessaloniki ✓

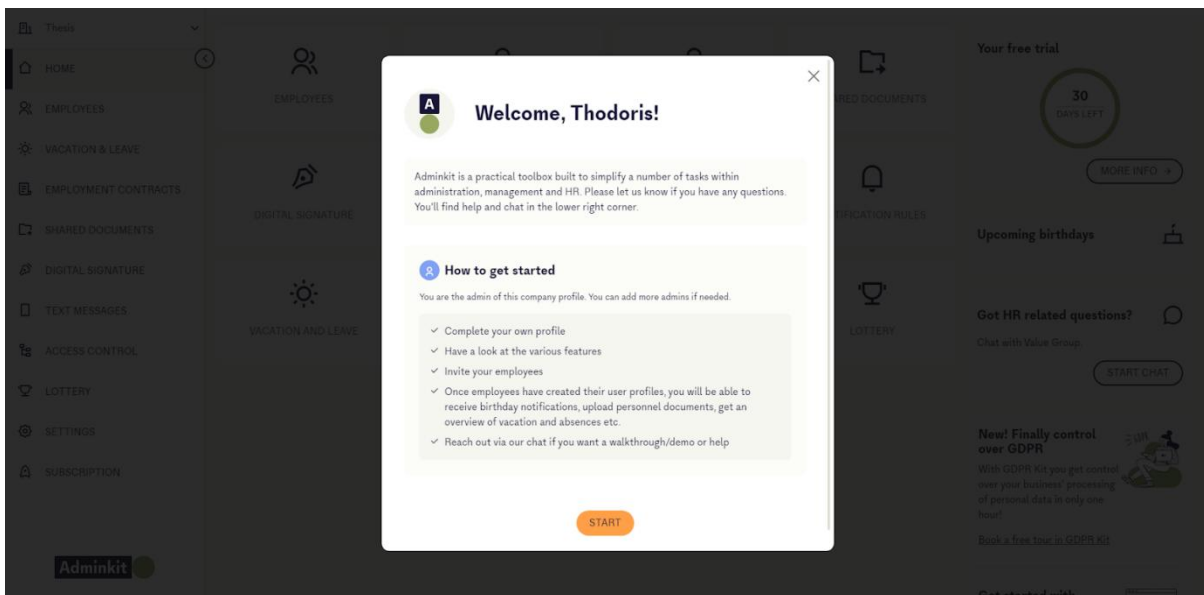
Country: GR - Greece × Number of employees: 100 ✓

I accept the Privacy statement for Adminkit AS
 I accept the Service agreement

← BACK REGISTER ✓

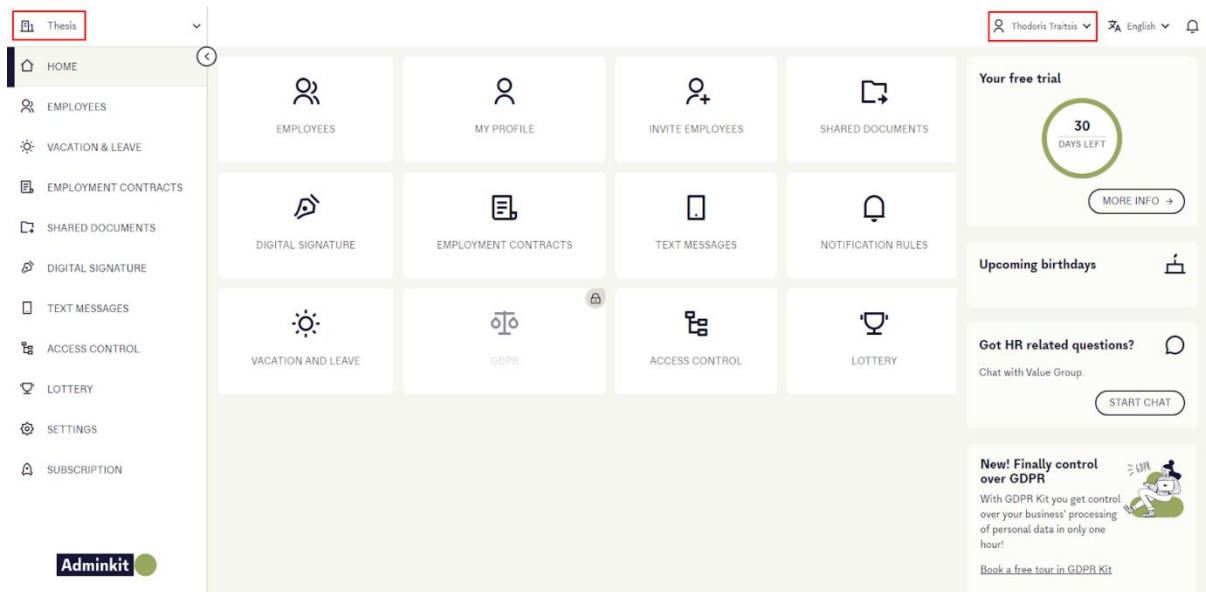
Σχήμα 3.3γ: Εγγραφή εταιρείας.

Στη συνέχεια θα λάβετε το μήνυμα καλωσορίσματος, όπως φαίνεται στο Σχήμα 3.3δ και θα πατήσετε το κουμπί ‘START’ για να κλείσει το μήνυμα.



Σχήμα 3.3δ: Μήνυμα καλωσορίσματος.

Τέλος, έχετε ολοκληρώσει τα βήματα εγγραφής και έχετε πρόσβαση στην εφαρμογή. Στο Σχήμα 3.3ε απεικονίζεται η αρχική σελίδα της εφαρμογής. Στο πάνω αριστερό μέρος διακρίνεται το όνομα της εταιρείας σας, ενώ στα δεξιά φαίνεται το ονοματεπώνυμο σας. Στο κέντρο διακρίνουμε όλες τις λειτουργίες (features) της εφαρμογής, όπως και στην μπάρα πλοήγησης στο αριστερό τμήμα.



Σχήμα 3.3ε: Αρχική σελίδα Adminkit.

3.3.1 Λειτουργίες Εμπλεκόμενου Λογισμικού

Η εφαρμογή είναι συνδρομητική και προσφέρει μηνιαία πακέτα τα οποία δίνουν πρόσβαση σε ορισμένες λειτουργίες της εφαρμογής. Πριν αναλύσουμε τις λειτουργίες, αξίζει να αναφέρουμε όλες τις κατηγορίες χρηστών της εφαρμογής, έτσι ώστε να κατανοήσουμε καλύτερα τις δυνατότητες και τον σκοπό της κάθε λειτουργίας της εφαρμογής. Πρώτα από όλα έχουμε τον διαχειριστή (administrator), ο οποίος αντιστοιχεί στον ιδιοκτήτη της εταιρείας, είναι ο διαχειριστής της εφαρμογής, έχει τα περισσότερα δικαιώματα και έχει την μέγιστη δυνατή πρόσβαση σε όλες τις λειτουργίες, ανάλογα με την συνδρομή που καταβάλλει. Έπειτα συναντάμε τον υπάλληλο (employee), ο οποίος αντιστοιχεί σε έναν υπάλληλο της εταιρείας, είναι η πιο απλή κατηγορία χρήστη, έχει ελάχιστα δικαιώματα και περιορισμένη πρόσβαση στις διαθέσιμες λειτουργίες. Ακόμη, υπάρχει ο διευθυντής (manager), που αντιστοιχεί σε έναν επικεφαλής μιας εταιρείας ο οποίος είναι υπεύθυνος για έναν αριθμό υπαλλήλων οι οποίοι αναφέρονται σε αυτόν για κάθε τους αίτημα, γι' αυτό και έχει μερικά δικαιώματα παραπάνω από τον υπάλληλο αλλά και μεγαλύτερη προσβασιμότητα στις λειτουργίες, συγκριτικά με τον υπάλληλο. Τέλος, υπάρχει μια ειδική κατηγορία, αυτή του απενεργοποιημένου χρήστη (deactivated user), η οποία αντιστοιχεί σε έναν πρώην υπάλληλο ο οποίος έχει διαγραφεί, πλέον, από την εταιρεία και δεν έχει πια πρόσβαση στις λειτουργίες της εφαρμογής, παρά μόνο την είσοδο στο προφίλ του.

3.3.1.1 Employees

Σε αυτή την σελίδα, όπως φαίνεται στο Σχήμα 3.3.1.1, διακρίνουμε τη λίστα των εργαζομένων της εταιρείας. Είναι εμφανές το ονοματεπώνυμο τους, ο ρόλος τους στην εταιρεία, το email τους, ο αριθμός του τηλεφώνου τους, η ημερομηνία γενεθλίων τους και η ένδειξη ότι έχουν ενεργούς λογαριασμούς. Ο διαχειριστής, πατώντας πάνω στην στήλη του, μεταφέρετε στη σελίδα του προφίλ του όπου μπορεί να επεξεργαστεί και να προσθέσει περισσότερες πληροφορίες για αυτόν. Ο διαχειριστής έχει την δυνατότητα να επισκέπτεται και να επεξεργάζεται όλα τα προφίλ των υπαλλήλων. Αντιθέτως, ένας διευθυντής έχει πρόσβαση και μπορεί να επεξεργαστεί το προφίλ του και αυτά των υφιστάμενων του. Από την άλλη, ο απλός χρήστης έχει πρόσβαση και δικαιώματα επεξεργασίας μόνο στο προσωπικό του προφίλ.

Επίσης, στη συγκεκριμένη σελίδα διακρίνουμε τα κουμπιά 'INVITE EMPLOYEES' και 'ADD EMPLOYEE' τα οποία μεταφέρουν τον διαχειριστή στα αντίστοιχες παράθυρα όπου πραγματοποιούνται οι εγγραφές και οι προσκλήσεις νέων χρηστών, με βάση το email τους.

First name	Last name	Job title	E-mail address	Phone	Birthday	User access
Thodoris	Traitsis	Boss	t.traitsis@dataverse.gr	+30 6956789356	Jan 17th	✓
Ash	Ketchum	Pokémon Trainer	ash@adminikit-test.com	+47 32435252455	Apr 1st	✓
Babis	Babinos	Bomber	babis@adminikit-test.com	+47 8943893893	Jan 15th	✓
Cristiano	Ronaldo	Centre Forward	ronaldo@adminikit-test.com	+47 74784892092	Feb 5th	✓
Damian	Lillard	Guard	lillard@adminikit-test.com	+47 34536464643	Jul 15th	✓
Deuteros	Noumero2	Back End	deuteros@adminikit-test.com	+60 3535353524	Nov 2nd	✓
Erling	Haaland	Centre Forward	haaland@adminikit-test.com	+47 7857823892	Jul 21st	✓

Σχήμα 3.3.1.1: Σελίδα Employee.

3.3.1.2 Vacation & Leave

Η λειτουργία Vacation & Leave (Σχήμα 3.3.1.2) αποτελεί ένα απλό ημερολόγιο διακοπών και απουσιών που διευκολύνει τόσο τους εργοδότες όσο και τους εργαζόμενους να παρακολουθούν τις διακοπές και τις απουσίες τους, έχοντας πρόσβαση σε ξεχωριστό ημερολόγιο για τον κάθε ένα εργαζόμενο και ένα κοινό ημερολόγιο για ολόκληρη την εταιρεία. Όλοι οι εργαζόμενοι μπορούν να υποβάλλουν αίτηση για διακοπές, οι οποίες εγκρίνονται ή απορρίπτονται από τους υπεύθυνους ή τον διευθυντή. Με αυτόν τον τρόπο, δίνεται στον διευθυντή μια πλήρη εικόνα τόσο των ιστορικών όσο και των μελλοντικών απουσιών για όλους τους εργαζόμενους της εταιρείας. Η εργασία από το σπίτι (τηλεργασία) είναι ακόμη ένα είδος απουσίας που μπορεί ο κάθε εργαζόμενος να επιλέξει, δίνοντας σε όλους τους συναδέλφους μια γενική εικόνα για το πόσοι άνθρωποι έρχονται στο γραφείο εκείνη την ημέρα.

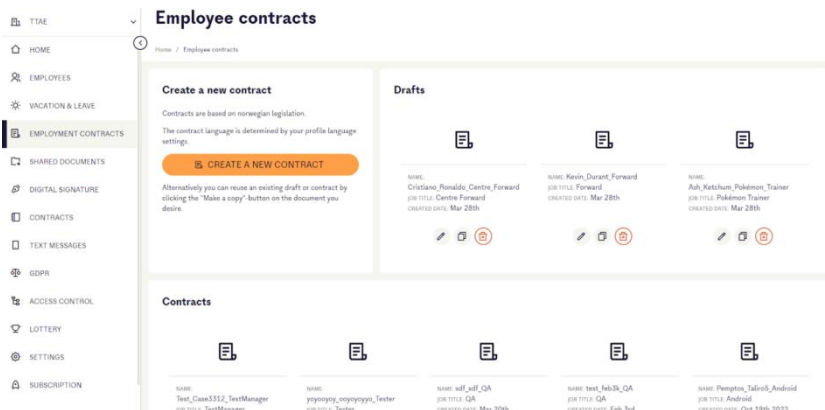
Ακόμη, σε περίπτωση που κάποιος οφείλει ημέρες άδειας, έχει την δυνατότητα να τις μεταφέρει στο επόμενο έτος. Ο διαχειριστής εγκρίνει ή απορρίπτει τα αιτήματα μεταφοράς με τον ίδιο τρόπο όπως τα αιτήματα διακοπών και απουσιών.

Name	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S								
Thodoris Traitsis Boss	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	—
Ash Ketchum Pokémon Trainer	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	—
Babis Babinos Bomber	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	—
Cristiano Ronaldo Centre Forward	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	—
Damian Lillard Guard	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	—
Deuteros Noumero2 Back End	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	—

Σχήμα 3.3.1.2: Σελίδα Vacation & Leave.

3.3.1.3 Employment Contracts

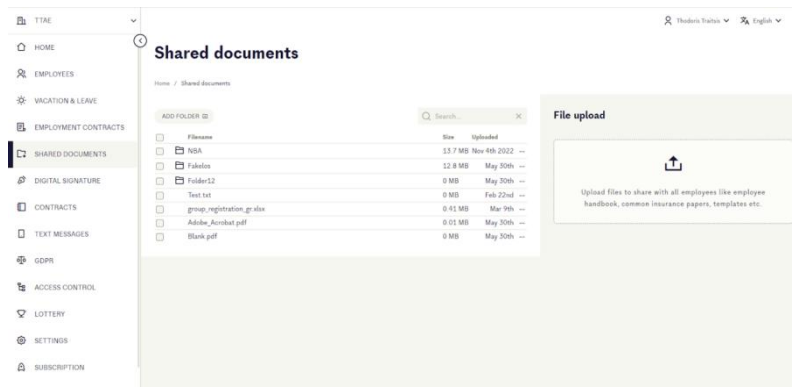
Η λειτουργία που παρουσιάζεται στο Σχήμα 3.3.1.3 είναι προσβάσιμο μόνο από τον διαχειριστή και έχει σχεδιαστεί, αποκλειστικά και μόνο, ακολουθώντας πιστά τη νομοθεσία της Νορβηγίας. Το Employee contract βοηθάει τον εργοδότη στο να δημιουργήσει εύκολα, ακολουθώντας τέσσερα απλά βήματα, μια σύμβαση εργασίας προς τους νέους υπαλλήλους της επιχείρησής του. Αρχικά, συμπληρώνει τα στοιχεία του εργαζομένου, συμπληρώνοντας λεπτομέρειες σχετικά με τον εργαζόμενο, το είδος της θέσης, τον μισθό αλλά και ποιός πρέπει να υπογράψει. Στο επόμενο βήμα, συμπληρώνονται περισσότερες λεπτομέρειες για τους όρους της σύμβασης, όπως τα ωράρια εργασίας, οι άδειες, κ.τ.λ. Έπειτα, διακρίνουμε τη επισκόπηση της σύμβασης, την οποία μπορούμε να προσαρμόσουμε τις λεπτομέρεια στις προτιμήσεις μας, αλλάζοντας τη σειρά, επεξεργάζοντας ή προσθέτοντας στοιχεία. Τέλος, μένει να επιλέξουμε σε ποιον πρέπει να στείλουμε την σύμβαση προς ψηφιακή υπογραφή. Η ολοκληρωμένη υπογεγραμμένη σύμβαση αποστέλλεται μέσω ηλεκτρονικού ταχυδρομείου σε όλους όσους υπογράφουν και ένα αντίγραφο της θα αποθηκεύεται αυτόματα στα προφίλ των δικαιούχων μέσα στην εφαρμογή.



Σχήμα 3.3.1.3: Σελίδα Employment Contracts.

3.3.1.4 Shared Documents

Η συγκεκριμένη λειτουργία που φαίνεται στο Σχήμα 3.3.1.4, είναι ένα κοινό μέρος για την αποθήκευση και τη διάθεση των κοινών εγγράφων και αρχείων μέσα στη εταιρεία. Μόνο ο διαχειριστής μπορεί να ανεβάζει, να μετονομάζει ή να διαγράφει αρχεία. Οι υπάλληλοι μπορούν μόνο να κατεβάζουν ή να βλέπουν μια προεπισκόπηση των εγγράφων. Κατά αυτόν τον τρόπο, μειώνεται ο περιττός χρόνος αναζήτησης και αποστολής πληροφοριών, αποθηκεύοντας και οργανώνοντας τις νεότερες εκδόσεις διαφόρων εγγράφων.



Σχήμα 3.3.1.4: Σελίδα Shared Documents.

3.3.1.5 Digital Signature

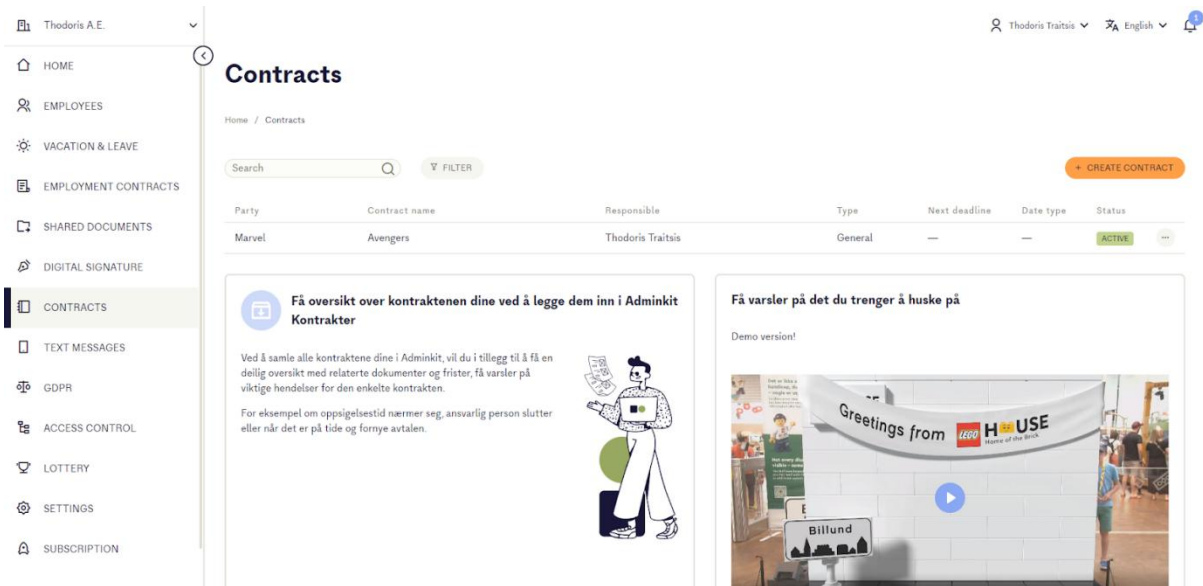
Το Digital signature απεικονίζεται στο Σχήμα 3.3.1.5 και αποτελεί άλλη μια λειτουργία που είναι προσβάσιμη μόνο από τον διαχειριστή και συμβάλλει στην αποθήκευση και αποστολή υπογεγραμμένων συμβάσεων και εγγράφων. Για τα προσωπικά έγγραφα, ο διευθυντής μπορεί να επιλέξει ένα από τα ανεβασμένα έγγραφα του προφίλ του και να το στείλει προς υπογραφή από έναν εργαζόμενο. Τα έγγραφα επιστρέφονται αυτόματα στο προφίλ του εργαζομένου όταν έχουν υπογραφούν και από τις δύο πλευρές. Η δεύτερη επιλογή είναι η υπογραφή άλλων εγγράφων. Αρχικά, ο διαχειριστής ανεβάζει ένα έγγραφο (τύπου .pdf) και προσθέστε τις διευθύνσεις ηλεκτρονικού ταχυδρομείου όλων όσων πρόκειται να υπογράψουν. Όταν όλοι υπογράψουν ηλεκτρονικά, το υπογεγραμμένο έγγραφο επιστρέφεται στο προφίλ των εργαζομένων. Επίσης, ο διαχειριστής έχει την δυνατότητα να παρακολουθεί σε τι στάδιο βρίσκονται τα απεσταλμένα έγγραφα, δηλαδή εάν έχουν σταλθεί, εάν έχουν υπογραφεί μόνο από έναν ή από όλους τους παραλήπτες.

Document name	Created date	Due date	Signing status	Requested by	Recipients
+ Test.pdf	Jan 25th	Feb 1st	SIGNED	Thodoris Traitsis	2 recipients
x Pemptos_Taliro5_Android.pdf	Oct 19th 2022	Oct 26th 2022	SIGNED	Thodoris Traitsis	2 recipients
Recipients					
Name	Signing status	Completed date	Reject reason		
pemptos@adminikit-test.com	Signed	Oct 19th 2022	—		
Thodoris Traitsis	Signed	Oct 19th 2022	—		
+ Tertartos_Number4_Front_End.pdf	Oct 19th 2022	Oct 26th 2022	SIGNED	Thodoris Traitsis	2 recipients
+ Tritos_Tria3_QA.pdf	Oct 19th 2022	Oct 26th 2022	SIGNED	Thodoris Traitsis	2 recipients

Σχήμα 3.3.1.5: Σελίδα Digital signature.

3.3.1.6 Contracts

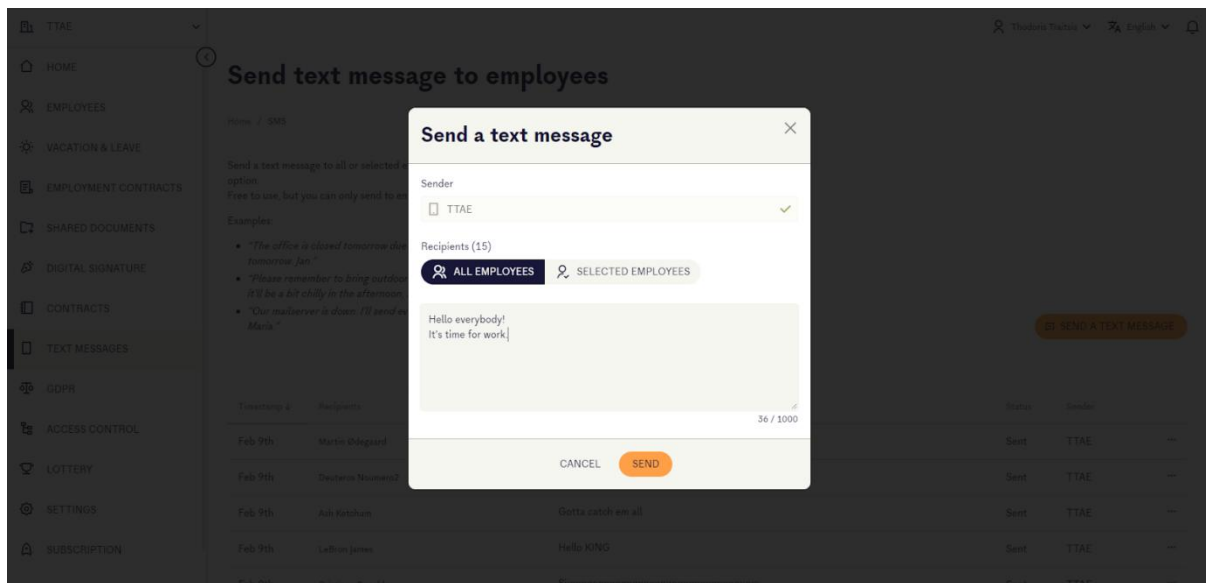
Η παρακάτω λειτουργία αποτελεί τη πιο πρόσφατη κυκλοφορία της Adminikit. Πολλοί οργανισμοί έχουν δημιουργήσει χάος αποθηκεύοντας συμβάσεις και συμφωνίες στο Sharepoint, στο Google Disk ή τοπικά σε έναν φάκελο. Το Contracts (Σχήμα 3.3.1.6) έρχεται να δώσει λύση σε αυτό το πρόβλημα, ως ένα κεντρικό μέρος για να αποθηκεύουν οι εταιρίες όλες τις σημαντικές συμβάσεις και συμφωνίες τους, καθορίζοντας παράλληλα το ποιος είναι υπεύθυνος για τις ενημερώσεις των εγγράφων. Τα αρμόδια μέλη της εκάστοτε εταιρείας, λαμβάνουν ειδοποιήσεις όταν πλησιάζει μια σημαντική ημερομηνία, λόγω χάρη τις ημερομηνίες αυτόματης ανανέωσης ή ακύρωσης συμβάσεων και συμφωνιών. Επίσης, εάν ένας εργαζόμενος αποχωρήσει, θα πρέπει να σταλούν ειδοποιήσεις σχετικά με τις συμφωνίες για τις οποίες ήταν υπεύθυνος ο εργαζόμενος. Με ένα κλικ, είναι εφικτό να μεταφερθούν τα έγγραφα, που ήταν υπεύθυνος ο πρώην εργαζόμενος, σε κάποιον άλλον. Αυτός είναι ο λόγος για τον οποίο είναι τόσο χρήσιμο και πρακτικό για μια εταιρεία να έχει τόσο την επισκόπηση προσωπικού όσο και την επισκόπηση συμφωνιών στο ίδιο σύστημα.



Σχήμα 3.3.1.6: Σελίδα Contracts

3.3.1.7 Text Messages

Στη λειτουργία των Text Messages έχει πρόσβαση αποκλειστικά και μόνο ο διαχειριστής. Η εφαρμογή της Adminkit αποθηκεύει τους αριθμούς των τηλεφώνων των υπαλλήλων της. Όπως φαίνεται στο Σχήμα 3.3.1.7, ο διαχειριστής σε περίπτωση που χρειάζεται να κάνει μια σημαντική ανακοίνωση, μπορεί να στείλει ένα γραπτό τηλεφωνικό μήνυμα σε επιλεγμένους ή και σε όλους τους υπαλλήλους. Με τον τρόπο αυτό, οι υπάλληλοι ενημερώνονται άμεσα, λαμβάνοντας μια ειδοποίηση στο κινητό τους τηλέφωνο. Επίσης, η εφαρμογή κρατάει ιστορικό με τα απεσταλμένα μηνύματα και δίνει την δυνατότητα της επαναποστολής των SMS.



Σχήμα 3.3.1.7: Σελίδα Text Messages.

3.3.1.8 GDPR

Η ελληνική μετάφραση του GDPR ορίζεται ως ο Γενικός Κανονισμός για την Προστασία των Δεδομένων (General Data Protection Regulation—GDPR). Στοιχεί να προσφέρει στους πολίτες της

ΕΕ (Ευρωπαϊκή Ένωση) μια ενιαία και εναρμονισμένη προσέγγιση όσον αφορά την προστασία της ιδιωτικής ζωής στην Ευρωπαϊκή Ένωση. Επιδιώκει να ενισχύσει τα δικαιώματα των πολιτών για την προστασία των δεδομένων τους, όπως ορίζεται στο άρθρο 8 του Χάρτη Θεμελιωδών Δικαιωμάτων της ΕΕ. Θέτει τα θεμελιώδη δικαιώματα των ατόμων στην ψηφιακή εποχή, τις υποχρεώσεις όσων επεξεργάζονται δεδομένα, τις μεθόδους διασφάλισης της συμμόρφωσης και τις κυρώσεις για όσους παραβιάζουν τους κανόνες. Μετά από σχεδόν τέσσερα χρόνια συζητήσεων, το GDPR εγκρίθηκε από το κοινοβούλιο της ΕΕ στις 14 Απριλίου 2016, ενώ η ημερομηνία υποχρεωτικής εφαρμογής του GDPR καθορίστηκε στις 25 Μαΐου 2018.

Ο GDPR επηρεάζει όλους επιχειρηματίες που απασχολούν υπαλλήλους ή πελάτες. Οι εργαζόμενοι, οι πελάτες και οι προμηθευτές περιμένουν από αυτούς να επεξεργάζονται τα προσωπικά τους δεδομένα με ασφάλεια και νομιμότητα. Ο GDPR και η προστασία των δεδομένων αποτελούν φυσικό μέρος των συνεντεύξεων των εργαζομένων, των προσλήψεων, των ενημερωτικών δελτίων, της μισθοδοσίας, των εργασιών του διοικητικού συμβουλίου και της περιήγησης στα μέσα κοινωνικής δικτύωσης. [28]

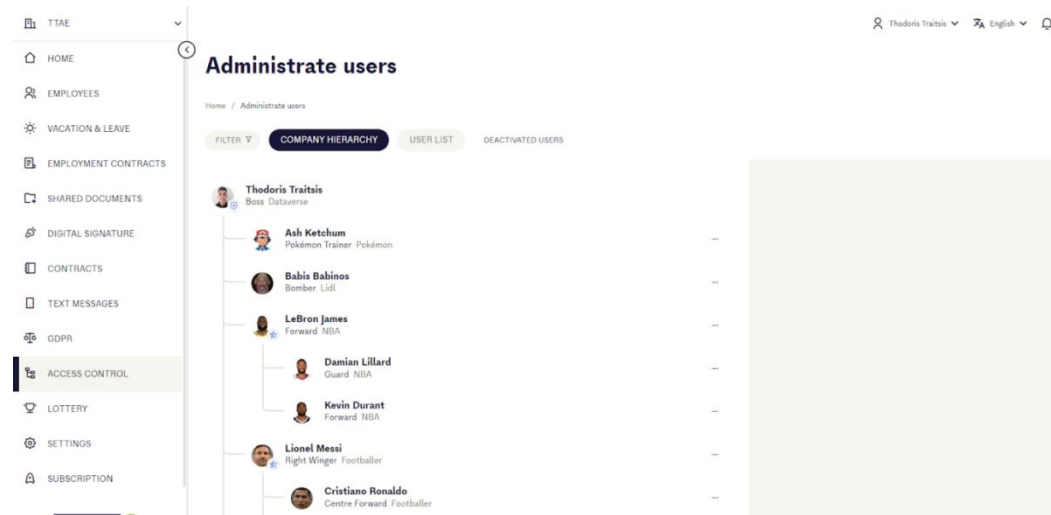
Με τη λειτουργία GDPR, που απεικονίζεται στο Σχήμα 3.3.1.8, είναι εφικτό για τις μικρές και μεσαίες επιχειρήσεις να συμμορφώνονται οι ίδιες με τους κανόνες του GDPR, αποφεύγοντας πρόστιμα. Μέσω της Adminkit μπορεί κανείς να αναλάβει τον έλεγχο του απορρήτου των δεδομένων της επιχείρησης, συμπληρώσει εύκολα το πρωτόκολλο με τις διάφορες δραστηριότητες που πραγματοποιούνται, όπως η διαχείριση της μισθοδοσίας ή η αποστολή ενημερωτικών δελτίων. Με βάση αυτές τις πληροφορίες, θα δημιουργηθεί αυτόματα μια συμφωνία επεξεργασίας δεδομένων, μια αξιολόγηση κινδύνου και μια πολιτική απορρήτου. Έτσι, αποκτώνται όλα όσα χρειάζονται για την εκπλήρωση των απαιτήσεων του GDPR, χωρίς τη χρήση ακριβών δικηγόρων.

Σχήμα 3.3.1.8: Σελίδα GDPR.

3.3.1.9 Access Control

Στο Σχήμα 3.3.1.9 συναντάμε τη λειτουργία του Access Control, στην οποία έχει πρόσβαση αποκλειστικά και μόνο ο διαχειριστής. Σε αυτό το σημείο ο διαχειριστής καθορίζει την ιεραρχία μεταξύ των υπαλλήλων. Στην κορυφή της ιεραρχίας βρίσκεται ο ίδιος, καθώς όλοι οι υπάλληλοι αναφέρονται σε αυτόν. Στο αμέσως επόμενο επίπεδο συναντάμε τους διευθυντές. Ο διαχειριστής, για να ορίσει έναν διευθυντή, πρέπει να ορίσει τον υπάλληλος 'ένα', να αναφέρεται στον υπάλληλο 'δύο'.

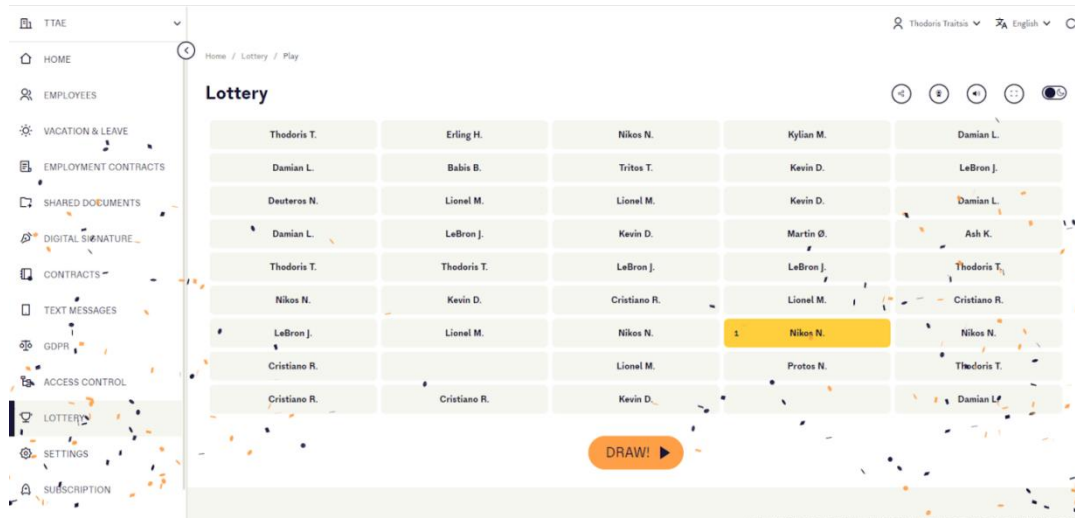
Έτσι, ο υπάλληλος ‘δύο’ αναβαθμίζεται σε υπεύθυνος έχοντας τα αντίστοιχα δικαιώματα προς τον υπάλληλο ‘ένα’, τα οποία έχουμε αναλύσει νωρίτερα. Με τον έλεγχο πρόσβασης, απελευθερώνεται πολύτιμος χρόνος από τις υποχρεώσεις του διαχειριστή, διότι ανατίθενται καθήκοντα και ευθύνες σε άλλα μέλη του οργανισμού. Τέλος, έχουμε τους απλούς υπαλλήλους, που αναφέρονται σε κάποιον υπεύθυνο ή απευθείας στον διαχειριστή.



Σχήμα 3.3.1.9: Σελίδα Access Control.

3.3.1.10 Lottery

Το Lottery αποτελεί έναν πίνακα κληρώσεων, με σκοπό να κάνει μια εργάσιμη ημέρα λίγο πιο διασκεδαστική. Η λειτουργία κλήρωσης είναι ένα πρακτικό και διασκεδαστικό εργαλείο που καθιστά εξαιρετικά εύκολη την προσθήκη συμμετεχόντων και τη διεξαγωγή τυχαίας κλήρωσης. Κάθε κλήρωση μπορεί εύκολα να κοινοποιηθεί, ώστε όλοι να μπορούν να συμμετέχουν ταυτόχρονα, όπου κι αν βρίσκονται. Όπως απεικονίζεται στο Σχήμα 3.3.1.10, πατώντας το κουμπί ‘DRAW’ ξεκινάει η κλήρωση από την οποία θα επιλεγεί ένας τυχαίος νικητής, την επόμενη φορά που θα πατηθεί ξανά το κουμπί θα κληρωθεί ένας διαφορετικός νικητής από αυτόν στον οποίο σταματά η προηγούμενη κλήρωση. Το Lottery αποτελεί ένα ιδανικό εργαλείο για να φέρει πιο κοντά τα μέλη της εταιρείας καθώς μπορούν να καθιερωθούν εβδομαδιαίες κληρώσεις, δίνοντας έπαθλα στους νικητές.



Σχήμα 3.3.1.10: Σελίδα Lottery.

3.4 Περιγραφή Εργαλείου

Το εργαλείο που χρησιμοποιεί η AdminKit για την δημιουργία και εκτέλεση αυτοματοποιημένων ελέγχων είναι το Learwork. Το Learwork είναι μια ισχυρή πλατφόρμα αυτοματοποιημένου ελέγχου που επιτρέπει στους χρήστες να δημιουργούν και να εκτελούν αυτοματοποιημένους ελέγχους χωρίς καμία γνώση προγραμματισμού. Παρέχει ένα οπτικό περιβάλλον εργασίας και ένα ευρύ φάσμα ενσωματωμένων ενεργειών για την αλληλεπίδραση με εφαρμογές, προγράμματα περιήγησης ιστού και συστήματα, καθιστώντας το προσιτό τόσο σε τεχνικούς όσο και σε μη τεχνικούς χρήστες.

Το Learwork χρησιμοποιεί το πρωτόκολλο Selenium WebDriver για την αλληλεπίδραση με τα προγράμματα περιήγησης ιστού κατά τη διάρκεια των αυτοματοποιημένων ελέγχων. Το Selenium WebDriver είναι ένα δημοφιλές framework ανοικτού κώδικα για την αυτοματοποίηση προγραμμάτων περιήγησης στο διαδίκτυο και το Learwork αξιοποιεί τις δυνατότητές του για την παροχή λειτουργιών αυτοματοποίησης στο διαδίκτυο.

Με το Learwork, οι χρήστες μπορούν να αυτοματοποιήσουν σενάρια χρήσης πάνω σε μια εφαρμογή ιστού χρησιμοποιώντας ενέργειες που βασίζονται στο Selenium. Αυτές οι ενέργειες επιτρέπουν στους χρήστες να εκτελούν εργασίες όπως η συμπλήρωση φορμών (form), το κλικ (click) σε στοιχεία (elements), η επαλήθευση κειμένου (text), η πλοήγηση (navigation) μεταξύ σελίδων και ο χειρισμός αλληλεπιδράσεων JavaScript. Το Learwork αφαιρεί τις πολυπλοκότητες του Selenium και παρέχει μια οπτική διεπαφή για τη δημιουργία και τη διαμόρφωση αυτών των ενεργειών χωρίς να απαιτείται από τους χρήστες να γράφουν κώδικα. Αυτό σημαίνει ότι οποιοσδήποτε οργανισμός μπορεί να δημιουργήσει και να εκτελέσει περιπτώσεις αυτοματοποίησης χωρίς να χρειάζεται να μάθει ένα νέο σύνολο δεξιοτήτων ή να βασιστεί σε εξωτερικούς συμβούλους. Με την ενσωμάτωση του Selenium WebDriver, το Learwork επιτρέπει στους χρήστες να αυτοματοποιούν αποτελεσματικά και αποδοτικά τα tests, εξασφαλίζοντας την ανθεκτικότητα και την αξιοπιστία των εφαρμογών ιστού.

3.4.1 Αρχιτεκτονική

Το Learwork ακολουθεί μια σπονδυλωτή αρχιτεκτονική, η οποία αποτελείται από ένα Studio (Designer), έναν Controller (Server) και έναν Agent (Runner), τα οποία είναι τα βασικά συστατικά που συνεργάζονται για να επιτρέψουν την αυτοματοποίηση δοκιμών.

3.4.1.1 Studio (Designer)

Το Studio (Στούντιο), γνωστό και ως Designer (Σχεδιαστής), είναι μια εφαρμογή για την επιφάνεια εργασίας που παρέχεται από το Learwork. Χρησιμοποιεί ως διεπαφή χρήστη όπου σχεδιάζονται, διαμορφώνονται και συντηρούνται οι έλεγχοι. Το Studio παρέχει μια οπτική διεπαφή αρπάζοντας και σύροντας (drag-and-drop) που επιτρέπει στους χρήστες να δημιουργούν ροές ελέγχων συνδυάζοντας διάφορες ενέργειες, συνθήκες, βρόχους και άλλα στοιχεία ροής ελέγχου. Οι χρήστες μπορούν επίσης να ορίσουν παραμέτρους εισόδου, μεταβλητές και πηγές δεδομένων μέσα στο Studio.

Το Studio παρέχει μια ολοκληρωμένη βιβλιοθήκη προκατασκευασμένων ενεργειών (blocks) που καλύπτουν ένα ευρύ φάσμα εργασιών αυτοματοποίησης. Αυτές οι ενέργειες περιλαμβάνουν αλληλεπιδράσεις με εφαρμογές, βάσεις δεδομένων, προγράμματα περιήγησης στο διαδίκτυο, API και άλλα. Επιπλέον, οι χρήστες μπορούν να δημιουργήσουν προσαρμοσμένες ενέργειες για τον χειρισμό συγκεκριμένων λειτουργιών ή την ενσωμάτωση με εξωτερικά συστήματα.

Μόλις σχεδιαστούν τα σενάρια ελέγχου στο Studio, μπορούν να δημοσιευτούν στον Controller για εκτέλεση. Το Studio χρησιμοποιείται κυρίως από τους σχεδιαστές των ελέγχων ή τους μηχανικούς αυτοματισμού για τη δημιουργία και τη συντήρηση των αυτοματοποιημένων ελέγχων.

3.4.1.2 Controller (Server)

Ο Controller (Ελεγκτής) λειτουργεί ως κεντρικός κόμβος διαχείρισης για το Leapwork. Είναι ένα στοιχείο που βασίζεται σε διακομιστή και συντονίζει την εκτέλεση των ελέγχων, διαχειρίζεται τα περιουσιακά στοιχεία των ελέγχων και παρέχει τη διεπαφή χρήστη για τον έλεγχο και την παρακολούθηση της διαδικασίας του ελέγχου.

Ο ελεγκτής λαμβάνει τις περιπτώσεις ελέγχου που δημοσιεύονται από το Studio και τις διανέμει στους Agents για εκτέλεση. Διαχειρίζεται τις ουρές των ελέγχων, προγραμματίζει τις εκτελέσεις τους και αναθέτει την εκτέλεση τους στους διαθέσιμους Agents με βάση τις ρυθμίσεις και τις δυνατότητες εκτέλεσης. Ο Controller χειρίζεται επίσης την επικοινωνία μεταξύ του Studio, των Agents και άλλων ολοκληρωμένων συστημάτων.

Επιπλέον, ο Controller παρέχει δυνατότητες αναφοράς και ανάλυσης. Δημιουργεί λεπτομερείς αναφορές μετά την εκτέλεση των ελέγχων, κρατάει αρχεία καταγραφής, στιγμιότυπα οθόνης και βιντεοσκοπήσεις των εκτελεσμένων ελέγχων. Αυτές οι αναφορές βοηθούν στην ανάλυση των αποτελεσμάτων των ελέγχων, στον εντοπισμό προβλημάτων και στην παρακολούθηση της συνολικής κάλυψης των ελέγχων. Η πρόσβαση στον Controller γίνεται συνήθως μέσω μιας διεπαφής που βασίζεται στο διαδίκτυο.

3.4.1.3 Agent (Runner)

Ο Agent (Πράκτορας) είναι ένα ελαφρύ στοιχείο λογισμικού που εγκαθίσταται στις μηχανές όπου εκτελούνται οι έλεγχοι. Οι πράκτορες λειτουργούν ως μηχανές εκτέλεσης για την αυτοματοποίηση των ελέγχων. Δημιουργούν την επικοινωνία μεταξύ του Controller και των υπό δοκιμή συστημάτων.

Όταν ένας έλεγχος προγραμματίζεται για εκτέλεση, ο Controller την αναθέτει σε έναν διαθέσιμο Agent. Ο Agent ανακτά την περίπτωση ελέγχου, εκτελεί τις ενέργειες που ορίζονται στη ροή ελέγχου και αλληλεπιδρά με το σύστημα. Αυτή η αλληλεπίδραση μπορεί να περιλαμβάνει αλληλεπιδράσεις με το UI, κλήσεις API, λειτουργίες βάσης δεδομένων, χειρισμούς αρχείων και πολλά άλλα.

Ο Agent καταγράφει και αναφέρει την κατάσταση εκτέλεσης, τα αρχεία καταγραφής και άλλες σχετικές πληροφορίες στον Controller. Μπορεί επίσης να λαμβάνει στιγμιότυπα οθόνης και να καταγράφει βίντεο κατά τη διάρκεια της εκτέλεσης του ελέγχου, παρέχοντας οπτικά στοιχεία για την αποσφαλμάτωση και την ανάλυση.

Οι Agents μπορούν να εγκατασταθούν σε διάφορα περιβάλλοντα, συμπεριλαμβανομένων φυσικών μηχανών, εικονικών μηχανών ή περιπτώσεις που βασίζονται στο νέφος (cloud). Μπορούν να εγκατασταθούν πολλαπλοί πράκτορες για να επιτραπεί η παράλληλη εκτέλεση ελέγχων, ενισχύοντας την επεκτασιμότητα και μειώνοντας το συνολικό χρόνο εκτέλεσης των ελέγχων.

Συνοπτικά, το Studio (Σχεδιαστής) χρησιμοποιείται για τη δημιουργία και τη συντήρηση περιπτώσεων ελέγχου, ο Ελεγκτής (Διακομιστής) διαχειρίζεται την εκτέλεση των ελέγχων και παρέχει τη διεπαφή χρήστη για τη διαχείριση των ελέγχων και ο Πράκτορας (Δρομέας) εκτελεί την πραγματική εκτέλεση των ελέγχων στις καθορισμένες μηχανές. Αυτά τα στοιχεία συνεργάζονται για να διευκολύνουν την αποτελεσματική και επεκτάσιμη αυτοματοποίηση των ελέγχων με το Leapwork.[29]

3.4.2 Building Blocks

Στο Learwork, ένα μπλοκ συναρμολόγησης (building block) είναι μια προ-δημιουργημένη ενέργεια που αντιπροσωπεύει μια συγκεκριμένη λειτουργία ή αλληλεπίδραση που μπορεί να εκτελεστεί κατά τη διάρκεια του αυτοματοποιημένου ελέγχου. Τα μπλοκ συναρμολόγησης χρησιμεύουν ως τα θεμελιώδη συστατικά που οι χρήστες μπορούν να τραβήξουν και να σύρουν στον καμβά σχεδίασης στο Studio (Designer) για να δημιουργήσουν αυτοματοποιημένες περιπτώσεις ελέγχου χωρίς να γράψουν κώδικα. Κάθε μπλοκ συναρμολόγησης ενθυλακώνει μια συγκεκριμένη λειτουργικότητα, διευκολύνοντας τους χρήστες να συνθέτουν ροές ελέγχων (test flows).

Το Learwork παρέχει μια μεγάλη βιβλιοθήκη με μπλοκ συναρμολόγησης, τα οποία κατηγοριοποιούνται σε διάφορους τύπους με βάση τις λειτουργίες που εκτελούν. Μερικές από τις πιο συχνά εφαρμοσμένες κατηγορίες μπλοκ συναρμολόγησης περιλαμβάνουν:

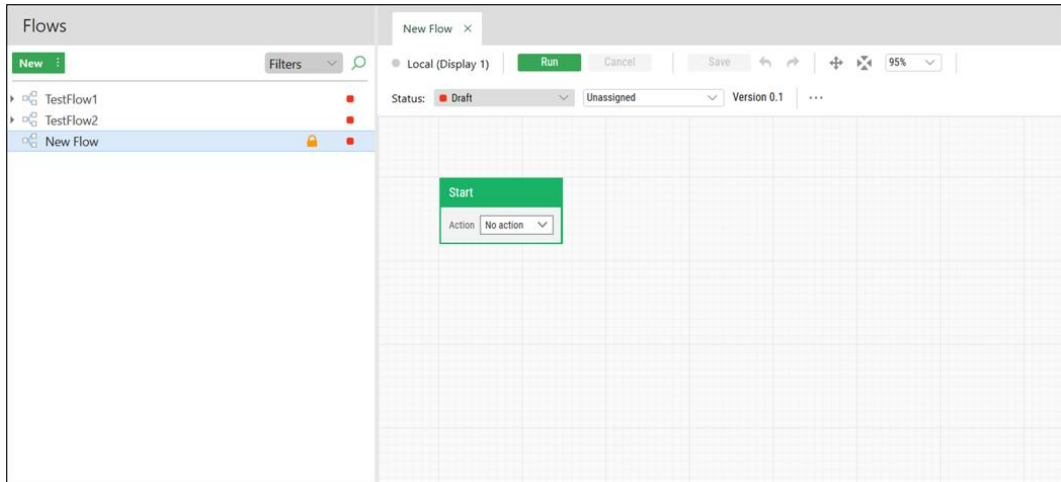
- *Application Building Blocks (Μπλοκ Συναρμολόγησης για Εφαρμογές):*
 - Open Application (Ανοιχτή Εφαρμογή): Χρησιμοποιείται για την εκκίνηση μιας εφαρμογής, όπως μιας desktop εφαρμογής ή μιας εφαρμογής για κινητά, για έλεγχο.
 - Close Application (Κλείσιμο εφαρμογής): Κλείνει την υπό έλεγχο εφαρμογή.
 - Wait for Application (Αναμονή για εφαρμογή): Διακόπτει την εκτέλεση του ελέγχου μέχρι να ανοίξει μια συγκεκριμένη εφαρμογή.
- *Web Building Blocks (Μπλοκ Συναρμολόγησης Ιστού):*
 - Open Browser (Άνοιγμα του προγράμματος περιήγησης): Εκκινεί ένα πρόγραμμα περιήγησης ιστού για έλεγχο των εφαρμογών διαδικτύου.
 - Navigate (Πλοήγηση): Πλοηγείται σε μια συγκεκριμένη ηλεκτρονική διεύθυνση (Uniform Resource Locator—URL) ή εκτελεί ενέργειες πλοήγησης στο πρόγραμμα περιήγησης (επιστροφή, προώθηση, ανανέωση).
 - Click: Κλικ σε στοιχεία του διαδικτύου όπως κουμπιά, συνδέσμους κ.λπ.
 - Input (Είσοδος): Εισάγει κείμενο ή δεδομένα σε πεδία εισαγωγής.
 - Verify Text (Επαλήθευση κειμένου): Ελέγχει αν ένα συγκεκριμένο κείμενο υπάρχει στην ιστοσελίδα.
 - Scroll (Κύλιση): Πραγματοποιεί κύλιση της ιστοσελίδας για την προβολή συγκεκριμένων στοιχείων.
- *File System Building Blocks (Μπλοκ Συναρμολόγησης για Συστήματα Αρχείων):*
 - Read File (Διάβασμα αρχείου): Διαβάζει δεδομένα από ένα αρχείο.
 - Write File (Γράψιμο αρχείου): Γράφει δεδομένα σε ένα αρχείο.
 - Copy File (Αντιγραφή αρχείου): Αντιγράφει ένα αρχείο από μια θέση σε μια άλλη.
- *Database Building Blocks (Μπλοκ Συναρμολόγησης για Βάσεις Δεδομένων):*
 - Connect to Database (Σύνδεση στη βάση δεδομένων): Δημιουργεί μια σύνδεση με μια βάση δεδομένων για την εκτέλεση λειτουργιών που σχετίζονται με τη βάση δεδομένων.
 - Execute SQL Query (Εκτέλεση ερωτήματος SQL): Εκτελεί ερωτήματα SQL για την αλληλεπίδραση με τη βάση δεδομένων.

- *API Building Blocks (Μπλοκ Συναρμολόγησης για API):*
 - Send HTTP Request (Αποστολή αιτήματος HTTP): Αποστέλλει αιτήματα HTTP (GET, POST, PUT, DELETE) για αλληλεπίδραση με APIs.
 - Verify JSON/XML (Επαλήθευση JSON/XML): Επικυρώνει απαντήσεις από APIs σε μορφή JSON ή XML.
- *Control Flow Building Blocks (Μπλοκ Συναρμολόγησης για Διαχείριση Ροής):*
 - If/Else: Προσθέτει λογικές συνθήκες στη ροή ελέγχου.
 - Loop (Βρόχος): Επαναλαμβάνει ένα σύνολο ενεργειών για συγκεκριμένο αριθμό φορών ή βάσει συγκεκριμένων συνθηκών.
- *Assertion Building Blocks (Μπλοκ Συναρμολόγησης Ισχυρισμού):*
 - Check (Έλεγχος): Επαληθεύει τις συνθήκες και ελέγχει αν συγκεκριμένες τιμές πληρούν τα αναμενόμενα κριτήρια.
 - Assert (Βεβαίωση): Αποτυγχάνει η δοκιμή εάν δεν ικανοποιείται μια συγκεκριμένη συνθήκη.
- *Miscellaneous Building Blocks (Διάφορα Μπλοκ Συναρμολόγησης):*
 - Pause (Παύση): Διακόπτει προσωρινά την εκτέλεση της δοκιμής για καθορισμένη διάρκεια.
 - Comment (Σχόλιο): Επιτρέπει στους χρήστες να προσθέτουν σχόλια σε περιπτώσεις ελέγχου για σκοπούς τεκμηρίωσης.
- *Sub-flows (Υπο-ροές):* Στο Learwork, μια υπο-ροή είναι μια επαναχρησιμοποιήσιμη ακολουθία από μπλοκ συναρμολόγησης που αντιπροσωπεύει ένα συγκεκριμένο σύνολο βημάτων ελέγχου ή μια ροή εργασίας. Οι υπο-ροές επιτρέπουν στους χρήστες να ενθυλακώσουν μια σειρά ενεργειών μέσα σε ένα ενιαίο μπλοκ, καθιστώντας εύκολη την οργάνωση και διαχείριση σύνθετων περιπτώσεων ελέγχου. Με τις υπο-ροές, οι χρήστες μπορούν να δημιουργήσουν αρθρωτές και συντηρήσιμες λύσεις αυτοματοποιημένων ελέγχων.

Αυτά είναι μερικά μόνο παραδείγματα των μπλοκ συναρμολόγησης που είναι διαθέσιμα στο Learwork. Η πλατφόρμα ενημερώνει συνεχώς τη βιβλιοθήκη της, προσθέτοντας νέα στοιχεία και βελτιώνοντας τα υπάρχοντα, ώστε να καλύπτει ένα ευρύ φάσμα σεναρίων ελέγχου. Οι χρήστες μπορούν να αξιοποιήσουν αυτά τα μπλοκ συναρμολόγησης για να δημιουργήσουν με ευκολία ολοκληρωμένες και ισχυρές αυτοματοποιημένες περιπτώσεις ελέγχου. [30]

3.4.3 Ροή Αυτοματοποιημένου Ελέγχου

Οι ροές ελέγχου στο Learwork σχεδιάζονται με την ένωση οπτικών μπλοκ συναρμολόγησης που το καθένα αντιπροσωπεύει και εκτελεί μία ή περισσότερες λειτουργίες. Από προεπιλογή, μια νεοδημιουργηθείσα ροή ελέγχου περιέχει μόνο ένα μπλοκ συναρμολόγησης, το μπλοκ Έναρξη (Start block), που βρίσκεται στον καμβά σχεδίασης, στο Σχήμα 3.4.3α.



Σχήμα 3.4.3α: Καμβάς σχεδίας στο Leawork.

Σε μια ροή αυτοματοποιημένου ελέγχου, τα μπλοκ συναρμολόγησης ενώνονται με τη χρήση συνδετήρων (connectors). Υπάρχουν δύο είδη συνδετήρων, οι πράσινοι που χρησιμοποιούνται για τον καθορισμό της σειράς εκτέλεσης ή της κατεύθυνσης της ροής ελέγχου και οι μπλε που χρησιμοποιούνται για τη μεταφορά τιμών και ιδιοτήτων μεταξύ των μπλοκ συναρμολόγησης.

Για να προσθέσει κανείς ένα νέο μπλοκ συναρμολόγησης, απλά πρέπει να περάσει με το ποντίκι πάνω από το πράσινο συνδετήρα ενός υπάρχοντος μπλοκ συναρμολόγησης, π.χ. του μπλοκ Start (Σχήμα 3.4.3α), και στη συνέχεια χρησιμοποιεί το ποντίκι για να τραβήξετε το πράσινο συνδετήρα προς οποιαδήποτε κατεύθυνση. Όταν αφήσει το ποντίκι, εμφανίζεται μια βιβλιοθήκη με όλα τα μπλοκ συναρμολόγησης του Leawork. Από αυτή τη λίστα, μπορεί είτε να χρησιμοποιήσει το ποντίκι για να επιλέξει ένα μπλοκ συναρμολόγησης είτε απλά να πληκτρολογήσει το όνομα για να φιλτράρετε τη λίστα που απεικονίζεται στο Σχήμα 3.4.3β.



Σχήμα 3.4.3β: Λίστα με όλες τις κατηγορίες των μπλοκ συναρμολόγησης.

Μέρος του σχεδιασμού ροών αυτοματοποιημένων ελέγχων είναι η σύλληψη στοιχείων στις εφαρμογές υπό αυτοματοποίηση. Στο Learwork, αυτό σημαίνει τη λήψη κάποιων πληροφοριών σχετικά με ένα στοιχείο της εφαρμογής, όπως το αναγνωριστικό ενός πεδίου (Identifier—ID), το όνομα ενός κουμπιού, μια περιοχή που περιέχει ένα συγκεκριμένο κείμενο (text) κ.λπ. Οι πληροφορίες/ιδιότητες των στοιχείων ονομάζονται Εντοπιστές (Locators) και αποτελούν βασικό μέρος κάθε ροής αυτοματισμού. Περιέχουν πληροφορίες που χρησιμοποιούνται κατά την εκτέλεση της ροής ελέγχου για την εύρεση και τον χειρισμό του συγκεκριμένου στοιχείου. Σχεδόν οποιαδήποτε εφαρμογή/ιστοσελίδα μπορεί να αυτοματοποιηθεί με το Learwork, συμπεριλαμβανομένων εφαρμογών ιστού, κινητών ιστοσελίδων, εφαρμογών επιφάνειας εργασίας (desktop) κάθε είδους, κ.λπ. Ιεραρχικά, όλοι οι Εντοπιστές τοποθετούνται στη λίστα κάτω από τη γονική ροή στην οποία ανήκουν και χωρίζονται στις εξής κατηγορίες:

- *Στοιχεία Ιστού (Web Elements)*: Ένα στοιχείο ιστού περιέχει όλες τις πληροφορίες σχετικά με ένα στοιχείο ιστού σε ένα διαδικτυακό περιβάλλον, π.χ. τη δομή HTML ενός στοιχείου ιστού με ένα στιγμιότυπο οθόνης του στοιχείου ελέγχου.
- *Στοιχεία Επιφάνειας εργασίας (Desktop Elements)*: Ένα στοιχείο επιφάνειας εργασίας περιέχει όλες τις πληροφορίες σχετικά με ένα στοιχείο επιφάνειας εργασίας σε μια εφαρμογή επιφάνειας εργασίας. Για παράδειγμα, την ιεραρχία ενός επιτραπέζιου στοιχείου ελέγχου με το στιγμιότυπο οθόνης του στοιχείου ελέγχου, όπως ένας φάκελος (folder).
- *Εικόνες (Images)*: Για την αυτοματοποίηση των εικονικών εφαρμογών, το Learwork χρησιμοποιεί την Αναγνώριση Εικόνας και Κειμένου (Optical Character Recognition—OCR) για τον εντοπισμό στοιχείων, την ανάγνωση κειμένου, το πάτημα κουμπιών κ.λπ. Κατά τη δημιουργία ροών αυτοματοποίησης για εικονικές εφαρμογές, συλλαμβάνεται μια εικόνα ενός συγκεκριμένου στοιχείου ελέγχου ή στοιχείου και η συλλαμβανόμενη εικόνα χρησιμοποιείται στη συνέχεια από τη μηχανή OCR για να εκτελέσει την απαιτούμενη λειτουργία στην οθόνη κατά την εκτέλεση της ροής ελέγχου.

Ένα σημαντικό μέρος του σχεδιασμού ροών αυτοματοποιημένου ελέγχου είναι η επαλήθευση της ροής που εκτελείται. Πέρασε η περίπτωση ελέγχου; Ολοκλήρωσε η αυτοματοποιημένη διαδικασία όλες τις εργασίες; Συνήθως, ενσωματώνετε η επαλήθευση στο τελευταίο μπλοκ συναρμολόγησης της ροής ελέγχου, χρησιμοποιώντας το μπλοκ Πέρασμα (Pass) εάν η επαλήθευση είναι επιτυχής. Εάν μια εκτέλεση ροής ελέγχου τελειώσει χωρίς να πετύχει ένα μπλοκ συναρμολόγησης τύπου Pass, η εκτέλεση ροής θεωρείται εξ ορισμού αποτυχημένη. Αυτό μπορεί να συμβεί αν δεν επιτευχθεί ένα συγκεκριμένο μπλοκ ή κατάσταση, όπως στην περίπτωση μιας αποτυχημένης σύνδεσης. Το Learwork έρχεται με ένα συγκεκριμένο μπλοκ συναρμολόγησης Αποτυχίας (Fail), αλλά δεν χρειάζεται να το προσθέσει κανείς για να χειριστεί όλα τα σενάρια αποτυχίας, αν ένα μπλοκ Pass σηματοδοτεί το τέλος μιας επιτυχημένης εκτέλεσης, τότε όλα τα άλλα σενάρια εκτέλεσης θα καταγραφούν εξ ορισμού ως αποτυχημένα. [31]

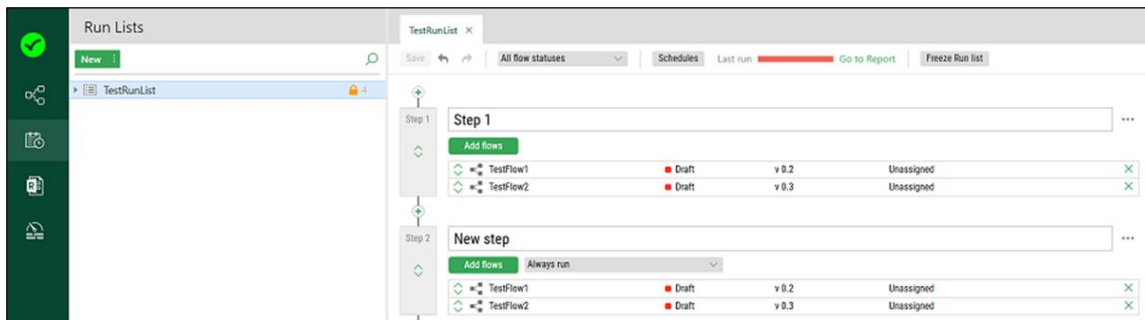
3.4.4 Εκτέλεση Ελέγχων και Αναφορά αποτελεσμάτων

Μια Λίστα Εκτέλεσης (Run List) είναι μια συλλογή από μία ή περισσότερες ροές αυτοματοποιημένων ελέγχων που εκτελούνται σε ένα ή περισσότερα χρονοδιαγράμματα. Η εκτέλεση μιας λίστας εκτέλεσης ελέγχεται από τον ελεγκτή Learwork Controller. Οι λίστες εκτέλεσης χρησιμοποιούνται σε πολλά σενάρια, μερικά από τα πιο συνηθισμένα είναι τα εξής:

- Εκτέλεση μεγαλύτερων συλλογών ροών αυτοματοποιημένων ελέγχων έναντι π.χ. περιβαλλόντων ανάπτυξης ή/και ελέγχου.
- Ομαδοποίηση ροών αυτοματοποιημένων ελέγχων σε βήματα και έλεγχος της σειράς εκτέλεσης των ελέγχων.

- Εκτέλεση μιας ή περισσότερων ροών αυτοματοποιημένων ελέγχων για την έγκαιρη ενεργοποίηση μιας επιχειρησιακής διαδικασίας.
- Εκτέλεση ροών αυτοματοποιημένων ελέγχων μέσω του δημόσιου API REST.
- Εκτέλεση των ροών αυτοματοποιημένων ελέγχων μέσω του Αγωγού Συνεχούς Ολοκλήρωσης και Παράδοσης (Continuous Integration and Delivery—CI-CD).
- Εκτέλεση των ίδιων ροών αυτοματισμού σε διαφορετικές διαμορφώσεις, περιβάλλοντα, μηχανές κ.λπ.

Αφού σχεδιάσουν οι ροές αυτοματοποιημένων ελέγχων, οι χρήστες μπορούν να δημιουργήσουν και να προγραμματίσουν μια λίστα εκτέλεσης ροών για αυτόματη εκτέλεση σε διαφορετικά λειτουργικά συστήματα και προγράμματα περιήγησης χρησιμοποιώντας τη διαμόρφωση του Χρονοπρογραμματιστή (Scheduler) που είναι διαθέσιμη κάτω από μια λίστα εκτέλεσης, όπως φαίνεται στο Σχήμα 3.4.4α. Μια λίστα εκτέλεσης αποτελείται πάντα από ένα ή περισσότερα βήματα. Κάθε βήμα σε μια λίστα εκτέλεσης μπορεί να περιέχει μία ή περισσότερες ροές προς εκτέλεση. Οι χρήστες μπορούν να προσθέσουν πολλαπλά βήματα για να ορίσουν και να ελέγξουν την εκτέλεση των ροών. Το πρώτο βήμα εκτελείται πάντα και τα υπόλοιπα βήματα μπορούν να τεθούν υπό όρους, δηλαδή η εκτέλεσή τους μπορεί να ελεγχθεί με βάση το αποτέλεσμα του προηγούμενου βήματος. [32]



Σχήμα 3.4.4α: Run List.

Η Αναφορά (Reporting) επιτρέπει στους χρήστες να παρακολουθούν και να αξιολογούν εύκολα την κατάσταση των ροών αυτοματοποιημένων ελέγχων τους σε έργα, προϊόντα, επιχειρηματικές μονάδες κ.λπ. Όλα τα αποτελέσματα εκτέλεσης, τόσο οι εκτελέσεις προεπισκόπησης, όσο και τα χρονοδιαγράμματα που ενεργοποιούνται χειροκίνητα και τα χρονοδιαγράμματα που ενεργοποιούνται μέσω API είναι προσβάσιμα στην ενότητα αναφοράς. Η αναφορά παρουσιάζει το αποτέλεσμα των εκτελέσεων των ροών αυτοματοποιημένων ελέγχων χρησιμοποιώντας δείκτες "φωτεινής σηματοδότησης". Κόκκινο για αποτυχημένες, κίτρινο για τελειωμένες/άγνωστες και πράσινο για επιτυχημένες εκτελέσεις ροής.

Ο αριθμός των αποτυχημένων, των πραγματοποιημένων και των επιτυχημένων εκτελέσεων συνοψίζεται αυτόματα για κάθε επίπεδο των ομαδοποιήσεων παρέχοντας μια γρήγορη επισκόπηση της κατάστασης των εκτελέσεων ροών αυτοματοποιημένων ελέγχων. Το μέρος των δεδομένων μιας αναφοράς μπορεί να εξαχθεί σε PDF, Excel ή HTML κάνοντας κλικ στην επιλογή 'Export' (Εξαγωγή).

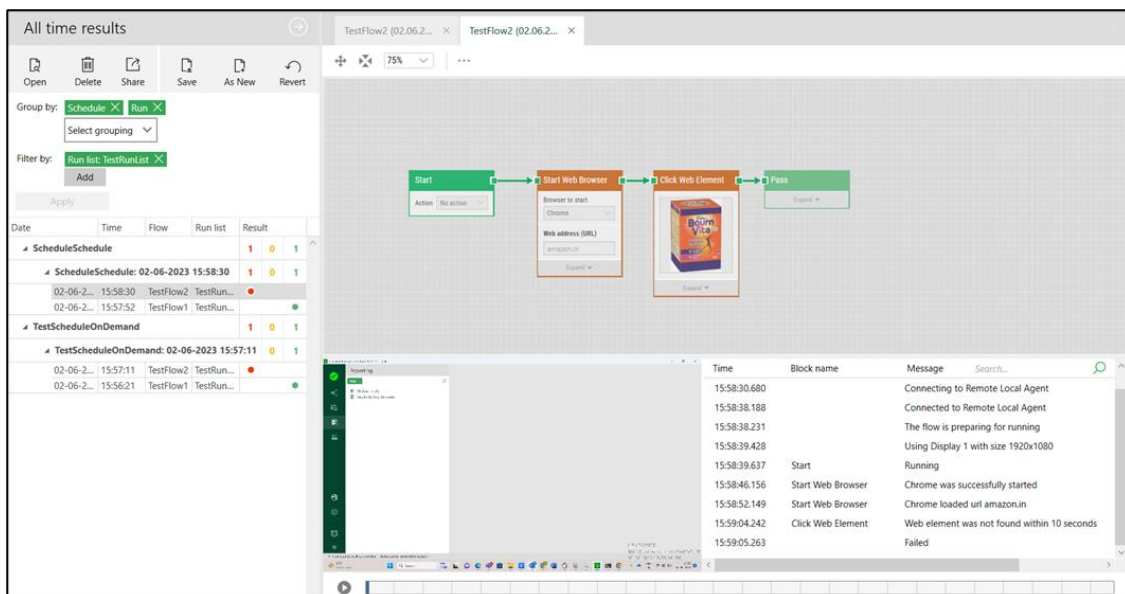
Κάνοντας κλικ σε μια καταχώρηση αναφοράς για να λάβετε περισσότερες πληροφορίες σχετικά με μια συγκεκριμένη εκτέλεση μιας ροής αυτοματοποιημένου ελέγχου. Αυτή θα ανοίξει στο δεξί παράθυρο στην ενότητα αναφοράς, όπως απεικονίζεται στο Σχήμα 3.4.4β. Η προβολή λεπτομερειών αποτελείται από τρία στοιχεία. Μια βιντεοσκόπηση ολόκληρης της εκτέλεσης της ροής, όπου μπορεί κανείς απλά να κάνει κλικ στο κουμπί αναπαραγωγής στο πρόγραμμα αναπαραγωγής βίντεο για να επανεξετάσετε ολόκληρη την εκτέλεση. Μια έκδοση εντοπισμού σφαλμάτων του καμβά σχεδίασης, όπου το ενεργό μπλοκ συναρμολόγησης επισημαίνεται με πορτοκαλί περίγραμμα. Τέλος, το αρχείο

καταγραφής δραστηριοτήτων που περιέχει πολύτιμες πληροφορίες εντοπισμού σφαλμάτων από τα επιμέρους μπλοκ συναρμολόγησης.

Και τα τρία στοιχεία συσχετίζονται μεταξύ τους και μπορούν να επιθεωρηθούν ταυτόχρονα, επομένως, αν σαρώσει κανείς το βίντεο, το πορτοκαλί πλαίσιο θα μετακινηθεί μεταξύ των μπλοκ συναρμολόγησης στον καμβά σχεδίασης και το στοιχείο που επισημαίνεται στο αρχείο καταγραφής δραστηριοτήτων θα αλλάξει ανάλογα. Η επιλογή ενός στοιχείου στο αρχείο καταγραφής δραστηριοτήτων από την άλλη πλευρά, θα μετακινήσει το βίντεο στη χρονική στιγμή όπου προστέθηκε η εγγραφή στο αρχείο καταγραφής δραστηριοτήτων.

Ο συνδυασμός της προβολής εντοπισμού σφαλμάτων, του βίντεο και του αρχείου καταγραφής δραστηριοτήτων είναι ένα πολύ ισχυρό και διαισθητικό εργαλείο για την εύρεση των αιτιών αποτυχίας και τον χειρισμό απροσδόκητης συμπεριφοράς στις ροές αυτοματοποιημένων ελέγχων.

Τα καταγεγραμμένα βίντεο αποθηκεύονται ως αρχεία τύπου AVI και είναι προσβάσιμα απευθείας στο μηχάνημα του ελεγκτή, ανάλογα με τις ρυθμίσεις ασφαλείας για το μηχάνημα του ελεγκτή. Για να λάβει κανείς έναν άμεσο σύνδεσμο για ένα βίντεο, χρειάζεται να κάνει δεξί κλικ στο βίντεο και να επιλέξει 'Get video file link' (Λήψη συνδέσμου αρχείου βίντεο). Ο άμεσος σύνδεσμος για το βίντεο αντιγράφεται στο πρόχειρο και μπορεί να εισαχθεί σε μηνύματα ηλεκτρονικού ταχυδρομείου κ.λπ. [33]

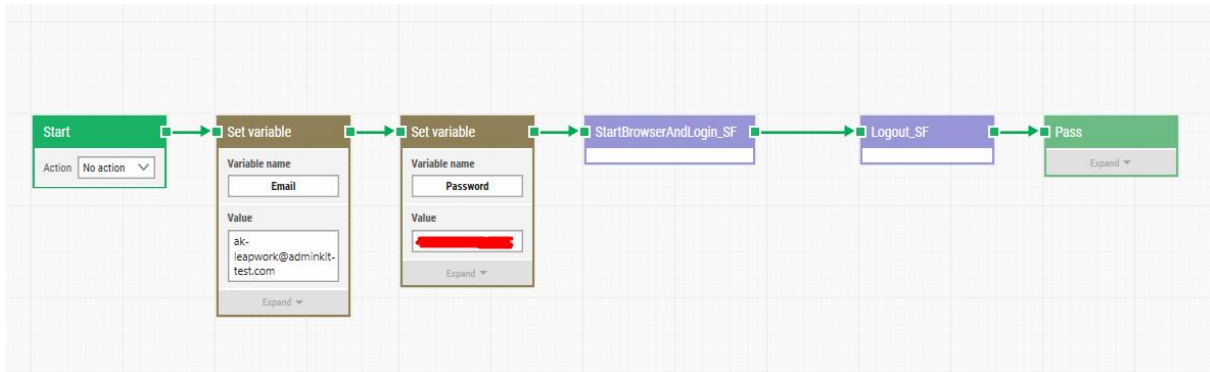


Σχήμα 3.4.4β: Reporting.

3.5 Αυτοματοποιημένοι Έλεγχοι που παρήχθησαν

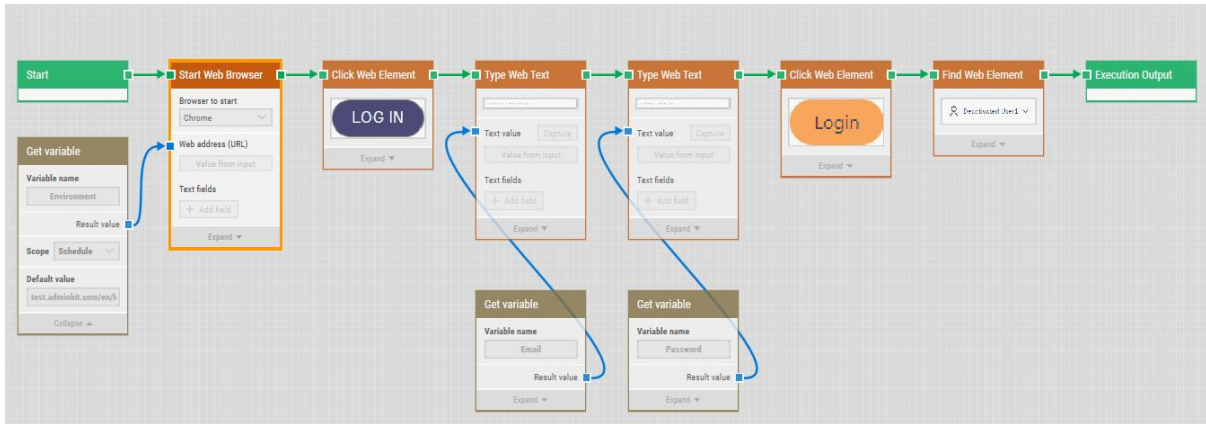
Σε αυτό το κεφάλαιο θα αναλύσουμε μερικές από τις ροές αυτοματοποιημένου ελέγχου που έχουν παραχθεί και εφαρμόζονται για τον έλεγχο του λογισμικού της AdminKit. Πριν προχωρήσουμε στην ανάλυση, αξίζει να σημειώσουμε ότι όλοι οι έλεγχοι εκτελούνται σε ένα περιβάλλον αποκλειστικά και μόνο για την εκτέλεση των ελέγχων. Επίσης, έχουν δημιουργηθεί μερικά προεπιλεγμένα δεδομένα ελέγχου (default testing data) τα οποία αποτελούνται από προκαθορισμένους χρήστες, emails, εταιρείες, αρχεία κ.τ.λ, τα οποία χρησιμοποιούνται πολλές φορές ως τιμές μεταβλητών ή ως αναφορές για την ανάπτυξη των ροών ελέγχου.

Αρχικά, ας εξετάσουμε τη ροή αυτοματοποιημένου ελέγχου 'Login and Logout', Σχήμα 3.5α. Σε αυτό το σενάριο ελέγχου στόχος είναι να συνδεθούμε και να αποσυνδεθούμε στην εφαρμογή της AdminKit, με τον χρήστη 'ak-leapwork@adminikit-test.com', όπου είναι και ο διαχειριστής της εταιρείας 'adminikit-leapwork'.



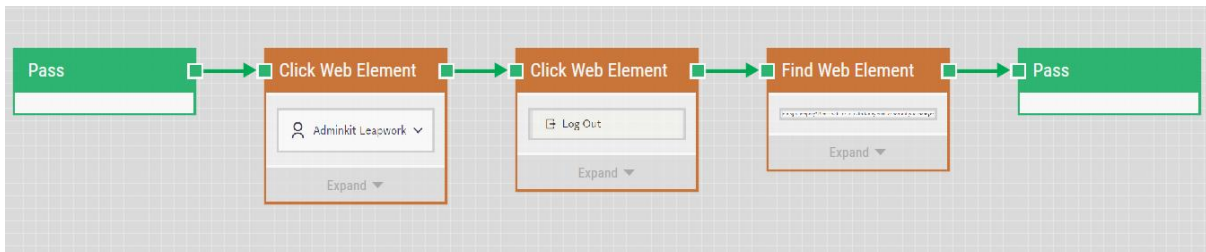
Σχήμα 3.5α: Login and Logout.

Το πρώτο μπλοκ, 'Start', το συναντάμε σε όλους τους ελέγχους που δημιουργούμε και η χρήση του είναι να ξεκινήσει η ροή ελέγχου. Στη συνέχεια, κάνοντας κλικ στην δεξιά άκρη του μπλόκ 'Start', τραβάμε τον πράσινο συνδετήρα και τον ενώνουμε με το επόμενο μπλοκ. Τα δύο επόμενα μπλοκ ονομάζονται 'Set variable' και η χρήση τους είναι να ορίσουμε μια μεταβλητή και να της δώσουμε μια τιμή. Στη συγκεκριμένη περίπτωση, αλλά και στην πλειοψηφία των ελέγχων, η μεταβλητή 'email' παίρνει την τιμή 'ak-leapwork@adminkit-test.com' που είναι το email του διαχειριστή της εταιρείας και η μεταβλητή 'password' που παίρνει τον κωδικό χρήστη (αποκρύβεται για λόγους ασφάλειας). Στη συνέχεια, ενώνουμε με τον συνδετήρα το επόμενο μπλοκ που ακολουθεί, το οποίο είναι η υπο-ροή με ονομασία 'StartBrowserAndLogin_SF', όπως φαίνεται στο Σχήμα 3.5β, η οποία χρησιμοποιείται σχεδόν σε όλα τα σενάρια ελέγχου, καθώς η χρήση της είναι να συνδέει τον χρήστη στην εφαρμογή της Adminkit. Πιο αναλυτικά, το μπλοκ 'Start' είναι υπεύθυνο για την εκτέλεση της υπο-ροής. Έπειτα, ενώνουμε με πράσινο συνδετήρα το 'Start' με το μπλοκ 'Start Web Browser' ορίζοντας τον φυλλομετρητή ιστού, στη συγκεκριμένη περίπτωση έχουμε επιλέξει τον Chrome, στον οποίο θέλουμε να εκτελεστεί τον έλεγχο. Επίσης, πρέπει να δώσουμε και ένα URL το οποίο διαβάζουμε από το μπλοκ 'Get variable', που έχουμε ορίσει στην μεταβλητή 'Environment' την τιμή 'test.adminkit.com/en/lobby', που αντιστοιχεί στο περιβάλλον ελέγχου της Adminkit και ενώνουμε με μπλε συνδετήρα από το σημείο 'Result value' του μπλοκ 'Get variable' προς το σημείο 'Web address' του μπλοκ 'Start Web Browser'. Το επόμενο μπλοκ είναι το 'Click Web Element' το οποίο προσομοιώνει την ενέργεια κλικ πάνω σε ένα συγκεκριμένο στοιχείο στο διαδίκτυο, στην προκειμένη περίπτωση γίνεται κλικ πάνω στο κουμπί 'LOG IN'. Εφόσον πατηθεί το κουμπί, μεταφερόμαστε σε μια άλλη σελίδα στην οποία πρέπει να εισάγουμε τα στοιχεία του χρήστη. Στα δύο επόμενα μπλοκ με τίτλο 'Type Web Text' έχουμε επιλέξει τα πεδία εισαγωγής email και κωδικού στα οποία πληκτρολογούμε το email και τον κωδικό πρόσβασης χρησιμοποιώντας τα μπλοκ 'Get variable', χρησιμοποιώντας ξανά τους μπλε συνδετήρες για να πάρουμε τις τιμές που θέλουμε από το 'Result value' προς τα αντίστοιχα 'Text value' των μπλοκ 'Type Web Text', για να πάρουμε τις τιμές που είχαμε ορίσει νωρίτερα, έξω από την υπο-ροή, στις μεταβλητές 'Email' και 'Password'. Μετά ενώνουμε με ένα μπλοκ 'Click Web Element' για να κάνουμε κλικ σε ένα κουμπί 'Login', ώστε να συνδεθούμε στην εφαρμογή. Το επόμενο μπλοκ με όνομα 'Find Web Element' χρησιμοποιείται για να βρούμε ένα στοιχείο που εμφανίζεται στην αρχική σελίδα της εφαρμογής για να βεβαιωθούμε ότι έχουμε συνδεθεί σωστά. Το στοιχείο αυτό είναι μια λίστα επιλογών για τον συνδεδεμένο χρήστη. Τέλος, το μπλοκ 'Execution Output' αποτελεί το τελευταίο βήμα της υπο-ροής και ορίζει την λήξη της εκτέλεσης της.



Σχήμα 3.5β: StartBrowserAndLogin_SF.

Στη συνέχεια της ροής ελέγχου συναντάμε ακόμη μια υπο-ροή με όνομα ‘Logout_SF’ η οποία είναι υπεύθυνη για την αποσύνδεση μας από την εφαρμογή. Αναλυτικότερα, η υπο-ροή ξεκινάει με ένα πράσινο μπλοκ ‘Pass’ το οποίο σηματοδοτεί την έναρξη της υπο-ροής. Τα πράσινα μπλοκ (Pass, Execution Input, Start) εφόσον βρίσκονται στην αρχή μιας υπο-ροής σημαίνει ότι έχουν επαληθευτεί τα προηγούμενα βήμα πριν την υπο-ροή, οπότε μπορούν να πυροδοτήσουν την έναρξη της νέας υπο-ροής. Το πρώτο μπλοκ ελέγχου σε αυτή την υπο-ροή είναι το ‘Click Web Element’ στο οποίο κάνουμε κλικ πάνω στο όνομα του χρήστη για να εμφανιστεί μια λίστα επιλογών. Αφού εμφανιστεί η λίστα αυτή, χρησιμοποιούμε ένα ακόμη μπλοκ ‘Click Web Element’, κάνοντας κλικ πάνω στην επιλογή ‘Log Out’. Το επόμενο μπλοκ που θα χρησιμοποιήσουμε είναι το ‘Find Web Element’ το οποίο θα ψάξει να βρει ένα στοιχείο κειμένου, το οποίο γνωρίζουμε ότι εμφανίζεται στην σελίδα που μεταφερόμαστε όταν αποσυνδεόμαστε από την εφαρμογή, ώστε να επιβεβαιώσουμε ότι έγινε η αποσύνδεση. Τέλος, το μπλοκ ‘Pass’ αποτελεί το τελευταίο βήμα της υπο-ροής ορίζει την λήξη της εκτέλεσης του.



Σχήμα 3.5γ: Logout_SF.

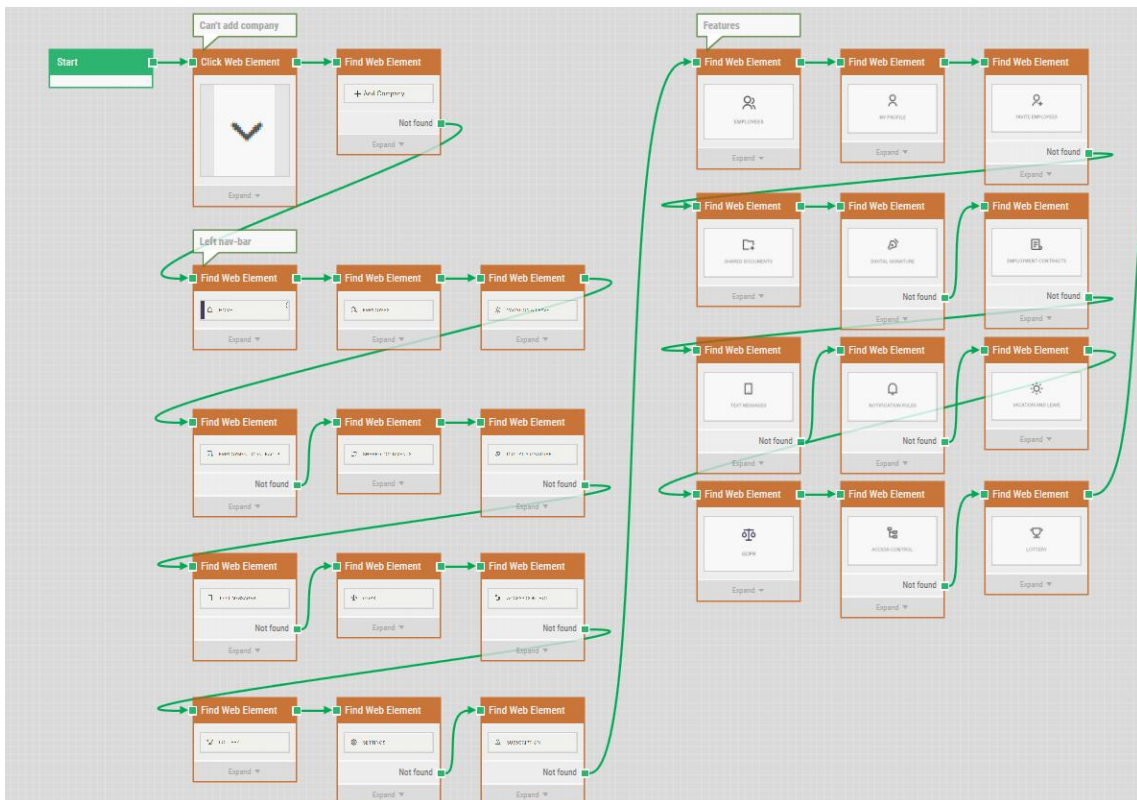
Το τελευταίο βήμα του σεναρίου αποτελεί το μπλοκ ‘Pass’, που ενώνεται με την υπο-ροή ‘Logout_SF’, και σηματοδοτεί τον τερματισμό της ροής ελέγχου, αναγράφοντας ένα μήνυμα ‘Pass’ σε περίπτωση που ο έλεγχος ήταν θετικό ή ‘Failed’ εάν υπήρξε κάποιο πρόβλημα.

Το σενάριο ‘Employee Correct Features’, Σχήμα 3.5δ, εξετάζει ποιες λειτουργίες είναι διαθέσιμες για έναν υπάλληλο και ποιες όχι. Αρχικά, το μπλοκ ‘Start’ σηματοδοτεί την έναρξη της εκτέλεσης του σεναρίου ελέγχου. Έπειτα ενώνουμε, διαδοχικά, με τον πράσινο συνδετήρα δύο μπλοκ ‘Set variable’ για να ορίσουμε το email και τον κωδικό πρόσβασης, όμως αυτή την φορά θα δώσουμε την τιμή ‘employee1@adminkit-test.com’ στη μεταβλητή ‘Email’, που αντιστοιχεί σε έναν λογαριασμό απλού χρήστη. Στη συνέχεια της ροής ελέγχου συναντάμε την υπο-ροή ‘StartBrowserAndLogin_SF’, αναλύθηκε στο προηγούμενο σενάριο, ώστε να συνδεθούμε στην εφαρμογή της Adminkit.



Σχήμα 3.5δ: Employee Correct Features.

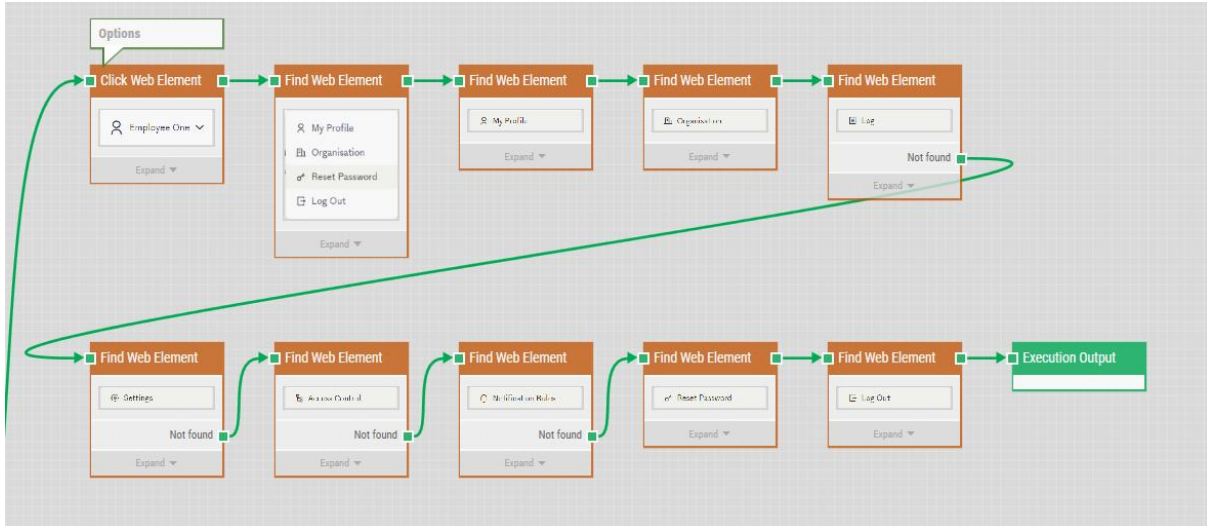
Το επόμενο μπλοκ είναι ακόμη μια υπο-ροή με τίτλο ‘EmployeeFeatures_SF’, απεικονίζεται στα σχήμα 3.53 και 3.5στ, η οποία είναι υπεύθυνη για τον εντοπισμό των σωστών λειτουργιών που αντιστοιχούν σε έναν υπάλληλο της εταιρείας. Η υπο-ροή ξεκινάει με ένα μπλοκ ‘Start’ για να ξεκινήσει η εκτέλεση της και έπειτα ενώνεται με ένα μπλοκ τύπου ‘Click Web Element’ το οποίο κάνει κλικ πάνω σε ένα κουμπί το οποίο ανοίγει μια λίστα με τις εταιρίες στις οποίες είναι εγγεγραμμένος ο χρήστης. Το επόμενο μπλοκ είναι της κατηγορίας ‘Find Web Element’ όπου ψάχνουμε για ένα κουμπί με τίτλο ‘+ Add Company’. Σε αυτή την περίπτωση δεν θέλουμε να βρούμε αυτό το κουμπί, καθώς το κουμπί αυτό εμφανίζεται μόνος στους συνδεδεμένους χρήστες της κατηγορίας των διαχειριστών. Στη περίπτωση που θέλουμε ο έλεγχος να μην βρίσκει ένα συγκεκριμένο στοιχείο, δεν τραβάμε τον πράσινο συνδετήρα από το πάνω μέρος του μπλοκ ώστε να συνεχίσουμε στα επόμενα βήματα, αλλά τραβάμε τον συνδετήρα από την επιλογή ‘Not found’ που σημαίνει ότι είναι επιθυμητό ο έλεγχος να ψάχνει και να μην βρίσκει το συγκεκριμένο στοιχείο. Στην συνέχεια ακολουθούν πολλά μπλοκ τύπου ‘Find Web Elements’ τα οποία βρίσκουν ή δεν βρίσκουν τις διαθέσιμες επιλογές ενός υπαλλήλου στην αριστερή μπάρα επιλογών ή στο μενού επιλογών.



Σχήμα 3.5ε EmployeeFeatures_SF.

Κεφάλαιο 3

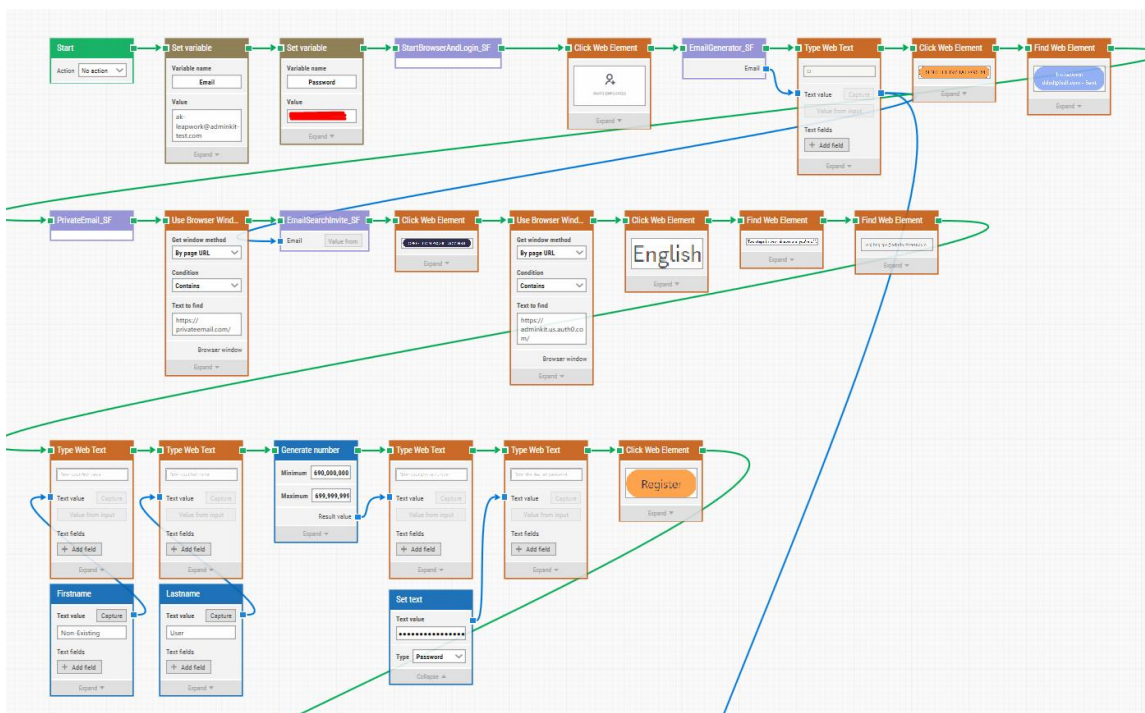
Στη συνέχεια, όπως φαίνεται στο Σχήμα 3.5στ, γίνεται κλικ πάνω στο όνομα χρήστη, χρησιμοποιώντας το μπλοκ ‘Click Web Element’ το οποίο ανοίγει μια λίστα με διάφορες επιλογές. Εφόσον ανοίξει η λίστα επιλογών, ακολουθούν διαδοχικά μπλοκ ‘Find Web Element’ τα οποία αναζητούν επιλογές που θα έπρεπε να είναι εμφανείς ή όχι προς τον υπάλληλο. Ολοκληρώνοντας την υπο-ροή, συναντάμε το μπλοκ ‘Execution Output’ που τερματίζει την υπο-ροή και δίνει την έγκριση για την συνέχεια της ροής ελέγχου έξω από την υπο-ροή.



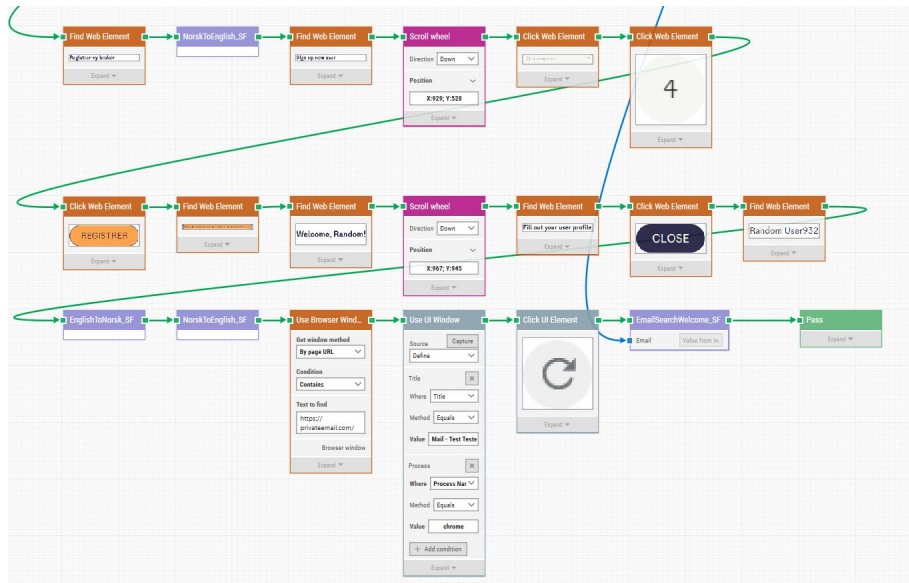
Σχήμα 3.5στ: EmployeeFeatures_SF.

Τέλος, το μπλοκ ‘Pass’ σηματοδοτεί τον τερματισμό της ροής ελέγχου, αναγράφοντας ένα μήνυμα ‘Pass’ σε περίπτωση που έλεγχος πέρασε με επιτυχία ή ‘Failed’ εάν απέτυχε.

Το σενάριο ‘Invite Non-Existing User’ απεικονίζεται στα Σχήμα 3.5ζ και Σχήμα 3.5η και προσομοιώνει μία πρόσκληση εγγραφής προς την εφαρμογή της Adminkit και την διαδικασία εγγραφής ενός νέου χρήστη.

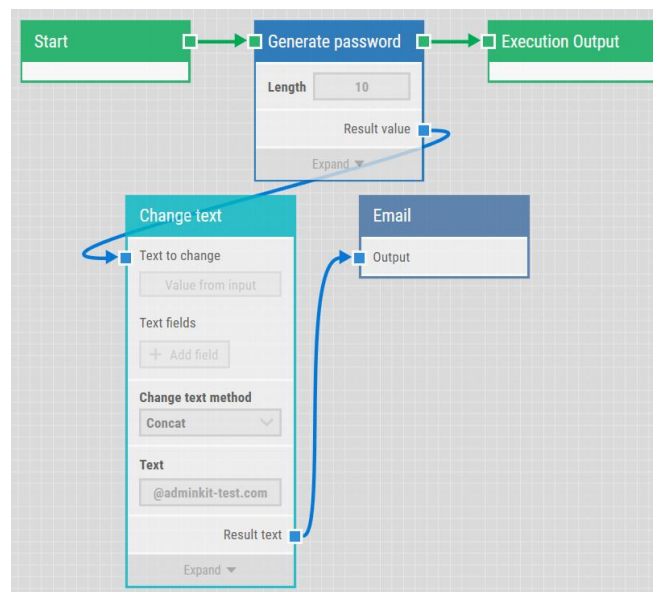


Σχήμα 3.5ζ: Invite Non-Existing User.



Σχήμα 3.5η: Invite Non-Existing User.

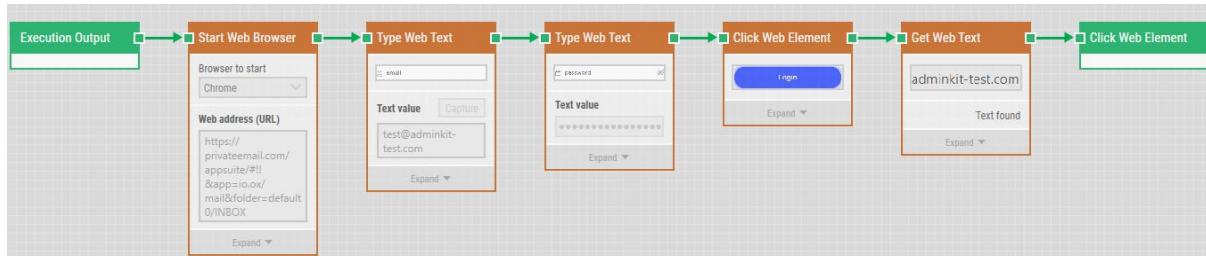
Όπως στα παραπάνω σενάρια, αρχικά ξεκινάμε την εκτέλεση της ροής ελέγχου, ορίζουμε τις μεταβλητές για email και κωδικό ώστε να συνδεθούμε στην εφαρμογή ως διαχειριστής. Χρησιμοποιούμε το μπλοκ ‘Click Web Element’ για να κάνουμε κλικ πάνω στο κουμπί ‘INVITE EMPLOYEES’, το οποίο θα ανοίξει ένα νέο παράθυρο. Στο επόμενο βήμα συναντάμε την υπο-ροή με τίτλο ‘EmailGenerator_SF’, η οποία έχει δημιουργηθεί για να δημιουργεί τυχαία emails, Σχήμα 3.5θ. Αναλυτικότερα, χρησιμοποιεί το μπλοκ ‘Generate password’ το οποίο παίρνοντας την τιμή ‘10’, δημιουργεί μια συμβολοσειρά (string) με δέκα (10) τυχαίους χαρακτήρες. Στη συνέχεια η συμβολοσειρά αυτή περνάει με μπλε συνδετήρα στο μπλοκ ‘Change text’ το οποίο με την επιλογή ‘Concat’ ενώνει η συμβολοσειρά με το κείμενο που έχουμε ορίσει (@adminkit-test.com) ώστε να δημιουργηθεί ένα τυχαίο email αλλά με σταθερή διεύθυνση (domain). Το τελικό αποτέλεσμα αποθηκεύεται στην μεταβλητή εξόδου ‘Email’.



Σχήμα 3.5θ: EmailGenerator_SF.

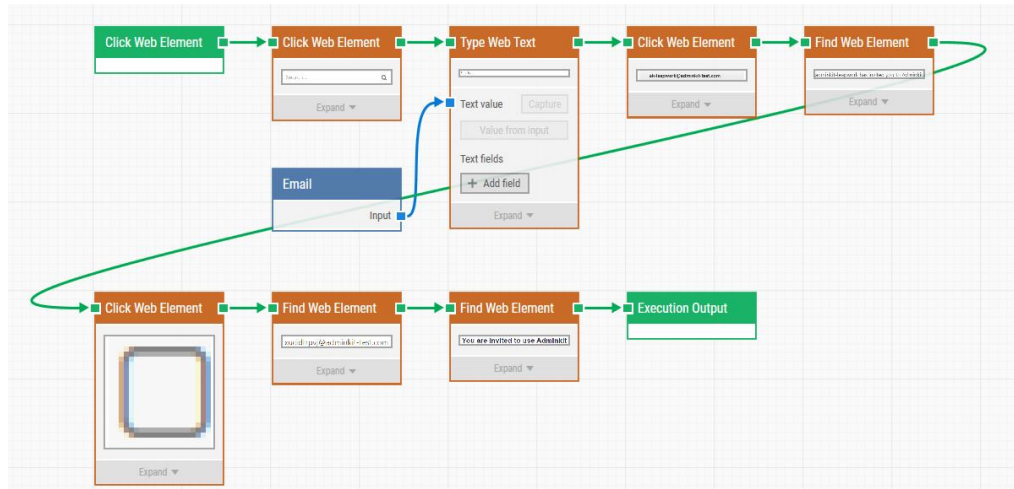
Στη συνέχεια, το email που δημιουργήθηκε περνάει με μπλε συνδετήρα ως είσοδος στο μπλοκ ‘Type Web Element’ για να τυπωθεί στην φόρμα εισαγωγής και έπειτα να πατηθεί το κατάλληλο κουμπί

ώστε να αποσταλεί η πρόσκληση. Το επόμενο βήμα είναι ένα ακόμη μια υπο-ροή που ονομάζεται 'PrivateEmail_SF', όπως φαίνεται στο Σχήμα 3.5i, η οποία ανοίγει ένα καινούργιο παράθυρο εργασίας στον φυλλομετρητή με διεύθυνση την υπηρεσία Private Email⁴ που χρησιμοποιούμε για τα δοκιμαστικά emails. Σε αυτή την υπο-ροή, δακτυλογραφούμε το όνομα χρήστη και τον κωδικό του λογαριασμού μας στην υπηρεσία, ώστε να συνδεθούμε στην υπηρεσία και να δεχόμαστε emails που είναι διαθέσιμα για ένα μικρό χρονικό διάστημα.



Σχήμα 3.5i: PrivateEmail_SF.

Το επόμενο μπλοκ που χρησιμοποιούμε μετά τη έξοδο από την προηγούμενη υπο-ροή, είναι το 'Use Browser Window' στο οποίο δίνοντας τις κατάλληλες οδηγίες, ελέγχουμε ότι χρησιμοποιούμε πλέον τη νέα σελίδα που ανοίξαμε και όχι την προηγούμενη. Αυτό επιτυγχάνεται επιλέγοντας το 'Contains' κάτω από την επιλογή 'Condition' και γράφοντας το σωστό URL στο πεδίο 'Text to find'. Το επόμενο βήμα είναι άλλη μια υπο-ροή εν ονόματι 'EmailSearchInvite_SF', Σχήμα 3.5ια, η οποία παίρνει ως εισαγωγή το τυχαίο email που δημιουργήθηκε στα προηγούμενα βήματα, ψάχνει στην αναζήτηση της πλατφόρμας για τα απεσταλμένα emails προς αυτή την διεύθυνση και ανοίγει την σωστή πρόσκληση διαβάζοντας τον σωστό τίτλο.



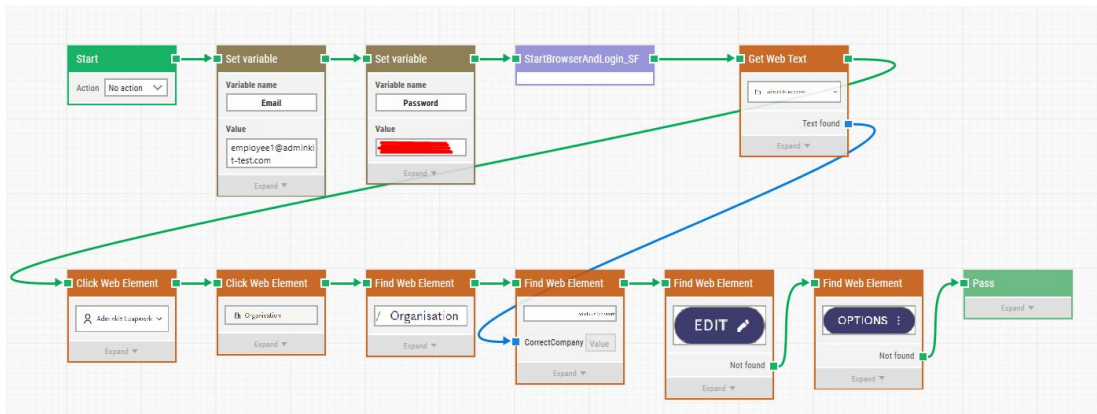
Σχήμα 3.5ια: EmailSearchInvite_SF.

Στη συνέχεια κάνουμε κλικ στο κουμπί της εγγραφής που ανοίγει ένα νέο παράθυρο και χρησιμοποιούμε ξανά το μπλοκ που ελέγχει το ότι βρισκόμαστε σε διαφορετικό παράθυρο. Σε αυτό το στάδιο ακολουθούνται τα απαραίτητα βήματα για την εγγραφή ενός χρήστη, χρησιμοποιώντας μπλοκς για να δώσουμε τιμές σε πεδία, να κάνουμε κλικ, να βρούμε αντικείμενα αλλά και να κυλίσουμε στην επιφάνεια εργασίας με το μπλοκ 'Scroll wheel'. Εφόσον ολοκληρωθεί η εγγραφή του καινούργιου χρήστη, το σενάριο θα τελειώσει με τον έλεγχο της λήψης ενός email καλωσορίσματος προς τον νέο χρήστη. Για να γίνει αυτό εφικτό, θα χρησιμοποιηθεί ξανά το μπλοκ 'Use Browser Window' ώστε να

⁴ <https://privateemail.com/#&ref=!%26app%3Dio.ox%2Fmail%26folder%3Ddefault0%2FINBOX%26storeLocale%3Dtrue>

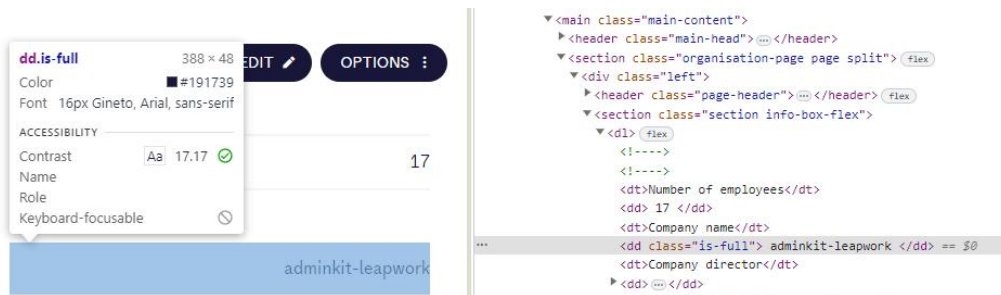
γυρίσουμε πίσω στο παράθυρο που είναι ανοιχτή η υπηρεσία του Private email. Έπειτα, θα χρησιμοποιηθούν τα μπλοκ ‘Use UI Window’ και ‘Click UI Element’ ώστε να γίνει κλικ στο κουμπί της επαναφοράς του φυλλομετρητή. Με τον τρόπο αυτό ανανεώνουμε την σελίδα και κερδίζουμε χρόνο για να σιγουρευτούμε ότι το email καλωσορίσματος έχει προλάβει να σταλθεί. Τέλος, ακολουθούμε η υπο-ροή ‘EmailSearchWelcome_SF’ που εκτελεί την ακολουθία βημάτων όπως το ‘EmailSearchInvite_SF’ αλλά αυτή την φορά ψάχνει για διαφορετικό τίτλο στο θέμα του email. Τέλος, το μπλοκ ‘Pass’ ολοκληρώνει την εκτέλεση του σεναρίου.

Στο σενάριο ‘Employee can’t Edit Organisation’, που εμφανίζεται στο Σχήμα 3.5ιβ, ελέγχουμε ότι ένας απλός υπάλληλος δεν μπορεί να αλλάξει τις ρυθμίσεις της εταιρείας που δουλεύει, καθώς τα δικαιώματα για να αλλάξει τις ρυθμίσεις έχει μόνο ο διαχειριστής.

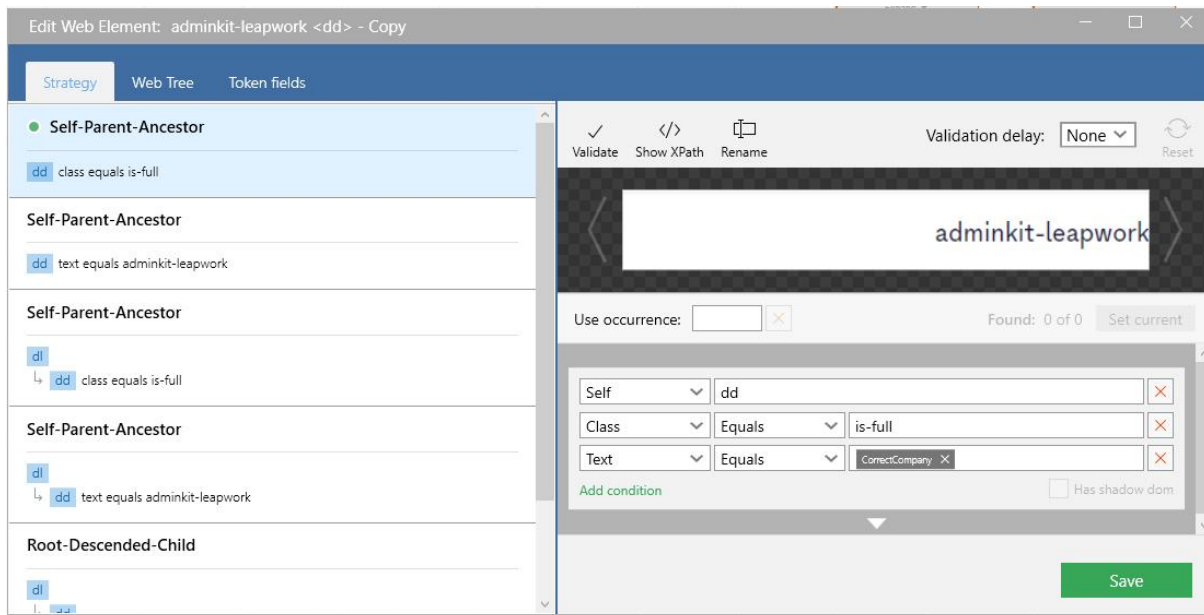


Σχήμα 3.5ιβ: Employee can’t Edit Organisation.

Αρχικά ξεκινάμε την εκτέλεση της ροής ελέγχου, ορίζουμε τις μεταβλητές για email και κωδικό ώστε να συνδεθούμε στην εφαρμογή ως υπάλληλος. Στο επόμενο βήμα χρησιμοποιούμε το μπλοκ ‘Get Web Text’ το οποίο μας επιτρέπει να αποθηκεύσουμε το κείμενο ενός συγκεκριμένου στοιχείου, στη συγκεκριμένη περίπτωση θέλουμε να αποθηκεύσουμε το όνομα της τρέχων εταιρείας (adminkit-learwork) από την λίστα των εταιρειών. Στη συνέχεια κάνουμε μερικά κλικ ώστε να βρεθούμε στην σελίδα με τις πληροφορίες της εταιρείας. Αμέσως μετά, θα χρησιμοποιήσουμε το ‘Find Web Element’ στο σημείο όπου αναγράφεται το όνομα της εταιρείας. Εκεί θα δημιουργήσουμε μια μεταβλητή ‘CorrectCompany’ και θα της δώσουμε την τιμή ‘adminkit-learwork’ που έχουμε αποθηκεύσει από το μπλοκ ‘Get Web Text’. Κάνοντας δεξί κλικ πάνω στο μπλοκ ‘Get Web Element’, επιλέγοντας το ‘Edit Web Element’ μπορούμε να δούμε την δομή DOM, Σχήμα 3.5ιγ, του στοιχείου που έχουμε επιλέξει αλλά και να ορίσουμε μεταβλητές σε οποιοδήποτε σημείο επιθυμούμε. Ορίζοντας την μεταβλητή ‘CorrectCompany’ στο σημείο ‘Text’ και τον έλεγχο ‘Equals’, ελέγχουμε ότι το όνομα της εταιρείας, σε μορφή κειμένου, είναι ίδιο με το κείμενο που αποθηκεύσαμε νωρίτερα, Σχήμα 3.5ιδ. Με αυτόν τον τρόπο ελέγχουμε ότι έχουμε συνδεθεί στην σωστή εταιρεία.



Σχήμα 3.5ιγ: Το κείμενο ‘adminkit-learwork’ από την σκοπιά του Chrome DevTools.

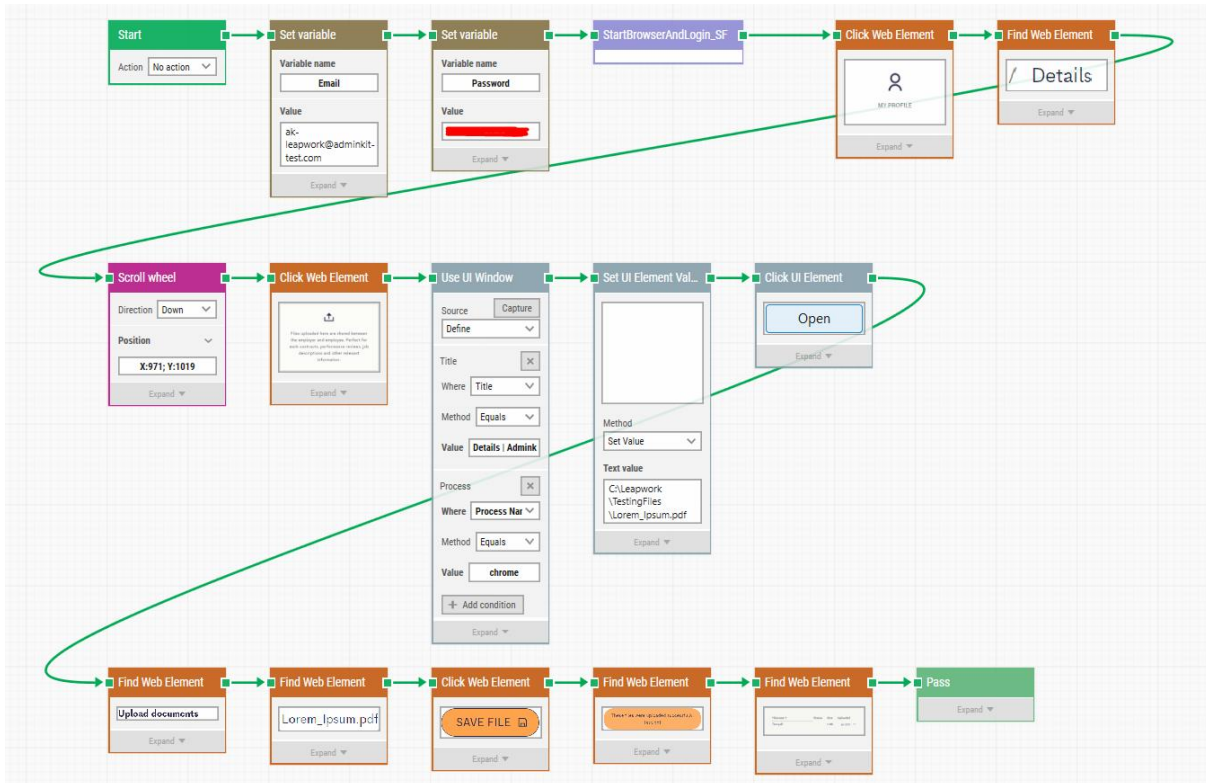


Σχήμα 3.5ιδ: Επιλογή 'Edit Web Element' στο μπλοκ 'Get Web Text'.

Τέλος, χρησιμοποιούμε δυο μπλοκ 'Find Web Element' για δυο κουμπιά που δεν θέλουμε να βρούμε, καθώς είναι ενεργοποιημένα μόνο για τον διαχειριστή της εταιρείας, οπότε συνεχίζουμε τα βήματα τραβώντας το συνδετήρα από το σημείο 'Not found' ώστε να περάσουμε στο επόμενο βήμα εφόσον το στοιχείο που έχουμε ορίσει στο μπλοκ δεν βρέθηκε. Καταλήγοντας στο τελευταίο μπλοκ 'Pass'.

Το σενάριο ελέγχου 'Upload a Document' που απεικονίζεται στο Σχήμα 3.5ιε, προσομοιώνει τη μεταφόρτωση ενός αρχείου τύπου .pdf στο προσωπικό προφίλ του διαχειριστή. Όπως κάθε ροή ελέγχου, ξεκινάμε με το μπλοκ 'Start' για να ξεκινήσει η εκτέλεση και τραβάμε τον πράσινο συνδετήρα τα δύο διαδοχικά μπλοκ 'Set variable' στα οποία ορίζουμε το email και τον κωδικό του διαχειριστή. Έπειτα χρησιμοποιούμε την υπο-ροή 'StartBrowserAndLogin_SF' ώστε να συνδεθούμε στην εφαρμογή. Στη συνέχεια, με το μπλοκ 'Click Web Element' επιλέγουμε να κάνουμε κλικ πάνω στο κουμπί 'MY PROFILE' το οποίο μας κατευθύνει στο προφίλ του διαχειριστή. Για να επιβεβαιώσουμε την μεταφορά προς τη σελίδα του προφίλ, θα χρησιμοποιήσουμε το μπλοκ 'Find Web Element' ψάχνοντας ένα στοιχείο που συναντάτε μόνο στη σελίδα προφίλ για να ελέγξουμε ότι η μεταφορά έχει γίνει ορθά. Στη συγκεκριμένη περίπτωση το στοιχείο που ψάχνουμε είναι το κείμενο 'Details' που απεικονίζεται στην κεφαλίδα της σελίδας. Έπειτα, με το μπλοκ 'Scroll wheel' σκρολάρουμε προς τα κάτω ώστε να εμφανιστεί στην οθόνη το επόμενο στοιχείο που ψάχνουμε. Θέλουμε να κάνουμε κλικ πάνω στο πεδίο για την μεταμόρφωση ενός αρχείου, για αυτό και χρησιμοποιούμε το μπλοκ 'Click Web Element'. Μετά το κλικ στο πεδίο μεταφόρτωσης αρχείου, θα ανοίξει ένα παράθυρο των Windows, επομένως θα πρέπει να χρησιμοποιήσουμε μπλοκ διαφορετικού τύπου. Το μπλοκ 'User UI Window' αναγνωρίζει το παράθυρο στην επιφάνεια εργασίας των Windows στο οποίο θα γίνουν οι επόμενες ενέργειες. Στο μπλοκ 'Set UI Element Value' επιλέγουμε ένα πεδίο εισόδου από το παράθυρο και στο πεδίο του μπλοκ με όνομα 'Text value' ορίζουμε τη διαδρομή στην οποία βρίσκεται το αρχείο που θέλουμε να μεταφορτώσουμε. Σε αυτό το σημείο, αξίζει να σημειώσουμε ότι τα αρχεία που χρησιμοποιούνται σε τέτοιου είδους ελέγχους, είναι αποθηκευμένα σε έναν συγκεκριμένο φάκελο του υπολογιστή, γνωρίζοντας την τοποθεσία και το περιεχόμενό του, έτσι ώστε τα αρχεία ελέγχου να είναι οργανωμένα και να επιλέγεται η σωστή διαδρομή κάθε φορά που θέλουμε να χρησιμοποιήσουμε ένα αρχείο κατά την εκτέλεση ενός ελέγχου. Το τελευταίο μπλοκ που σχετίζεται με τα windows είναι το μπλοκ 'Click UI Element', το οποίο

προβαίνει στην ενέργεια του να κάνουμε κλικ πάνω στο κουμπί ‘Open’, έτσι ώστε να ανέβει το αρχείο στην εφαρμογή, σύμφωνα με τη διαδρομή που συμπληρώσαμε στο πεδίο εισαγωγής. Στη συνέχεια, ακολουθούν δύο μπλοκ ‘Find Web Element’ για να ελέγξουν ότι το αρχείο με όνομα ‘Lorem_Ipsum.pdf’ έχει φορτωθεί στο πεδίο μεταφοράς αρχείων. Το επόμενο μπλοκ ‘Click Web Element’ κάνει κλικ πάνω στο κουμπί ‘SAVE FILE’ ώστε να γίνει η αποθήκευση του αρχείου. Τέλος, συναντάμε δύο μπλοκ ‘Find Web Element’, το πρώτο ελέγχει ότι έχει εμφανιστεί ένα μήνυμα επιτυχής μεταφόρτωσης του αρχείου και το δεύτερο ελέγχει ότι το αρχείο με όνομα ‘Lorem_Ipsum.pdf’ βρίσκεται στη λίστα με τα αποθηκευμένα αρχεία. Όπως πάντα, το μπλοκ ‘Pass’ τερματίζει την εκτέλεση της ροής ελέγχου.

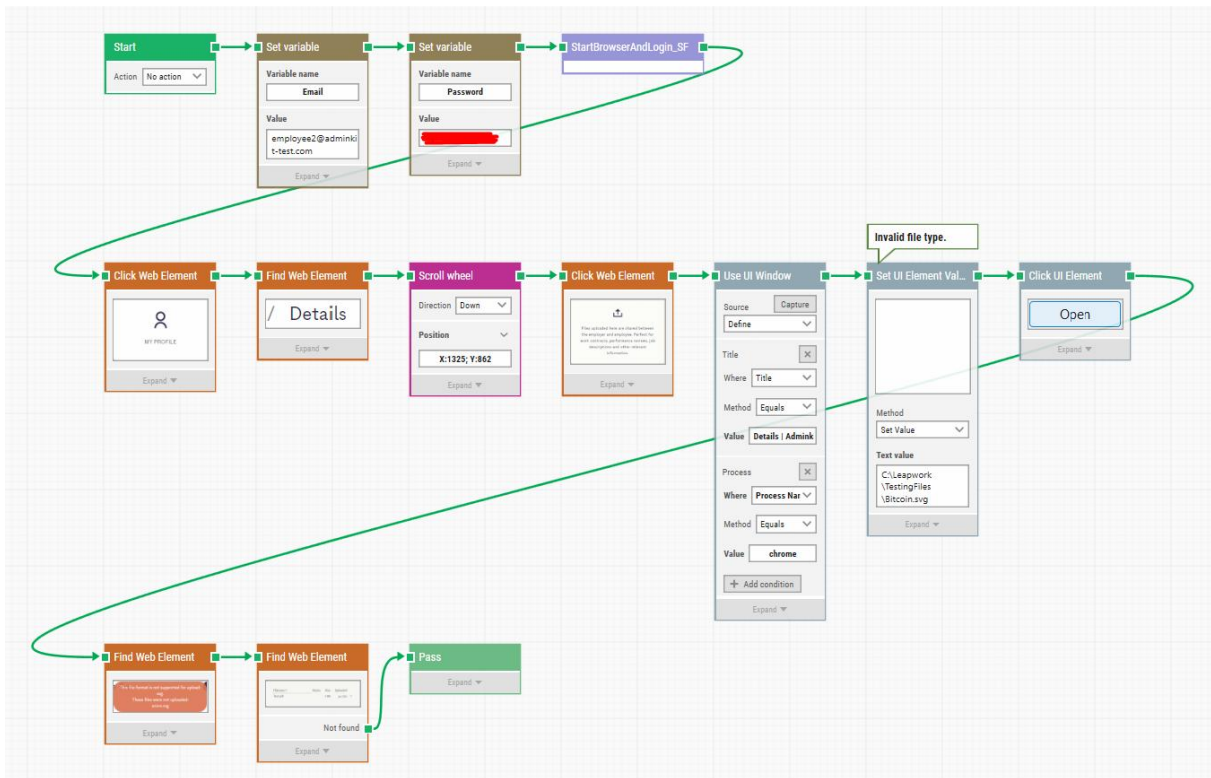


Σχήμα 3.5ιε: Upload a Document.

Το σενάριο ελέγχου ‘Upload Invalid Files’ που απεικονίζεται στο Σχήμα 3.5ιστ, προσομοιώνει την προσπάθεια μεταφόρτωσης ενός αρχείου τύπου .svg, το οποίο δεν υποστηρίζεται για μεταφόρτωση, στο προσωπικό προφίλ ενός υπαλλήλου. Το παρόν σενάριο ακολουθεί τα ίδια βήματα με το προηγούμενο, ‘Upload a Document’, αλλά με μερικές διαφορές. Πρώτη διαφορά είναι ότι συνδεόμαστε με τον λογαριασμό ενός υπαλλήλου, οπότε θέτουμε διαφορετικό email και κωδικό στα μπλοκ ‘Set variable’. Στη συνέχεια, η επόμενη διαφορά συναντάται στο μπλοκ ‘Set UI Element Value’ όπου θέτουμε μια διαφορετική διαδρομή για την επιλογή του αρχείου με όνομα ‘Bitcoin.svg’, το οποίο ανήκει σε μία κατηγορία αρχείων τύπου .svg η οποία δεν υποστηρίζεται προς μεταφόρτωση από την εφαρμογή της Adminkit. Το αναμενόμενο αποτέλεσμα μετά το πάτημα του κουμπιού ‘Open’ θα ήταν η μη μεταφόρτωση του αρχείου, για αυτό και το επόμενο μπλοκ ‘Find Web Element’ ελέγχει ένα μήνυμα λάθους που αναγράφει ότι η μεταφόρτωση του συγκεκριμένου αρχείου δεν είναι δυνατή. Το επόμενο μπλοκ ‘Find Web Element’ ψάχνει το πεδίο με τη λίστα των ανεβασμένων αρχείων. Στον συγκεκριμένο όμως λογαριασμό του υπαλλήλου δεν υπάρχουν ανεβασμένα αρχεία, οπότε η ροή του σεναρίου θα επαληθευτεί εφόσον δεν έχει μεταφορτωθεί το αρχείο ‘Bitcoin.svg’, όπως επίσης δεν θα βρεθεί και η λίστα των ανεβασμένων αρχείων. Οπότε ο πράσινος συνδετήρας θα τραβηχτεί από την

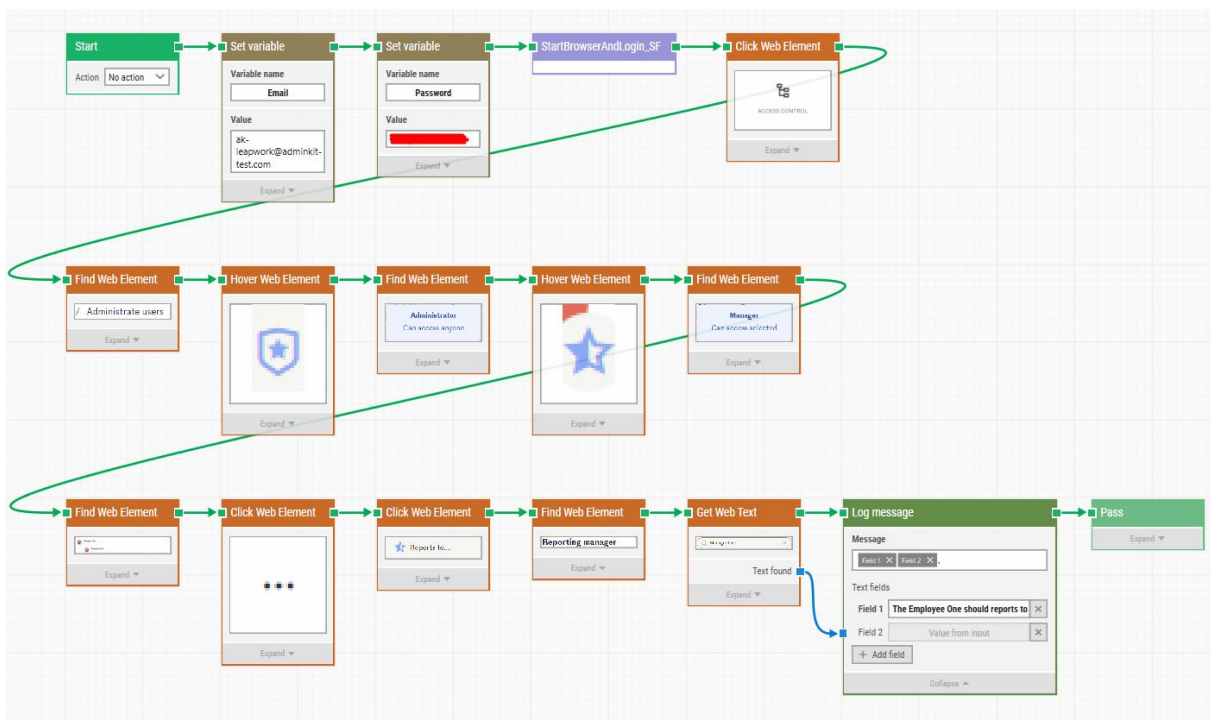
Κεφάλαιο 3

επιλογή 'Not found' του μπλοκ 'Find Web Element' και θα καταλήξει στο μπλοκ 'Pass' που σηματοδοτεί τον τερματισμό της ροής ελέγχου.



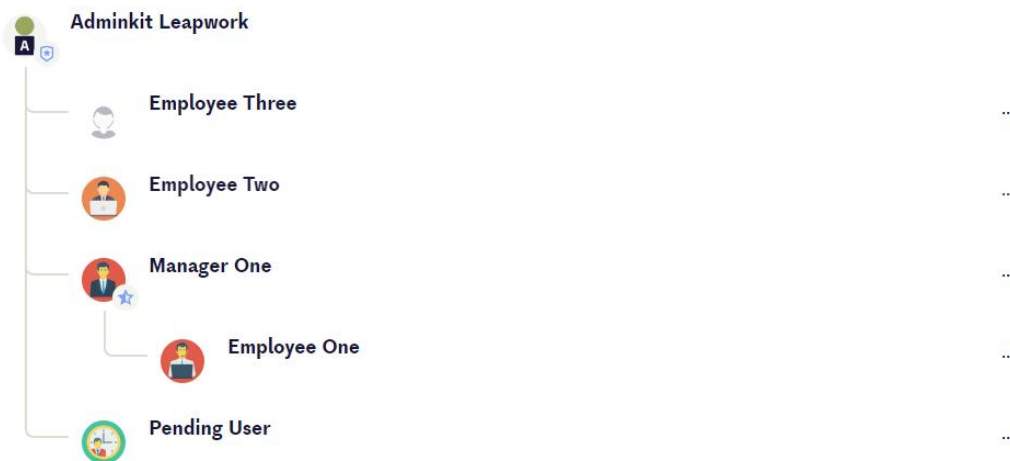
Σχήμα 3.5ιστ: Upload Invalid Files.

Ο έλεγχος ροής 'Access level', όπως φαίνεται στο Σχήμα 3.5ιζ, σε αυτό το σενάριο στόχος είναι να συνδεθούμε ως διαχειριστής, καθώς είναι ο μόνος που έχει πρόσβαση στη λειτουργία 'Access Control', ώστε να ελέγξουμε την ιεραρχική δομή των μελών της εταιρείας.



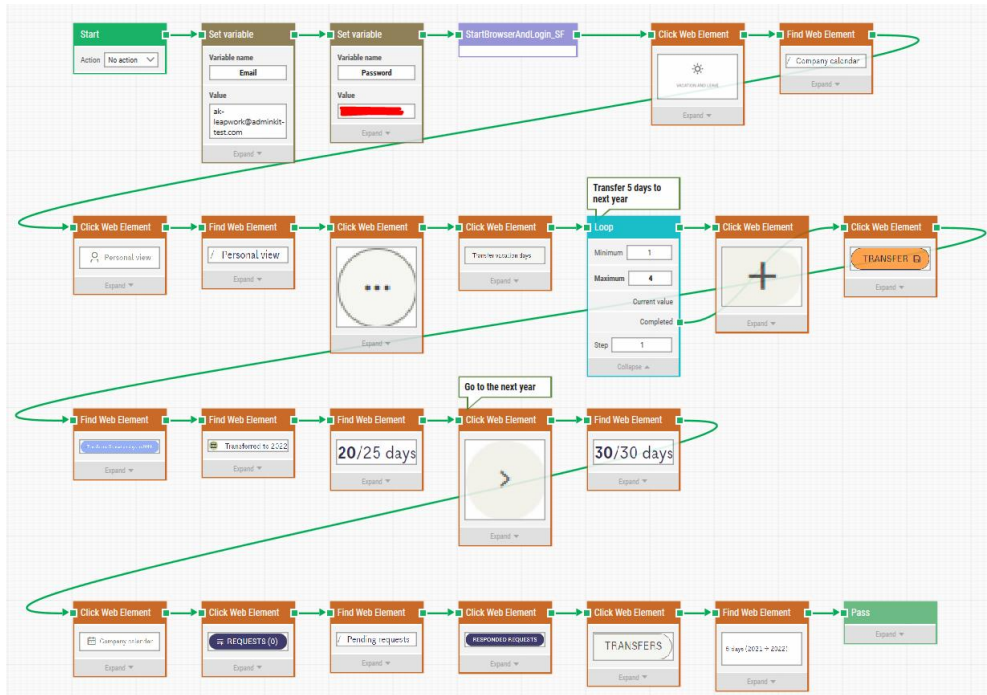
Σχήμα 3.5ιζ: Access level.

Αρχικά ξεκινάμε την εκτέλεση της ροής ελέγχου, ορίζουμε τις μεταβλητές για email και κωδικό ώστε να συνδεθούμε στην εφαρμογή ως διαχειριστής, με τη βοήθεια της υπο-ροής 'StartBrowserAndLogin_SF'. Χρησιμοποιούμε το μπλοκ 'Click Web Element' για να κάνουμε κλικ πάνω στο κουμπί 'ACCESS CONTROL', το οποίο θα μας ανακατευθύνει στην σελίδα με την ιεραρχική δομή της εταιρείας. Το επόμενο μπλοκ 'Find Web Element' ψάχνει το κείμενο 'Administrators' στην κεφαλίδα της σελίδα, έτσι ώστε να επιβεβαιώσουμε ότι βρισκόμαστε στη σωστή σελίδα. Το επόμενο βήμα είναι να ελέγξουμε το ιεραρχικό δέντρο της εταιρείας, όπως απεικονίζεται στο Σχήμα 3.5.η. Χρησιμοποιώντας το μπλοκ 'Hover Web Element' φέρνουμε τον κέρσορα του ποντικού πάνω στο εικονίδιο δίπλα από την φωτογραφία του 'Adminkit Learwork'. Με αυτή την κίνηση, εμφανίζεται ένα ενημερωτικό μήνυμα πως ο συγκεκριμένος χρήστης με αυτό το εικονίδιο είναι ο διαχειριστής, για αυτό και το μπλοκ 'Find Web Element' ψάχνει το μήνυμα με κείμενο 'Administrator Can access anyone'. Τα επόμενα δύο μπλοκ της κατηγορίας 'Hover Web Element' και 'Find Web Element' εκτελούν τις ίδιες ενέργειες με τα προηγούμενα, όμως αυτή την φορά πάνω στον χρήστη 'Manager One'. Ο συγκεκριμένος χρήστης είναι διευθυντής, οπότε ψάχνουμε για ένα διαφορετικό εικονίδιο, σε σχέση με αυτό του διαχειριστή, και το μήνυμα 'Can access selected'. Στη συνέχεια, το μπλοκ 'Find Web Element' ελέγχει ότι ο χρήστης 'Employee One' βρίσκεται ιεραρχικά κάτω από τον 'Manager One'. Έπειτα, κάνουμε κλικ πάνω στις τρεις τελείες που αντιστοιχούν στον 'Employee One' χρησιμοποιώντας το μπλοκ 'Click Web Element'. Εφόσον πατήσουμε τις τρεις τελείες, θα εμφανιστεί μια λίστα επιλογών, όπου θα χρησιμοποιήσουμε το μπλοκ 'Click Web Element' για να κάνουμε κλικ πάνω στην επιλογή 'Reports to...'. Με αυτή την ενέργεια θα ανοίξει ένα βοηθητικό παράθυρο που αναγράφει ποιος είναι ο διευθυντής του υπαλλήλου 'Employee One', για να επιβεβαιώσουμε ότι έχει ανοίξει το σωστό παράθυρο χρησιμοποιούμε το μπλοκ 'Find Web Element', ψάχνοντας την κεφαλίδα 'Reporting manager'. Το επόμενο βήμα είναι να αποθηκεύουμε το κείμενο στο οποίο αναγράφεται ο υπεύθυνος του 'Employee One', ο οποίος είναι ο 'Manager One', με τη βοήθεια του μπλοκ 'Get Web Text'. Στη συνέχεια, χρησιμοποιώντας το μπλοκ 'Log message' μπορούμε να εμφανίσουμε ένα μήνυμα στο αρχείο καταγραφής δραστηριοτήτων, στο συγκεκριμένο σημείο εκτέλεσης της ροής ελέγχου. Το μήνυμα αυτό αναγράφεται στο πεδίο 'Message' όπου έχουμε θέσει τις μεταβλητές 'Field 1' και 'Field 2'. Η πρώτη μεταβλητή έχει ως όρισμα την πρόταση 'The Employee One should reports to', ενώ η δεύτερη, με τη χρήση του μπλε συνδετήρα, παίρνει ως όρισμα την αποθηκευμένη τιμή από το μπλοκ 'Get Web Text'. Έτσι, το τελικό μήνυμα αναμένεται να είναι το 'The Employee One should reports to Manager One.', που μεταφράζεται ως 'Ο Employee One πρέπει να αναφέρεται στον Manager One.'. Η ροή ελέγχου κλείνει με το μπλοκ 'Pass'.



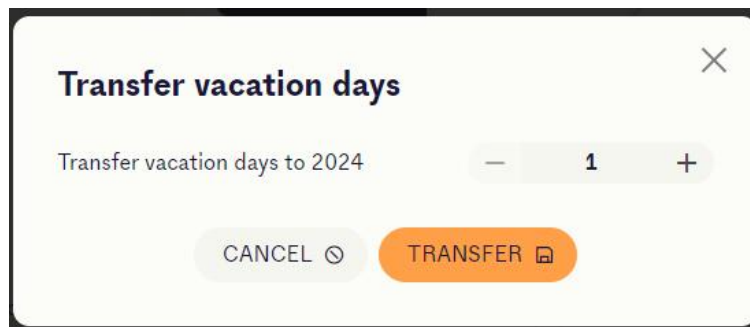
Σχήμα 3.5.η: Access level.

Η ροή ελέγχου ‘Transfer to Next Year as Admin’, Σχήμα 3.5ιθ, ελέγχει το σενάριο στο οποίο ο διαχειριστής συνδέεται στην εφαρμογή, πηγαίνει στη λειτουργία ‘Vacation and Leave’ με σκοπό να μεταφέρει μερικές μέρες αδείας από το τρέχων έτος στο επόμενο.



Σχήμα 3.5ιθ: Transfer to Next Year as Admin.

Όπως συμβαίνει στη πλειοψηφία των ροών ελέγχου, έτσι και σε αυτή την περίπτωση ξεκινάει η εκτέλεση του ελέγχου με το μπλοκ ‘Pass’ και ακολουθούν τα μπλοκ που ορίζουν τις μεταβλητές εισόδου ως διαχειριστής αλλά και η υπο-ροή που είναι υπεύθυνη για την εισαγωγή στην εφαρμογή. Για το επόμενο βήμα χρησιμοποιούμε το μπλοκ ‘Click Web Element’ το οποίο κάνει κλικ πάνω στο κουμπί ‘VACATION AND LEAVE’, το μας μεταφέρει στη λειτουργία με το ημερολόγιο διακοπών και απουσιών. Για να ελεγχθεί ότι βρισκόμαστε στη σωστή σελίδα, χρησιμοποιούμε το μπλοκ ‘Find Web Element’ για να βρούμε την κεφαλίδα ‘Company calendar’ που επιβεβαιώνει ότι έχουμε μεταφερθεί στη σωστή σελίδα. Αμέσως μετά, κάνουμε κλικ στη στήλη ‘Personal view’ με τη χρήση του μπλοκ ‘Click Web Element’ και το επιβεβαιώνουμε την μεταφορά στη νέα σελίδα, ψάχνοντας για την κεφαλίδα ‘Personal view’ με το μπλοκ ‘Find Web Element’. Έπειτα ακολουθούν δύο μπλοκ τύπου ‘Click Web Element’, το πρώτο κάνει κλικ στο κουμπί με τις τρεις τελείες και ανοίγει μια λίστα με επιλογές, ώστε το δεύτερο μπλοκ να κάνει κλικ πάνω στην επιλογή ‘Transfer vacation days’. Με αυτή την ενέργεια ανοίγει ένα βοηθητικό παράθυρο, όπως φαίνεται στο Σχήμα 3.5κ.



Σχήμα 3.5κ: Βοηθητικό παράθυρο ‘Transfer vacation days’.

Η επόμενη μας κίνηση είναι να κάνουμε κλικ πάνω στο κουμπί με το σύμβολο '+', έτσι ώστε να αυξήσουμε τις ημέρες που θέλουμε να μεταφέρουμε. Σε αυτή την περίπτωση θα έπρεπε να χρησιμοποιήσουμε τέσσερις φορές το μπλοκ 'Click Web Element' πάνω στο κουμπί '+', μία πιο έξυπνη λύση θα ήταν να χρησιμοποιήσουμε έναν βρόχο που θα επανέλαβε την ενέργεια το κλικ. Την επιλογή αυτή μας τη δίνει το μπλοκ 'Loop', το οποίο παίρνει ως 'Step', βήμα επανάληψης, την τιμή '1', ως 'Minimum' με τιμή το '1', για να ξεκινήσει η επανάληψη και ως 'Maximum' με τιμή το '4' ώστε να ολοκληρωθεί η επανάληψη σε αυτό το σημείο. Για να εκτελεστεί σωστά ο βρόγχος, θα πρέπει ο πάνω πράσινος συνδετήρας να ενώνει το μπλοκ 'Loop' με το μπλοκ 'Click Web Element', που αφορά το κουμπί '+', και για να ολοκληρωθεί ο βρόγχος και να περάσουμε στο επόμενο μπλοκ, θα πρέπει να τραβήξουμε με τον πράσινο συνδετήρα στο σημείο 'Completed', που σηματοδοτεί την λήξη του βρόχου, από το μπλοκ 'Loop' προς το πάνω μέρος του επόμενου μπλοκ. Το επόμενο μπλοκ είναι της κατηγορίας 'Click Web Element' και χρησιμοποιείται για το πάτημα του κουμπιού 'TRANSFER', που πυροδοτεί την μεταφορά πέντε (5) ημερών αδείας. Το αμέσως επόμενο μπλοκ 'Find Web Element' θα ψάξει για ένα μήνυμα επιβεβαιώσεως της μεταφοράς των ημερών. Ο χρήστης που χρησιμοποιούμε σε αυτό το σενάριο είναι διαχειριστής, γεγονός που σημαίνει ότι δεν χρειάζεται να περάσει από κάποια έγκρυση η μεταφορά των ημερών αδείας του. Ακολουθούν διαδοχικά μπλοκς 'Click Web Element' και 'Find Web Element' τα οποία περιηγούνται στα πεδία και τις κατάλληλες σελίδες της εφαρμογής όπου αναγράφονται οι μέρες που έχουν αφαιρεθεί και προστεθεί στις αντίστοιχες χρονιές, για να επιβεβαιωθεί ότι η μεταφορά ημερών έγινε επιτυχώς. Τέλος, ενώνουμε με το μπλοκ 'Pass' για τον τερματισμό της ροής ελέγχου.

Κεφάλαιο 4ο: Βιομηχανική Αξιολόγηση

Σε αυτό το κεφάλαιο, αναλαμβάνουμε το κρίσιμο βήμα της βιομηχανικής αξιολόγησης για να απαντήσουμε στο ερευνητικό ερώτημα "Η χρησιμότητα των αυτοματοποιημένων ελέγχων". Συλλέχθηκαν δεδομένα από ένα ερωτηματολόγιο που διανεμήθηκε σε συναδέλφους της εταιρείας Adminkit, από ειδικούς σε θέματα αυτοματοποιημένων ελέγχων μέχρι λιγότερο έμπειρους.

Αυτή η διερεύνηση με βάση τα δεδομένα εμβαθύνει στις πρακτικές εμπειρίες, εξετάζοντας τον τρόπο με τον οποίο ο αυτοματοποιημένος έλεγχος επηρεάζει τη διαδικασία ανάπτυξης λογισμικού. Αναλύοντας τις απαντήσεις, αποκαλύπτουμε μοτίβα που ρίχνουν φως στα απτά οφέλη των αυτοματοποιημένων ελέγχων, ανεξάρτητα από το υπόβαθρο των ερωτηθέντων. Μέσω αυτής της βιομηχανικής αξιολόγησης, προχωράμε πέρα από τη θεωρία για να κατανοήσουμε τις επιπτώσεις των αυτοματοποιημένων ελέγχων στον πραγματικό κόσμο.

4.1 Ερευνητικό Ερώτημα

Ο κύριος στόχος της βιομηχανικής αξιολόγησης γίνεται με σκοπό να βρεθεί απάντηση στο ερώτημα *"Βοηθάνε τον προγραμματιστή οι αυτοματοποιημένοι έλεγχοι;"*. Για να απαντηθεί το συγκεκριμένο ερώτημα, αρχικά θα πρέπει να αξιολογηθεί το επίπεδο γνώσεων των προγραμματιστών πάνω στον τομέα των αυτοματοποιημένων ελέγχων. Στη συνέχεια, πρέπει να αναρωτηθούμε σε ποια στάδια του κύκλου ζωής ανάπτυξης λογισμικού θα μπορούσαν οι προγραμματιστές να επωφεληθούν από τους αυτοματοποιημένους ελέγχους. Τέλος, πρέπει να αναλογιστούμε για το αν το κόστος παραγωγής και συντήρησης των αυτοματοποιημένων ελέγχων δικαιολογεί το χρόνο και το χρήμα που επενδύονται για την πρόσληψη προσωπικού και συνδρομής εργαλείων εργασίας για μια θέση απασχόλησης, όπως αυτή του μηχανικού αυτοματοποιημένων ελέγχων.

4.2 Οργάνωση Βιομηχανικής Αξιολόγησης και Συλλογή Δεδομένων

Για να απαντήσουμε στα προαναφερθέντα ερωτήματα, πραγματοποιήσαμε μια ενσωματωμένη μελέτη στη βιομηχανία λογισμικού. Η περίπτωση της μελέτης είναι μια ευρωπαϊκή εταιρεία ανάπτυξης λογισμικού με έδρα τη Νορβηγία (Σάντεφιρντ) με εξωτερικούς συνεργάτες στην Ελλάδα (Θεσσαλονίκη), και συγκεκριμένα: Adminkit. Η μελέτη είναι ενσωματωμένη, με την έννοια ότι μέσα στην ενιαία περίπτωση (εταιρεία), έχουν μελετηθεί περισσότερες από μία μονάδες ανάλυσης. Οι μονάδες ανάλυσης αντιστοιχούν στους 10 συμμετέχοντες (προγραμματιστές και μη) της μελέτης περίπτωσης.

Για τις ανάγκες της μελέτης, δημιουργήθηκε ένα ερωτηματολόγιο, τύπου Google Forms, με τίτλο *'Η χρησιμότητα του Αυτοματοποιημένου Ελέγχου'*⁵. Το ερωτηματολόγιο διανεμήθηκε στα μέλη της εταιρείας Adminkit, που αποτελούν τους βασικούς φορείς που θα παρέχουν απαντήσεις για την αξιολόγηση της χρησιμότητας των αυτοματοποιημένων ελέγχων.

Το ερωτηματολόγιο αποτελείται από τρεις ενότητες. Η πρώτη ενότητα απαρτίζεται από ένα εισαγωγικό σημείωμα, δίνοντας έναν ορισμό για τον αυτοματοποιημένο έλεγχο καθώς και μια σύντομη περιγραφή του στόχου του ερωτηματολογίου. Στη συνέχεια, οι συμμετέχοντες καλούνται να ορίσουν την ειδικότητά τους στην εταιρεία, διαλέγοντας μία από τις επιλογές: (1) DevOps, (2) Υπεύθυνος Αρχιτεκτονικής Λογισμικού, (3) Προγραμματιστής, (4) Μηχανικός Ελέγχου ή (5) 'Άλλος'.

⁵ https://docs.google.com/forms/d/e/1FAIpQLSdvvWLqMGSqfAPI97zIP95IBGQrTueEzYg5oS1wbdcsmMY-rg/viewform?usp=sf_link

Η δεύτερη ενότητα συγκροτείται από δέκα ερωτήσεις πολλαπλής επιλογής με πέντε επιλογές απαντήσεων (χρησιμοποιώντας το Likert Scale) για τους ερωτηθέντες: από το "Διαφωνώ απόλυτα" έως το "Συμφωνώ απόλυτα". Οι ερωτήσεις αυτές μπορούν να χρησιμοποιηθούν για να αξιολογήσουν τις απόψεις και τις πεποιθήσεις των συμμετεχόντων σχετικά με τους αυτοματοποιημένους ελέγχους και τη χρήση τους σε διάφορα πλαίσια εργασίας. Τα στοιχεία αξιολόγησης παρουσιάζονται παρακάτω:

Πρέπει να μάθω πολλά πράγματα πριν αρχίσω να εργάζομαι με τους αυτοματοποιημένους ελέγχους.

Αισθάνομαι πολύ σίγουρος για τη χρήση των αυτοματοποιημένων ελέγχων.

Βρήκα τους αυτοματοποιημένους ελέγχους πολύ δύσκολους στη χρήση.

Φαντάζομαι ότι οι περισσότεροι προγραμματιστές θα μάθουν να χρησιμοποιούν τους αυτοματοποιημένους ελέγχους πολύ γρήγορα.

Νόμιζα ότι υπήρχαν πάρα πολλές αλλαγές πριν και μετά τη χρήση αυτοματοποιημένων ελέγχων.

Διαπίστωσα ότι οι αυτοματοποιημένοι έλεγχοι ήταν καλά ενσωματωμένοι στη διαδικασία ανάπτυξης.

Νομίζω ότι θα χρειαστώ μεγάλη υποστήριξη για να μπορέσω να χρησιμοποιήσω αυτούς τους αυτοματοποιημένους ελέγχους.

Νομίζω ότι οι αυτοματοποιημένοι έλεγχοι θα είναι εύκολοι στη χρήση.

Βρήκα τους αυτοματοποιημένους ελέγχους περιττά πολύπλοκους, δεδομένων των πλεονεκτημάτων που παρέχουν.

Νομίζω ότι θα ήθελα να επεκτείνω τη χρήση των αυτοματοποιημένων ελέγχων σε περισσότερα μέρη του συστήματος.

Τέλος, η τρίτη ενότητα απαρτίζεται από τέσσερις ερωτήσεις ανάπτυξης. Δύο από αυτές αναφέρονται στην αναγνώριση των οφελών σχετικά με την εφαρμογή των αυτοματοποιημένων ελέγχων, όπως φαίνονται παρακάτω:

Ποια είναι τα οφέλη που έχετε διαπιστώσει κατά την εφαρμογή των αυτοματοποιημένων ελέγχων;

Πιστεύετε ότι μακροπρόθεσμα θα προκύψουν επιπλέον οφέλη (δεδομένου ότι η εταιρεία μόλις άρχισε να εφαρμόζει τους αυτοματοποιημένους ελέγχους);

Μία ερώτηση προς τους συμμετέχοντες για το αν γνωρίζουν το κόστος, είναι η ακόλουθη:

Ποιο είναι το κόστος για την εφαρμογή αυτοματοποιημένων ελέγχων;

Τέλος, τίθεται το ερώτημα εάν εξισορροπείται η αξία που προσφέρουν οι αυτοματοποιημένοι έλεγχοι έναντι του κόστους που εμπεριέχεται, αναγράφεται στο σημείο αυτό:

Πώς αντισταθμίζεται τα οφέλη έναντι του κόστους των αυτοματοποιημένων ελέγχων;

Κεφάλαιο 5ο: Αποτελέσματα και Συμπεράσματα

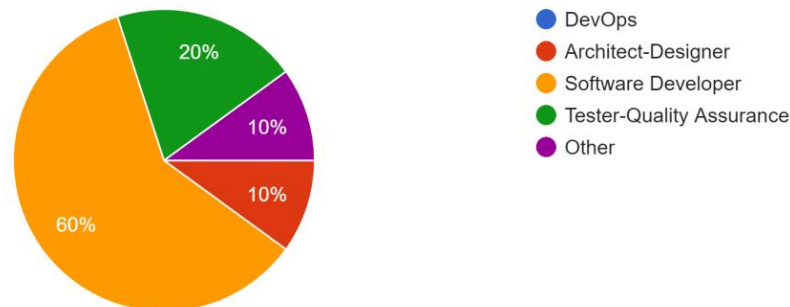
Στο πέμπτο και τελευταίο κεφάλαιο, παρουσιάζεται η συλλογή των απαντήσεων της βιομηχανικής αξιολόγησης που εξετάστηκε στο προηγούμενο κεφάλαιο, καθώς και η ανάλυση των αποτελεσμάτων της. Τέλος, αναλύονται τα συμπεράσματα της διπλωματικής εργασίας και παρουσιάζονται προτάσεις βελτίωσης της.

5.1 Αποτελέσματα

Από το ερωτηματολόγιο που διανεμήθηκε στα μέλη της εταιρείας Adminkit, λήφθηκαν δέκα απαντήσεις. Στο Σχήμα 5.1α διακρίνουμε τις απαντήσεις στην πρώτη ενότητα του ερωτηματολογίου, μέσω ενός διαγράμματος, σε μορφή πίτας, στο οποίο αναγράφεται με ποσοστά ο ρόλος όπου κατέχουν οι συμμετέχοντες στην εταιρεία που απάντησαν το ερωτηματολόγιο. Με πορτοκαλί χρώμα και ποσοστό 60%, απάντησαν έξι προγραμματιστές. Με πράσινο χρώμα και ποσοστό 20%, συμπλήρωσαν τις ερωτήσεις δύο μηχανικοί ελέγχου. Με κόκκινο χρώμα και ποσοστό 10%, λάβαμε απάντηση από έναν υπεύθυνο αρχιτεκτονικής λογισμικού. Τέλος, με μωβ χρώμα και ποσοστό 10%, απάντησε ένα μέλος της εταιρείας που κατέχει έναν διαφορετικό ρόλο στην εταιρεία από τις επιλογές που υπήρχαν. Ο διαφορετικός ρόλος του στην εταιρεία είναι αυτός του UI/UX Designer και κατέχει ελάχιστες γνώσεις προγραμματισμού.

Role in the Company

10 απαντήσεις



Σχήμα 5.1α: Ειδικότητα των συμμετεχόντων της έρευνας.

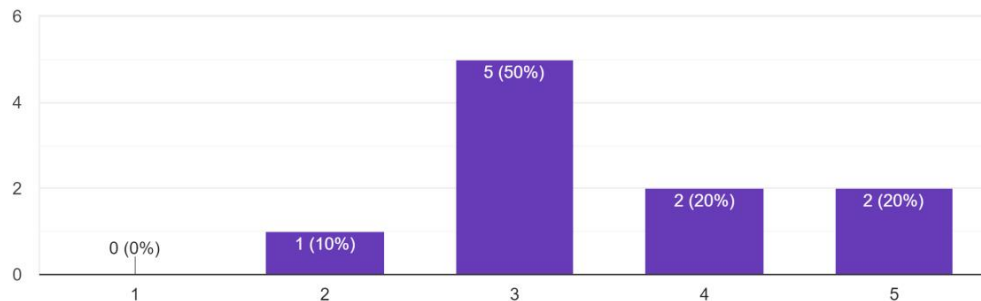
Στη δεύτερη ενότητα, στις ερωτήσεις πολλαπλής επιλογής, οι συμμετέχοντες κλήθηκαν να αξιολογήσουν τις ερωτήσεις σε μία κλίμακα από το ένα (1) έως το πέντε (5), ανάλογα με το πόσο διαφωνούν ή συμφωνούν με την πρόταση της εκάστοτε ερώτησης. Σε αυτό το σημείο αξίζει να αναφέρουμε ότι τουλάχιστον εννέα (9) άτομα έδωσαν απάντηση σε όλες τις ερωτήσεις.

Πιο συγκεκριμένα, για την 1^η ερώτηση πολλαπλής επιλογής ‘Πρέπει να μάθω πολλά πράγματα πριν αρχίσω να εργάζομαι με τους αυτοματοποιημένους ελέγχους.’ λάβαμε 10 απαντήσεις. Όπως φαίνεται στο Σχήμα 5.1β, τη πλειοψηφία των απαντήσεων μάζεψε η επιλογή 3, μαζεύοντας 5 ψήφους, καλύπτοντας το 50% των συμμετεχόντων. Οι επιλογές 4 και 5 μάζεψαν από 2 ψήφους, σημειώνοντας το ποσοστό του 20% εκάστοτε. Τέλος, η επιλογή 2 έλαβε μόλις 1 απάντηση, συμπληρώνοντας το υπόλοιπο 10% των συνολικών απαντήσεων. Από το γράφημα αυτό μπορούμε να συμπεράνουμε ότι η πλειοψηφία των συμμετεχόντων έχει μία ουδέτερη στάση ως προς την άποψη ότι πρέπει να μάθουν

πράγματα πριν ξεκινήσει να δουλεύει με τους αυτοματοποιημένους ελέγχους. Ενώ, οι υπόλοιποι τείνουν να συμφωνούν απόλυτα στο ότι θα χρειαστούν να μάθουν πολλά πράγματα πάνω στους αυτοματοποιημένους ελέγχους πριν αρχίσουν να τους εφαρμόζουν.

I need to learn a lot of things before start working with the automated tests.

10 απαντήσεις

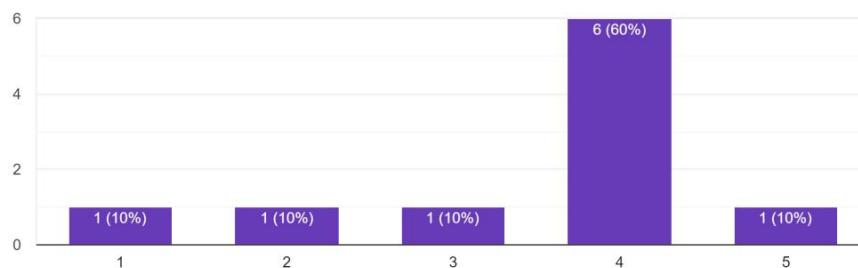


Σχήμα 5.1β: Γράφημα 1^{ης} ερώτησης πολλαπλής επιλογής.

Συνεχίζοντας με τη 2^η ερώτηση πολλαπλής επιλογής ‘Αισθάνομαι πολύ σίγουρος για τη χρήση των αυτοματοποιημένων ελέγχων.’, λήφθηκαν απαντήσεις από όλους τους συμμετέχοντες. Στο Σχήμα 5.1γ, μπορεί να διακρίνει κανείς ότι λήφθηκαν απαντήσεις σε κάθε κλίμακα επιλογών. Οι επιλογές 1, 2, 3 και 5 σημείωσαν από 1 ψήφο, καλύπτοντας το 10% η καθεμία. Η επιλογή 4 μάζεψε 6 ψήφους και διακρίθηκε ως η πιο δημοφιλής απάντηση, σημειώνοντας το υπόλοιπο 60% των απαντήσεων. Το υψηλότερο σημείο του γραφήματος είναι η επιλογή 4, με διαφορά από όλες τις υπόλοιπες απαντήσεις. Αυτό μας δείχνει πως οι συμμετέχοντες νιώθουν αρκετά σίγουροι για τη χρήση των αυτοματοποιημένων ελέγχων.

I will feel very confident using the automated tests.

10 απαντήσεις

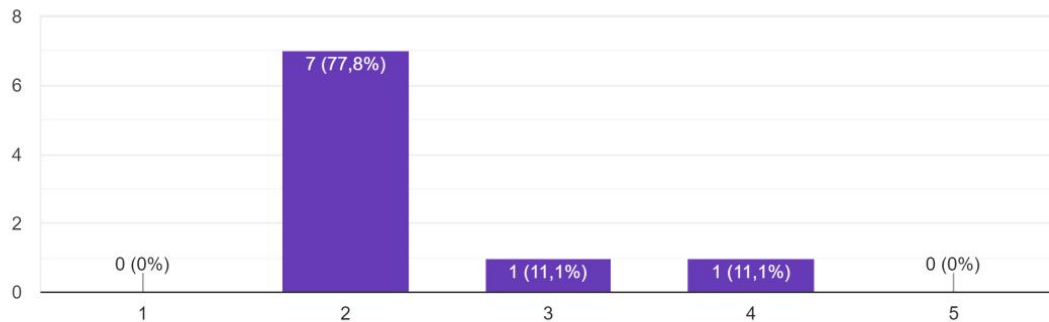


Σχήμα 5.1γ: Γράφημα 2^{ης} ερώτησης πολλαπλής επιλογής.

Το γράφημα στο Σχήμα 5.1δ απεικονίζει τις απαντήσεις της 3^{ης} ερώτησης πολλαπλής επιλογής ‘Βρήκα τους αυτοματοποιημένους ελέγχους πολύ δύσκολους στη χρήση.’, την οποία απάντησαν 9 άτομα. Μόλις 2 άτομα επέλεξαν τις επιλογές 3 και 4, καλύπτοντας αθροιστικά το 22,2% των συμμετεχόντων. Η πιο λαοφιλής απάντηση αναδείχθηκε η επιλογή 2, μαζεύοντας 7 ψήφους και σημειώνοντας το ποσοστό της τάξης του 77,8%. Για ακόμη μια ερώτηση φαίνεται πως οι απόψεις των περισσότερων ατόμων συμφωνούν και υποστηρίζουν ότι οι αυτοματοποιημένοι έλεγχοι δεν είναι δύσκολοι στη χρήση τους.

I found the automated tests very cumbersome to use.

9 απαντήσεις

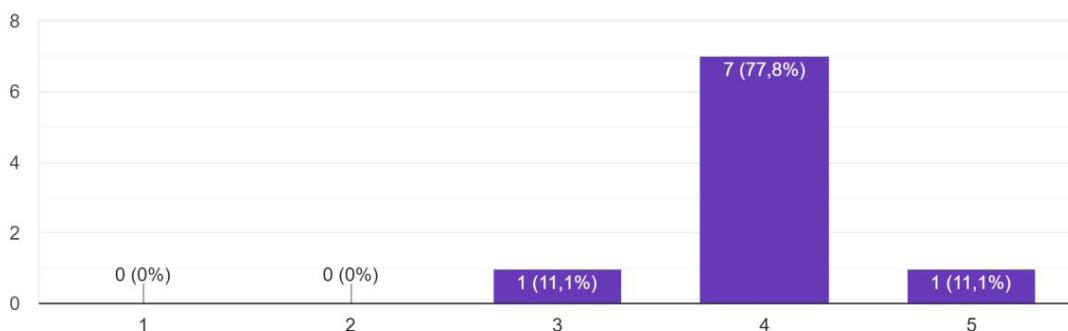


Σχήμα 5.1δ: Γράφημα 3^{ης} ερώτησης πολλαπλής επιλογής.

Στη συνέχεια, η 4^η ερώτηση πολλαπλής επιλογής ‘Φαντάζομαι ότι οι περισσότεροι προγραμματιστές θα μάθουν να χρησιμοποιούν τους αυτοματοποιημένους ελέγχους πολύ γρήγορα.’, απαντήθηκε από 9 συμμετέχοντες. Στο Σχήμα 5.1ε διακρίνουμε πως από 1 απάντηση έλαβαν οι επιλογές 3 και 5, καταλαμβάνοντας εκάστοτε το ποσοστό 11,1% των συνολικών απαντήσεων. Οι υπόλοιποι 7 συμμετέχοντες επέλεξαν την απάντηση 4, αναδεικνύοντας για άλλη μια φορά την πιο δημοφιλή επιλογή με ποσοστό 77,8%. Τα δεδομένα δείχνουν πως οι πλειοψηφία των συμμετεχόντων θεωρεί ότι οι προγραμματιστές μπορούν εύκολα να κατανοήσουν το πως σχεδιάζονται και εφαρμόζονται οι αυτοματοποιημένοι έλεγχοι.

I would imagine the most developers would learn to use the automated tests very quickly

9 απαντήσεις

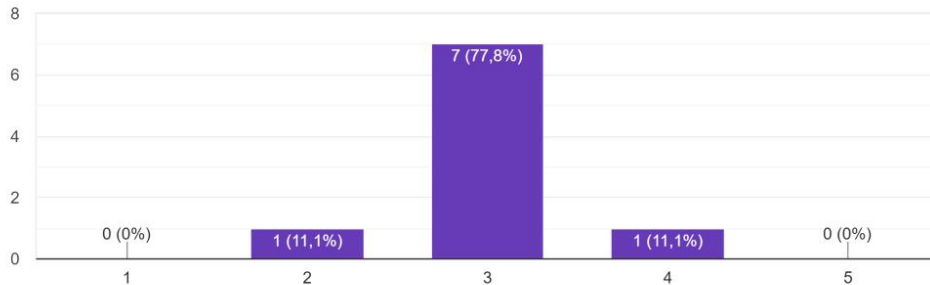


Σχήμα 5.1ε: Γράφημα 4^{ης} ερώτησης πολλαπλής επιλογής.

Περνώντας στην 5^η ερώτηση πολλαπλής επιλογής ‘Νόμιζα ότι υπήρχαν πάρα πολλές αλλαγές πριν και μετά τη χρήση αυτοματοποιημένων ελέγχων.’, λήφθηκαν απαντήσεις από 9 συμμετέχοντες. Διαβάζοντας το Σχήμα 5.1στ, η επιλογή 2 μάζεψε 1 ψήφο, διαμορφώνοντας το 11,1% των απαντήσεων. Η επιλογή 4 είχε επίσης 1ψήφο, διαμορφώνοντας και αυτή το 11,1% των απαντήσεων. Τις υπόλοιπες 7 ψήφους σύλλεξε η επιλογή 3, όπου και σημείωσε το μεγαλύτερο ποσοστό με 77,8%. Το υψηλότερο σημείο του γραφήματος είναι η επιλογή 3, κάνοντας τη διαφορά συγκριτικά με τις υπόλοιπες απαντήσεις, ταυτόχρονα όμως η επιλογή 3 βρίσκεται στη μέση της κλίμακας των επιλογών.

Θα μπορούσε κανείς να χαρακτηρίσει αρκετά ουδέτερο το αποτέλεσμα, καθώς οι συμμετέχοντες δε γνωρίζουν είτε αν υπήρξαν είτε αν δεν υπήρξαν σημαντικές αλλαγές πριν και μετά τη χρήση αυτοματοποιημένων ελέγχων.

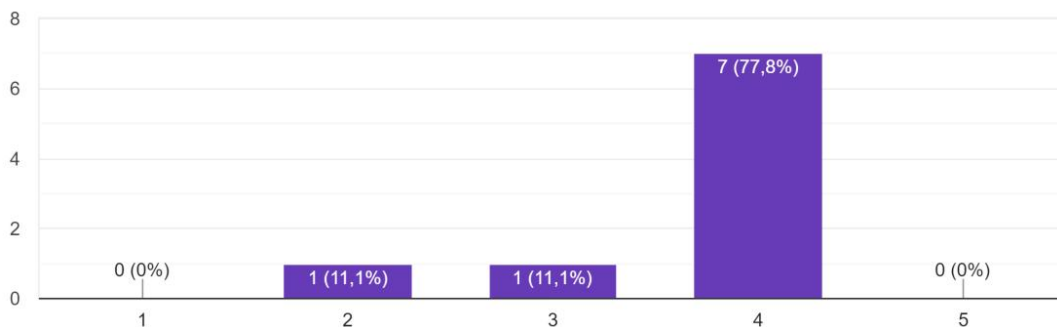
I thought there was too many changes before and after the use of automated tests.
9 απαντήσεις



Σχήμα 5.1στ: Γράφημα 5^{ης} ερώτησης πολλαπλής επιλογής.

Για την 6^η ερώτηση πολλαπλής επιλογής ‘Διαπίστωσα ότι οι αυτοματοποιημένοι έλεγχοι ήταν καλά ενσωματωμένοι στη διαδικασία ανάπτυξης.’, λάβαμε 9 απαντήσεις. Σύμφωνα με το Σχήμα 5.1ζ, μόλις 2 άτομα επέλεξαν τις επιλογές 2 και 3, καλύπτοντας αθροιστικά το 22,2% των συμμετεχόντων. Η πιο δημοφιλής απάντηση αναδείχθηκε η επιλογή 4, μαζεύοντας 7 ψήφους και σημειώνοντας το ποσοστό της τάξης του 77,8%. Για άλλη μια φορά, τα δεδομένα δείχνουν πώς η πλειοψηφία των συμμετεχόντων διαμορφώνουν κοινή άποψη ως προς την καλά ενσωματωμένη χρήση των αυτοματοποιημένων ελέγχων στη διαδικασία ανάπτυξης.

I found the automated tests were well integrated in the development process.
9 απαντήσεις



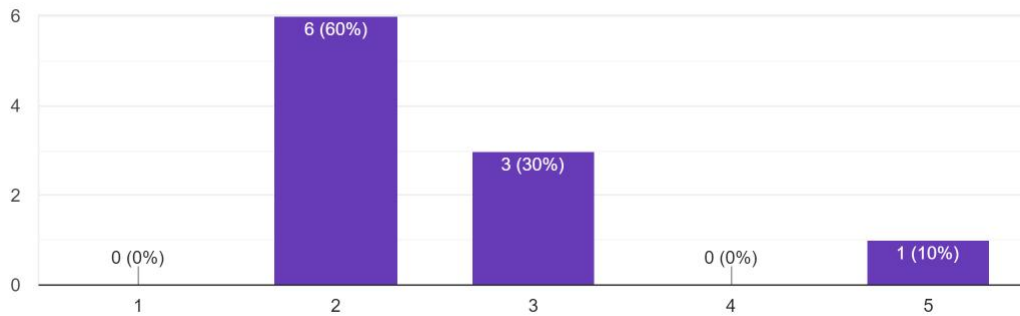
Σχήμα 5.1ζ: Γράφημα 6^{ης} ερώτησης πολλαπλής επιλογής.

Το γράφημα στο Σχήμα 5.1η απεικονίζει τις απαντήσεις της 7^{ης} ερώτησης πολλαπλής επιλογής ‘Νομίζω ότι θα χρειαστώ μεγάλη υποστήριξη για να μπορέσω να χρησιμοποιήσω αυτούς τους αυτοματοποιημένους ελέγχους.’, την οποία απάντησαν 10 συμμετέχοντες. Η επιλογή 5 έλαβε 1 ψήφο, καλύπτοντας το 10%. Η επιλογή 3 έλαβε 3 ψήφους, καλύπτοντας το 30%. Ενώ η επιλογή 2 μάζεψε τους περισσότερους ψήφους, υπ’ αριθμών 6, διαμορφώνοντας το 60% των συμμετεχόντων. Το γεγονός ότι οι περισσότεροι συμμετέχοντες δείχνουν να διαφωνούν ως προς την βοήθεια που θα

Κεφάλαιο 5

χρειαστούν για τη χρήση των αυτοματοποιημένων ελέγχων, είναι ένα θετικό δείγμα. Επίσης, η δεύτερη δημοφιλέστερη απάντηση, η 3η επιλογή, βρίσκεται στο κέντρο της κλίμακας, οπότε διατυπώνει μια αρκετά ουδέτερη στάση. Διακρίνοντας τις 9 από τις 10 απαντήσεις, ποσοστό 90%, να βρίσκονται κεντροαριστερά στην κλίμακα επιλογών, μπορούμε να καταλήξουμε στο συμπέρασμα πως οι συμμετέχοντες τείνουν να θεωρούν ότι δεν θα χρειαστούν ιδιαίτερη βοήθεια για να χρησιμοποιήσουν τους αυτοματοποιημένους ελέγχους.

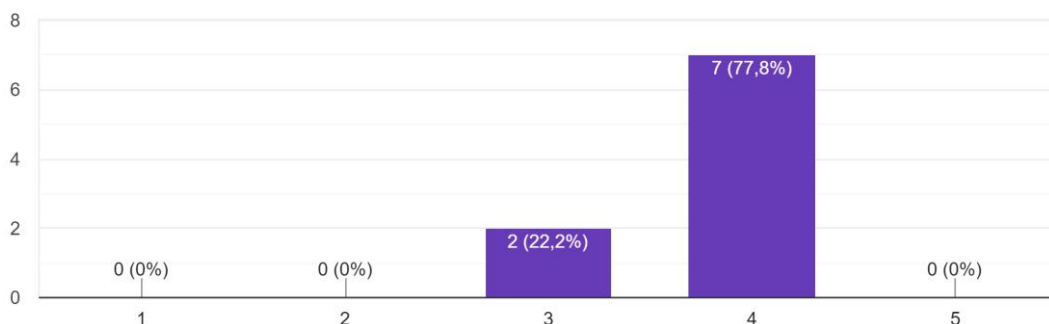
I think that I would need a lot of support to be able to use these automated tests.
10 απαντήσεις



Σχήμα 5.1η: Γράφημα 7^{ης} ερώτησης πολλαπλής επιλογής.

Στη συνέχεια, η 8^η ερώτηση πολλαπλής επιλογής ‘Νομίζω ότι οι αυτοματοποιημένοι έλεγχοι θα είναι εύκολοι στη χρήση.’, απαντήθηκε από 9 συμμετέχοντες. Στο Σχήμα 5.1θ διακρίνουμε πως έλαβαν απαντήσεις μόνο οι επιλογές 3 και 4. Η επιλογή 3 σύλλεξε 2 απαντήσεις, με ποσοστό 22,2%, ενώ η επιλογή 4 μάζεψε 7 απαντήσεις, με ποσοστό 77,8%. Αυτό το γράφημα μας επιτρέπει να έχουμε μια αρκετά καθαρή εικόνα των δεδομένων, καθώς η πλειοψηφία των συμμετεχόντων θεωρεί εύκολη την χρήση των αυτοματοποιημένων ελέγχων, ενώ κανένας δεν αντιτίθεται σε αυτή την άποψη.

I think the automated tests will be easy to use.
9 απαντήσεις



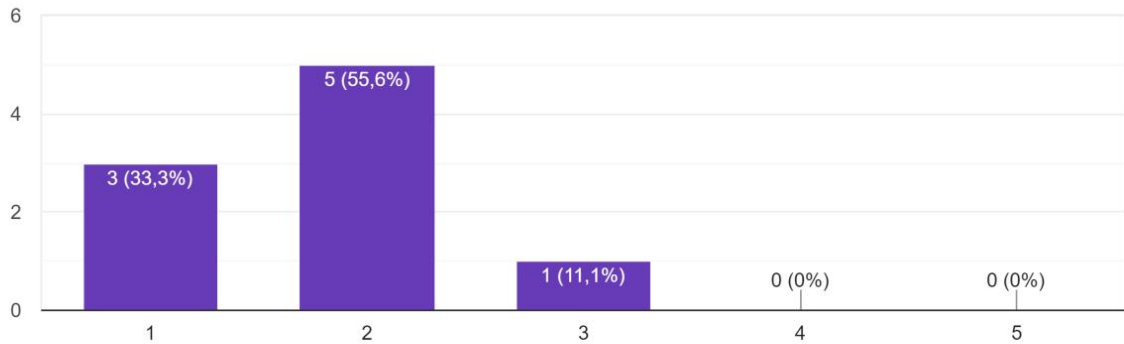
Σχήμα 5.1θ: Γράφημα 8^{ης} ερώτησης πολλαπλής επιλογής.

Για το 9^ο ερώτημα πολλαπλής επιλογής ‘Βρήκα τους αυτοματοποιημένους ελέγχους περιττά πολύπλοκους, δεδομένων των πλεονεκτημάτων που παρέχουν.’, λήφθηκαν 9 απαντήσεις. Στο Σχήμα 5.1ι φαίνεται πως η επιλογή 3 μάζεψε 1 ψήφο, με ποσοστό 11,1%. Η επιλογή 1 ψηφίστηκε από 3

άτομα, με ποσοστό 33,3%. Τέλος, η επιλογή 2 συγκέντρωσε 5 ψήφους, με ποσοστό 55,6%. Συνολικά, αυτό το γράφημα μας παρέχει μια ενδιαφέρουσα εικόνα, καθώς 8 από τις 9 απαντήσεις, ποσοστό 88,8%, τείνουν προς τα αριστερά. Το γεγονός αυτό δείχνει πως οι συμμετέχοντες δεν θεωρούν περίπλοκους τους αυτοματοποιημένους ελέγχους, δεδομένων των πλεονεκτημάτων τους.

I found the automated tests unnecessarily complex, given the benefits that they provide.

9 απαντήσεις

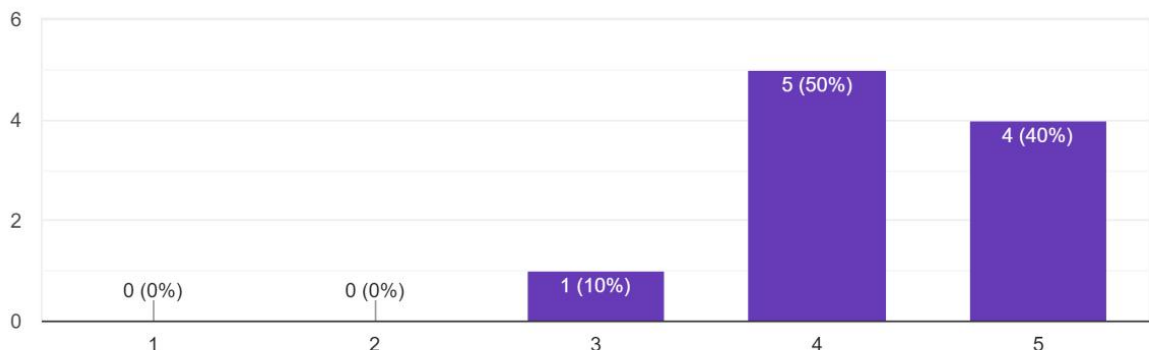


Σχήμα 5.1: Γράφημα 9^{ης} ερώτησης πολλαπλής επιλογής.

Οι απαντήσεις για την 10^η και τελευταία ερώτηση πολλαπλής επιλογής ‘Νομίζω ότι θα ήθελα να επεκτείνω τη χρήση των αυτοματοποιημένων ελέγχων σε περισσότερα μέρη του συστήματος.’, απεικονίζονται στο Σχήμα 5.1α. Τη συγκεκριμένη ερώτηση απάντησαν 10 άτομα. Η επιλογή 3 μάζεψε 1 ψήφο, με ποσοστό 10%. Η επιλογή 5 ψηφίστηκε από 4 άτομα, με ποσοστό 40%. Τέλος, η επιλογή 4 συγκέντρωσε 5 ψήφους, με ποσοστό 50%. Βλέποντας αυτό το γράφημα, μπορούμε να βγάλουμε το συμπέρασμα ότι 9 από τις 10 απαντήσεις, ποσοστό 90%, δείχνουν τάση προς τα δεξιά. Το γεγονός ότι τα δεδομένα δείχνουν αυτήν την τάση μπορεί να εξηγηθεί από τη θετική πρόθεση των συμμετεχόντων για επέκταση των αυτοματοποιημένων ελέγχων σε περισσότερα μέρη του συστήματος.

I think that I would like to expand the use of automated tests in more parts of the system.

10 απαντήσεις



Σχήμα 5.1α: Γράφημα 10^{ης} ερώτησης πολλαπλής επιλογής.

Κεφάλαιο 5

Στην τρίτη ενότητα, στις ερωτήσεις ανάπτυξης, οι συμμετέχοντες κλήθηκαν να απαντήσουν σε τέσσερις ερωτήσεις. Ο κάθε συμμετέχοντας ήταν ελεύθερος να εκφράσει την αντικειμενική του άποψη με ένα σύντομο σχόλιο πάνω στα ερωτήματα που του τέθηκαν. Σε αυτό το σημείο αξίζει να αναφέρουμε ότι τουλάχιστον οκτώ (8) άτομα έδωσαν απάντηση σε όλες τις ερωτήσεις.

Πιο συγκεκριμένα, για την 1^η ερώτηση ανάπτυξης *‘Ποια είναι τα οφέλη που έχετε διαπιστώσει κατά την εφαρμογή των αυτοματοποιημένων ελέγχων;’* λάβαμε 9 απαντήσεις. Στον Πίνακα 5.1α εμφανίζονται όλες οι απαντήσεις των συμμετεχόντων.

Πίνακας 5.1α: Απαντήσεις στην 1^η ερώτηση ανάπτυξης.

Τα σφάλματα βρέθηκαν ταχύτερα από τον έλεγχο παλινδρόμησης.
Επιδιόρθωση σφαλμάτων πριν από την έκδοση στο παραγωγικό περιβάλλον.
Λιγότερες εργατοώρες που δαπανήθηκαν για χειροκίνητους ελέγχους, με την εύρεση σφαλμάτων που ξέφυγαν από την ομάδα QA κατά τη διάρκεια των χειροκίνητων ελέγχων.
Η εύρεση σφαλμάτων και το σωστό χρονοδιάγραμμα για την εκτέλεση των ελέγχων μας είναι πραγματικά ωραίο. Μας δίνει τη δυνατότητα να τα εντοπίζουμε και να τα επιλύουμε πριν από την ανάπτυξη, αποφεύγοντας έτσι τα σφάλματα στο παραγωγικό περιβάλλον.
Νομίζω ότι το όφελος έρχεται μετά από ένα συγκεκριμένο χρονικό διάστημα, όχι αμέσως. Οι αυτοματοποιημένοι έλεγχοι φαίνονται πιο εύκολο να συντηρηθούν από τους κωδικοποιημένους ελέγχους.
Μπορούμε να τους εκτελέσουμε μια ημέρα πριν από τους χειροκίνητους ελέγχους παλινδρόμησης για να δούμε αν υπάρχει κάτι λάθος με το προϊόν.
Αισθάνομαι πιο ασφαλής να κάνω αλλαγές στον κώδικα.
Οι ενότητες (modules) συνεχίζουν να λειτουργούν μεμονωμένα ακόμη και όταν εισάγουμε νέες λειτουργίες.
Έλεγχος της βασικής λειτουργικότητας που απλά πρέπει να λειτουργεί χωρίς να ξοδεύουμε εργατοώρες κάθε φορά.

Οι απαντήσεις που λάβαμε από τους συμμετέχοντες αναδεικνύουν **σημαντικά οφέλη** από την εφαρμογή αυτοματοποιημένων ελέγχων σε μια εταιρεία ανάπτυξης λογισμικού. Αναλύοντας αυτές τις απαντήσεις συμπεράνουμε ότι είναι σημαντική η γρήγορη ανίχνευση σφαλμάτων, καθώς *συμβάλλει στη μείωση του κόστους και του χρόνου που απαιτείται για τη διόρθωσή τους*. Επίσης, η *επιδιόρθωση σφαλμάτων πριν από την έκδοση στο παραγωγικό περιβάλλον βοηθά στην πρόληψη της είσοδου σφαλμάτων στο παραγωγικό περιβάλλον, που μπορεί να έχει καταστροφικές συνέπειες*. Επιπλέον, η *μείωση του χρόνου που απαιτείται για χειροκίνητους ελέγχους μειώνει το κόστος και τον κίνδυνο ανθρώπινων λαθών*. Ακόμη, η *εύρεση σφαλμάτων και το σωστό χρονοδιάγραμμα επιτρέπει στην ομάδα ανάπτυξης να διαχειριστεί τα σφάλματα αποτελεσματικότερα και να αποφύγει την πίεση των τελευταίων στιγμών*. Η *συντήρηση των αυτοματοποιημένων ελέγχων αποτελεί σημαντικό σχόλιο*. Ενώ οι αυτοματοποιημένοι έλεγχοι μπορεί να απαιτούν αρχικά περισσότερη δουλειά, μπορούν να είναι πιο βιώσιμοι στον μακροπρόθεσμο χρόνο. Επίσης, η εκτέλεση πριν τους χειροκίνητους ελέγχους επιτρέπει να εντοπιστούν προβλήματα εκ των προτέρων, βοηθώντας στην αποφυγή πιθανών καθυστερήσεων. Επιπρόσθετα, η χρήση αυτοματοποιημένων ελέγχων μπορεί να αυξήσει την εμπιστοσύνη των προγραμματιστών στον κώδικά τους. Επίσης, η δυνατότητα να εξακολουθούν να λειτουργούν οι

ενότητες μεμονωμένα μετά από αλλαγές είναι σημαντική για τη διατήρηση της σταθερότητας του συστήματος. Τέλος, οι αυτοματοποιημένοι έλεγχοι είναι κατάλληλοι για τη διασφάλιση της βασικής λειτουργίας του προϊόντος, ενώ εξοικονομούν εργατοώρες και εξασφαλίζουν ότι οι βασικές λειτουργίες δεν θα επηρεαστούν από ανθρώπινα λάθη.

Στον Πίνακα 5.1β διακρίνουμε τις απαντήσεις για τη 2^η ερώτηση ανάπτυξης ‘*Ποιο είναι το κόστος για την εφαρμογή αυτοματοποιημένων ελέγχων;*’, για την οποία λήφθηκαν 8 απαντήσεις.

Πίνακας 5.1β: Απαντήσεις στη 2^η ερώτηση ανάπτυξης.

Χρειάζεται κάποιος χρόνος ανάπτυξης.
Πολύ λίγα, εάν όχι τίποτα.
Εξαρτάται για ποια ρύθμιση μιλάμε. Απαιτεί κάποιο συγχρονισμό, προετοιμασία και ευθυγράμμιση. Αλλά όταν έχουμε ένα νέο πακέτο για να εκτελέσουμε τους προγραμματισμένους ελέγχους μας είναι πολύ εύκολο και μια ωραία διαδικασία που μπορούμε να συμπεριλάβουμε στις συνήθειες μας.
Δεν έχω ιδέα, δεν πληρώνω.
Δεν γνωρίζω.
Ένα άτομο που εργάζεται πάνω σε αυτούς συν τη συνδρομή του προγράμματος που χρησιμοποιεί.
Μια συνδρομή (όταν χρησιμοποιήθηκε μια εμπορική λύση).
Ξέρω ότι η άδεια χρήσης ήταν υψηλή, εκτός από αυτό, δεν ξέρω.

Οι απαντήσεις σχετικά με το κόστος για την εφαρμογή αυτοματοποιημένων ελέγχων δείχνουν διάφορες προσεγγίσεις και γνώμες σχετικά με το θέμα. Πιο συγκεκριμένα, η ανάπτυξη αυτοματοποιημένων ελέγχων απαιτεί χρόνο για τη δημιουργία των σεναρίων και των ελέγχων, αυτό αποτελεί ένα αρχικό κόστος. Κάποιοι συμμετέχοντες δεν βλέπουν σημαντικό κόστος στην εφαρμογή αυτοματοποιημένων ελέγχων, ίσως επειδή εργάζονται σε ένα περιβάλλον όπου η αυτοματοποίηση είναι ευκολότερη ή έχει ήδη υλοποιηθεί. Επίσης, το κόστος εξαρτάται από την πολυπλοκότητα της ρύθμισης των ελέγχων. Πιο απλές ρυθμίσεις μπορεί να είναι πιο οικονομικές, αλλά πολύπλοκες ρυθμίσεις μπορεί να απαιτούν περισσότερη προετοιμασία και κόστος. Επιπρόσθετα, ορισμένα μέλη της εταιρείας δεν διαθέτουν πληροφορίες σχετικά με το κόστος, πιθανώς επειδή δεν είναι αρμόδιοι για το συγκεκριμένο ζήτημα. Μία ακόμη άποψη αναφέρει ότι το κόστος περιλαμβάνει την αμοιβή του ατόμου που αναπτύσσει τους ελέγχους και το κόστος της συνδρομής στο λογισμικό που χρησιμοποιείται για την ανάπτυξη αυτοματοποιημένων ελέγχων, καθώς και άλλα πιθανά έξοδα που δεν είναι γνωστά.

Για την 3^η ερώτηση ανάπτυξης ‘*Πιστεύετε ότι μακροπρόθεσμα θα προκύψουν επιπλέον οφέλη (δεδομένου ότι η εταιρεία μόλις άρχισε να εφαρμόζει τους αυτοματοποιημένους ελέγχους);*’, λήφθηκαν απαντήσεις από όλους τους συμμετέχοντες. Ο Πίνακας 5.1γ παρουσιάζει όλες τις απαντήσεις της 3^{ης} ερώτησης ανάπτυξης. Σε αυτό το σημείο, αξίζει να σημειωθεί ότι 5 από τους 10 συμμετέχοντες έδωσαν την απάντηση “Ναι”.

Πίνακας 5.1γ: Απαντήσεις στην 3^η ερώτηση ανάπτυξης.

Ναι
Σίγουρα

Όσο το έργο μεγαλώνει σε μέγεθος, τόσο πιο ωφέλιμοι θα γίνονται οι αυτοματοποιημένοι έλεγχοι.
Απολύτως ναι. Τα οφέλη μακροπρόθεσμα θα είναι σημαντικά, εφόσον οι έλεγχοι συντηρούνται καλά και εφαρμόζονται στο σχέδιό μας. Είναι πάντα ωραίο να έχουμε αυτό το αίσθημα ασφάλειας δεύτερου επιπέδου πριν από τους ελέγχους παλινδρόμησης και να βεβαιωνόμαστε ότι τα πράγματα φαίνονται καλά.
Σίγουρα βοηθάει όταν η κλιμάκωση του κώδικα θα είναι πιο επιρρεπής σε σφάλματα.
Δεδομένης της πολυπλοκότητας ενός λογισμικού, είναι σημαντικό να είστε σίγουροι ότι τα πράγματα λειτουργούν όπως αναμένεται- καθισχυράζοντας και τους πελάτες σας σχετικά με αυτό.

Οι απαντήσεις αναφέρουν συνολικά ότι οι συμμετέχοντες πιστεύουν ότι υπάρχουν επιπλέον *οφέλη* που μπορούν να προκύψουν *μακροπρόθεσμα* από την εφαρμογή των αυτοματοποιημένων ελέγχων. Για την ακρίβεια, η πολλαπλή απάντηση “Ναι” υποδηλώνει την πεποίθηση ότι θα υπάρξουν επιπλέον οφέλη στο μέλλον από την εφαρμογή των αυτοματοποιημένων ελέγχων. Όπως επίσης και η απάντηση “Σίγουρα” που είναι ακόμη πιο θετική και υποδηλώνει μεγαλύτερη βεβαιότητα ότι θα υπάρξουν επιπλέον οφέλη. Επίσης, η κλιμάκωση του κώδικα επισημαίνει ότι με την αύξηση της πολυπλοκότητας του λογισμικού, οι αυτοματοποιημένοι έλεγχοι θα γίνουν πιο χρήσιμοι. Αυτό είναι ιδιαίτερα σημαντικό σε περιβάλλοντα όπου το λογισμικό αναπτύσσεται σε μεγάλη κλίμακα. Πέρα από αυτό, οι αυτοματοποιημένοι έλεγχοι προσφέρουν ένα αίσθημα ασφάλειας δεύτερου επιπέδου πριν από τους ελέγχους παλινδρόμησης. Αυτό μπορεί να συμβάλει στην πρόληψη προβλημάτων σε παραγωγικό επίπεδο. Ακόμη, οι αυτοματοποιημένοι έλεγχοι βοηθούν κατά την κλιμάκωση του κώδικα, όπου οι πιθανότητες σφαλμάτων αυξάνονται. Τέλος, υπογραμμίζεται η σημασία των αυτοματοποιημένων ελέγχων στην διασφάλιση της λειτουργικότητας του λογισμικού, ιδίως σε περιβάλλοντα με πολύπλοκο κώδικα.

Οι απαντήσεις για την 4^η και τελευταία ερώτηση ανάπτυξης ‘Πώς αντισταθμίζεται τα οφέλη έναντι του κόστους των αυτοματοποιημένων ελέγχων;’, παρουσιάζονται στον Πίνακα 5.1δ και διατυπώθηκαν 9 απαντήσεις.

Πίνακας 5.1δ: Απαντήσεις στην 4^η ερώτηση ανάπτυξης.

Οφέλη > βάρη, μειώνει πολύ το χρόνο ελέγχων.
Οι αυτοματοποιημένοι έλεγχοι ενισχύουν την εμπιστοσύνη για ένα προϊόν, παρόλο που το αρχικό κόστος μπορεί να είναι υψηλό.
Από τις εργατοώρες που εξοικονομούνται με τη χρήση αυτοματοποιημένων ελέγχων και την ακρίβεια των ελέγχων.
Τα οφέλη υπερβαίνουν το κόστος όταν μιλάμε για ένα στάδιο ανάπτυξης. Όσο γρηγορότερα ανακαλύπτουμε τα προβλήματα και όσο μεγαλύτερη ανάπτυξη επιτυγχάνεται με την εφαρμογή αυτοματοποιημένων ελέγχων, τόσο πιο ασφαλής αισθάνεστε. Όταν πρόκειται για το κόστος, όπως είναι το κόστος και τα εργαλεία που χρησιμοποιούμε, αυτό είναι ένα άλλο πράγμα για το οποίο δεν μπορώ να προβληματιστώ πάρα πολύ.
Δεν μπορώ να απαντήσω σε αυτό.
Νομίζω ότι είναι ενδιαφέρον και αν γίνεις καλός σε αυτό, μπορεί να είναι πολύ ωφέλιμο.
Το να έχεις ένα άτομο να εκτελεί και να κατασκευάζει αυτοματοποιημένους ελέγχους, είναι ένα κόστος, αλλά εξοικονομεί χρόνο από τους χειροκίνητους ελέγχους, και ακόμη πιο σημαντικό είναι ότι

αποτρέπει λάθη και σφάλματα που θα μπορούσαν να προκαλέσουν διαφόρων ειδών ζημιά στην εταιρεία.
Τα πλεονεκτήματα της χρήσης ενός σετ ελέγχων είναι πολύ πιο σημαντικά από οποιοδήποτε σχετικό κόστος ή την πρόσληψη νέων υπαλλήλων.
Νομίζω ότι τα οφέλη είναι μεγαλύτερα από το κόστος μακροπρόθεσμα.

Οι απαντήσεις δείχνουν ότι οι συμμετέχοντες πιστεύουν ότι τα οφέλη της χρήσης αυτοματοποιημένων ελέγχων υπερνικούν το κόστος της υλοποίησής τους. Αναλυτικότερα, υπογραμμίζεται ότι τα οφέλη είναι πολύ μεγαλύτερα από το κόστος, καθώς οι αυτοματοποιημένοι έλεγχοι μειώνουν σημαντικά τον χρόνο που απαιτείται για τους ελέγχους. Επίσης, το αρχικό κόστος μπορεί να είναι υψηλό, αλλά οι αυτοματοποιημένοι έλεγχοι ενισχύουν την εμπιστοσύνη στο προϊόν, προσθέτοντας αξία. Ακόμη, αναφέρεται ότι τα οφέλη προέρχονται από την εξοικονόμηση εργατοωρών και την ακρίβεια των ελέγχων, δείχνοντας ότι οι αυτοματοποιημένοι έλεγχοι προσφέρουν αποτελεσματική λύση. Επιπρόσθετα, τα οφέλη υπερβαίνουν το κόστος, ειδικά στα αρχικά στάδια ανάπτυξης, καθώς η πρόωγη ανίχνευση προβλημάτων εξοικονομεί χρόνο και αποφεύγει πιθανές ζημίες. Παρ' όλα αυτά, δεν έλειψε και η απάντηση ενός συμμετέχοντα ο οποίος δεν έχει συγκεκριμένη γνώση για το πώς αντισταθμίζονται τα οφέλη έναντι του κόστους. Μία ακόμη άποψη είναι ότι το κόστος συμπεριλαμβάνει την αμοιβή του ατόμου που ασχολείται με τους αυτοματοποιημένους ελέγχους. Παρόλα αυτά, επισημαίνει ότι αυτό το κόστος επιστρέφει ως εξοικονομημένος χρόνος από τους χειροκίνητους ελέγχους και προλαμβάνει σφάλματα που θα μπορούσαν να προκαλέσουν ζημίες στην εταιρεία. Τέλος, γίνεται διατυπώνεται ότι τα πλεονεκτήματα της χρήσης αυτοματοποιημένων ελέγχων είναι πολύ σημαντικότερα από το συνολικό κόστος ή την πρόσληψη νέων υπαλλήλων.

5.2 Συμπεράσματα

Τα συμπεράσματα της διπλωματικής εργασίας αναδεικνύουν σαφώς τα *οφέλη* και τα *κόστη* που σχετίζονται με τη χρήση αυτοματοποιημένων ελέγχων σε μια εταιρεία ανάπτυξης λογισμικού. Οι απαντήσεις που λάβαμε από τους συμμετέχοντες παρέχουν μια εμπειριστατωμένη εικόνα της εμπειρίας τους και των απόψεών τους σχετικά με το θέμα της χρησιμότητας των αυτοματοποιημένων ελέγχων.

Πρώτον, όσον αφορά τα οφέλη, παρατηρήσαμε ότι οι αυτοματοποιημένοι έλεγχοι επιτρέπουν την ταχύτερη ανίχνευση σφαλμάτων, επιτρέποντας την επιδιόρθωσή τους πριν από την έκδοση στο παραγωγικό περιβάλλον. Αυτό οδηγεί σε λιγότερες ώρες που δαπανώνται για χειροκίνητους ελέγχους, καθώς και στην ασφαλή ανάπτυξη του προϊόντος.

Δεύτερον, οι αυτοματοποιημένοι έλεγχοι βοηθούν στον έλεγχο της βασικής λειτουργικότητας χωρίς να απαιτείται χειροκίνητη επέμβαση, ενώ διατηρούν τη λειτουργικότητα των ενοτήτων ακόμη και κατά την εισαγωγή νέων λειτουργιών.

Τρίτον, όσον αφορά το κόστος, οι απαντήσεις στη βιομηχανική αξιολόγηση που πραγματοποιήσαμε υποδεικνύουν ότι το κόστος είναι περίπλοκο να εκτιμηθεί, καθώς εξαρτάται από πολλούς παράγοντες, όπως τον χρόνο ανάπτυξης και τη συνδρομή σε εργαλεία. Ωστόσο, οι απαντήσεις επιβεβαιώνουν ότι τα οφέλη αντισταθμίζουν το κόστος, ειδικά στα αρχικά στάδια ανάπτυξης.

Τέλος, αναφορικά με τη μακροπρόθεσμη προοπτική, οι απαντήσεις είναι θετικές. Οι συμμετέχοντες της αξιολόγησης πιστεύουν ότι με την αύξηση του μεγέθους του έργου, τα οφέλη από τη χρήση αυτοματοποιημένων ελέγχων θα αυξηθούν περαιτέρω, προσθέτοντας αξία στην εταιρεία και ενισχύοντας την εμπιστοσύνη στο προϊόν.

Συνολικά, οι απαντήσεις επιβεβαιώνουν τη σημαντική συνεισφορά των αυτοματοποιημένων ελέγχων στη διαδικασία ανάπτυξης λογισμικού. Το κόστος ανάπτυξης και συντήρησης αυτών των ελέγχων είναι αναμφίβολα παράγοντες που πρέπει να ληφθούν υπόψη, αλλά τα οφέλη είναι πολύ ισχυρότερα και πιο αναδειγμένα. Συνοψίζοντας, τα βασικά συμπεράσματα που προκύπτουν από τις απαντήσεις των συμμετεχόντων είναι τα εξής:

- Οι αυτοματοποιημένοι έλεγχοι συμβάλλουν σημαντικά στην ταχύτερη ανίχνευση και αποκατάσταση σφαλμάτων, μειώνοντας έτσι τον χρόνο που αφιερώνεται σε χειροκίνητους ελέγχους.
- Οι αυτοματοποιημένοι έλεγχοι ενισχύουν την εμπιστοσύνη στο προϊόν, προσφέροντας πρόληψη από δυνητικές ζημιές και ακριβούς ελέγχους.
- Το κόστος ανάπτυξης και συντήρησης αυτοματοποιημένων ελέγχων είναι σύνθετο, αλλά τα οφέλη αντισταθμίζουν αυτό το κόστος.
- Με την αύξηση του μεγέθους του έργου, αναμένεται να αυξηθούν περαιτέρω τα οφέλη των αυτοματοποιημένων ελέγχων.

Εξ ολοκλήρου, οι αυτοματοποιημένοι έλεγχοι αποτελούν μια αποτελεσματική πρακτική για τη βελτίωση της διαδικασίας ανάπτυξης λογισμικού και την εξοικονόμηση χρόνου και πόρων, καθώς και για την ενίσχυση της ποιότητας και της εμπιστοσύνης στο προϊόν. Εν κατακλείδι, η χρήση αυτοματοποιημένων ελέγχων είναι μια αναπτυξιακή πρακτική που αξίζει να υιοθετηθεί και να ενσωματωθεί στις διαδικασίες ανάπτυξης λογισμικού μιας εταιρείας ανάπτυξης λογισμικού για τη βελτίωση της ποιότητας και την αποφυγή δυνητικών προβλημάτων.

5.3 Μελλοντική Έρευνα

Σε αυτήν την ενότητα παρέχεται μια επισκόπηση των αποτελεσμάτων και των συμπερασμάτων που προκύπτουν από τις απαντήσεις των συμμετεχόντων στο ερωτηματολόγιο. Ωστόσο, επίσης παρέχει μια ενδελεχή ανάλυση των τάσεων και των προοπτικών για το μέλλον σχετικά με τη χρήση αυτοματοποιημένων ελέγχων στη διαδικασία ανάπτυξης λογισμικού.

Ένα ενδιαφέρον σχόλιο που μπορεί να προστεθεί σε αυτό το κεφάλαιο είναι η αναφορά σε ενδεχόμενες μελλοντικές εξελίξεις και τάσεις που ενδέχεται να επηρεάσουν τη χρήση αυτοματοποιημένων ελέγχων. Για παράδειγμα, η εξέλιξη της τεχνητής νοημοσύνης και της μηχανικής μάθησης μπορεί να παρέχει νέες δυνατότητες για την αυτόματη ανίχνευση σφαλμάτων και την πρόβλεψη προβλημάτων στο λογισμικό. Επίσης, η ανάλυση των δεδομένων και η εκμετάλλευση των μετρήσεων από τους αυτοματοποιημένους ελέγχους μπορεί να εξελιχθεί περαιτέρω για να παρέχει πληροφορίες για τη βελτίωση της απόδοσης του λογισμικού.

Επιπλέον, θα ήταν ενδιαφέρον να εξεταστεί η πιθανή επίδραση της αυξανόμενης πολυπλοκότητας των συστημάτων λογισμικού στην ανάγκη για αυτοματοποιημένους ελέγχους και πώς αυτό μπορεί να επηρεάσει τις προτιμήσεις και τις ανάγκες των αναπτυσσόμενων ομάδων λογισμικού.

Περαιτέρω, μια διπλωματική εργασία που εξετάζει την ανάπτυξη, τη λειτουργία και τη συντήρηση ενός πιο ολοκληρωμένου προϊόντος που χρησιμοποιεί αυτοματοποιημένους ελέγχους για μεγάλο χρονικό διάστημα θα ήταν μια ενδιαφέρουσα προοπτική για μελλοντική έρευνα. Συγκριτικά με ένα προϊόν το οποίο ακόμα βρίσκεται υπό ανάπτυξη και πρόσφατα ξεκίνησε να εφαρμόζει αυτοματοποιημένους ελέγχους, θα μπορούσε να εξερευνηθεί η σημασία των αυτοματοποιημένων

ελέγχων στη βιομηχανία του λογισμικού. Ακόμη θα μπορούσε να πραγματοποιηθεί μία έρευνα που θα παρακολουθεί την συχνότητα με την οποία αναγνωρίζουν σφάλματα οι αυτοματοποιημένοι έλεγχοι.

Τέλος, θα μπορούσαν να γίνουν μελέτες πάνω στη σημασία της συνεχούς εκπαίδευσης και εξοικείωσης των ανθρώπων με τη χρήση αυτοματοποιημένων ελέγχων, καθώς η τεχνολογία εξελίσσεται και οι απαιτήσεις αλλάζουν.

Βιβλιογραφία

- [1] ISTQB Certified Tester - Foundation Level Syllabus v4.0, https://istqb-main-web-prod.s3.amazonaws.com/media/documents/ISTQB_CTFL_Syllabus-v4.0.pdf
- [2] <https://www.javatpoint.com/software-development-life-cycle>
- [3] <https://www.toolsqa.com/software-testing/istqb/why-is-testing-necessary/>
- [4] <https://innoroo.com/blog/2018/11/22/application-under-test-glossary/>
- [5] <https://www.javatpoint.com/types-of-software-testing>
- [6] <https://www.javatpoint.com/manual-testing>
- [7] <https://www.javatpoint.com/white-box-testing>
- [8] <https://www.javatpoint.com/black-box-testing>
- [9] <https://www.javatpoint.com/grey-box-testing>
- [10] <https://www.javatpoint.com/functional-testing>
- [11] <https://www.browserstack.com/guide/functional-testing>
- [12] <https://www.guru99.com/non-functional-testing.html>
- [13] <https://www.javatpoint.com/non-functional-testing>
- [14] <https://www.globalapptesting.com/blog/what-is-automation-testing>
- [15] <https://www.javatpoint.com/automation-testing>
- [16] <https://www.javatpoint.com/automation-testing-vs-manual-testing>
- [17] <https://www.javatpoint.com/advantages-of-automation-testing>
- [18] <https://intellipaat.com/blog/what-is-automation-testing/?US>
- [19] <https://www.selenium.dev/documentation/>
- [20] <https://appium.io/docs/en/2.0/>
- [21] <https://docs.katalon.com/docs>
- [22] <https://cucumber.io/docs/cucumber/>
- [23] <https://docs.cypress.io/guides/overview/why-cypress>
- [24] Kong, P., Li, L., Gao, J., Liu, K., Bissyandé, T. F., & Klein, J. (2018). Automated testing of android apps: A systematic literature review. *IEEE Transactions on Reliability*, 68(1), 45-66.
- [25] Rafi, D. M., Moses, K. R. K., Petersen, K., & Mäntylä, M. V. (2012, June). Benefits and limitations of automated software testing: Systematic literature review and practitioner survey. In *2012 7th International Workshop on Automation of Software Test (AST)* (pp. 36-42). IEEE.
- [26] MACAKOĞLU, Ş. S., & Peker, S. (2022, June). Web accessibility performance analysis using web content accessibility guidelines and automated tools: a systematic literature review. In *2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)* (pp. 1-8). IEEE.

[27] Méndez, A., Quesada, L. C. U., & Jenkins, C. M. (2015). Automated testing of mobile applications: A systematic map and review. In Proceedings of the 10th International Conference on Web Information Systems and Technologies.

[28] <https://www.consilium.europa.eu/en/policies/data-protection/data-protection-regulation/>

[29] <https://www.leapwork.com/product/documentation/architecture>

[30] <https://www.leapwork.com/product/documentation/building-blocks>

[31] <https://www.leapwork.com/product/documentation/designing-automation>

[32] <https://www.leapwork.com/product/documentation/run-lists>

[33] <https://www.leapwork.com/product/documentation/reporting>