



ΑΛΕΞΑΝΔΡΕΙΟ Τ.Ε.Ι. ΘΕΣΣΑΛΟΝΙΚΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Πλατφόρμες και Εργαλεία Ελέγχου στις Web και Desktop Εφαρμογές

Του φοιτητή
καθηγητής

Τσιμπιρίδη Κυριάκου

Αρ. Μητρώου: 06/3070

Επιβλέπων

Σφέτσος Παναγιώτης

Θεσσαλονίκη 2019

ΠΕΡΙΛΗΨΗ

Τόσο το διαδίκτυο όσο και η χρήση των προσωπικών υπολογιστών είναι σήμερα ισχυρά μέσα για την ικανοποίηση των αναγκών των χρηστών τόσο στη πρόσβαση πληροφοριών και υπηρεσιών μέσω του Διαδικτύου όσο και στη χρήση προγραμμάτων στους Η/Υ αντίστοιχα. Και στις δύο περιπτώσεις ο έλεγχος των προγραμμάτων και εφαρμογών πριν την διάθεση τους προς χρήση παρουσιάζει πολυπλοκότητα λόγω της δυναμικής που επιφέρουν οι σύγχρονες εφαρμογές. Σε αυτή την πτυχιακή εργασία παρουσιάζουμε μία συστηματική ανασκόπηση της βιβλιογραφίας αναλύοντας μια σειρά άρθρων που δημοσιεύθηκαν από το 2000 και ύστερα. Εστιάζουμε σε δύο τομείς ελέγχου εφαρμογών: διαδίκτυο (web) και Η/Υ (desktop). Ακολουθώντας ένα συγκεκριμένο πρωτόκολλο επιλογής, εξάγουμε ένα συγκεκριμένο σύνολο εγγράφων που χρησιμοποιείται ως βάση για να απαντήσουμε σε τέσσερα ερευνητικά ερωτήματα.

Τα αποτελέσματα της έρευνας μας ερμηνεύουν τις τάσεις σχετικά με τα παραπάνω ερωτήματα. Στο πρώτο ερώτημα διαπιστώνεται ότι μόλις 48 από τα 125 έγγραφα (38%) παρουσίαζαν κάποια εργαλεία για έλεγχο εφαρμογών. Και στους δύο τομείς υπάρχουν μελέτες που εστιάζουν σε τεχνικές αντιστροφής κώδικα σε κάποιο μοντέλο που να προσεγγίζει τον αρχικό σχεδιασμό (33%) για την αναγνώριση και ταξινόμηση σφαλμάτων. Στο δεύτερο ερώτημα, περισσότερο αναδιπλώνεται μία ευρύτερη γκάμα εξειδικευμένων τεχνικών ελέγχου σε εφαρμογές Η/Υ (λειτουργιών, διεπαφών, αντοχής, απόδοσης) παρά σε διαδικτυακές εφαρμογές. Στο τρίτο ερώτημα διαπιστώνεται ότι εμπειρικές μελέτες έχουν πραγματοποιηθεί και στους δύο τομείς ωστόσο για διαφορετικούς σκοπούς. Όπως και στο τέταρτο ερευνητικό ερώτημα διαπιστώνεται ότι λίγες είναι αυτές οι έρευνες που εστιάζουν στην περιγραφή των τεχνολογικών εξελίξεων σχετικά με τον έλεγχο λογισμικού. Παρατηρείται επίσης σημαντική απόκλιση μεταξύ των στόχων στους δύο τομείς (πχ εργαλεία, διαδικασίες και μελλοντικές τεχνολογικές προκλήσεις στις web εφαρμογές έναντι δημιουργία περιπτώσεων ελέγχου στις desktop εφαρμογές).

Διαπιστώνεται επίσης ότι οι δοκιμές στο διαδίκτυο κυριαρχούν στην έρευνα σε αντίθεση με τις δοκιμές σε εφαρμογές για Η/Υ. Τα αποτελέσματα παρέχουν επίσης χρήσιμα μαθήματα για τα διαφορετικά μοντέλα στη πραγματοποίηση δοκιμών σε προγράμματα λογισμικού, τα μοντέλα σφαλμάτων ή ταξινόμησης σφαλμάτων για εφαρμογές στο διαδίκτυο. Παρέχουν επίσης συμπεράσματα σχετικά με τα εργαλεία πραγματοποίησης δοκιμών σε εφαρμογές στο διαδίκτυο καθώς και εργαλεία δοκιμών για εφαρμογές Η/Υ. Αντλούνται επίσης χρήσιμα μαθήματα για τις δυνατότητές τους, τις διαδικασίες και μεθοδολογίες για την εκτέλεση εφαρμογών, μετρήσεις που χρησιμοποιούνται για την αξιολόγηση του κόστους και των χρονικών περιορισμών και στους δύο τύπους δοκιμών, στην εξέλιξη της τεχνολογίας και στις μελλοντικές τάσεις και στους δύο τύπους.

ABSTRACT

Both the web and the desktop are powerful channels nowadays to satisfy user needs over the Internet and the office/home work respectively. In both areas complexity occurs due to the complexity and dynamism modern applications enforce. In this systematic literature review work we present a range of papers published since 2000 in both fields of testing. Based on our selection protocol a specific set of papers are extracted seeking to provide answer to four research questions.

The results of our research interpret the trends in the above questions. The first question found that only 48 of the 125 documents (38%) had any testing tools. In both areas there are studies focusing on reverse engineering techniques (from code or script to a model) that reverses approximately to the original design (33%) for error identification and classification. In the second question, there is a wider range of specialized testing techniques in desktop (functions, interfaces, durability, and performance) rather than in web applications. The third question states that empirical studies have been carried out in both areas however for different purposes. As in the fourth research question, there are few studies focusing on describing state-of-the-art technology related to software testing. There is also a significant discrepancy between objectives in the two areas (e.g. tools, processes and future technological challenges in web applications versus creating test cases in desktop applications).

Interestingly we conclude that web testing dominates research in contrast to desktop testing. Results also provide useful lessons for the types of test models, the fault models/bug taxonomy for the web applications, test tools for web application testing as well as test tools for desktop application testing and their capabilities, procedures and methodologies for performing desktop applications, metrics used for evaluating cost and time constraints in both types of testing, state of the art and future trends in both types.

ΠΕΡΙΕΧΟΜΕΝ

ΠΕΡΙΛΗΨΗ.....	2
ABSTRACT.....	3
ΠΕΡΙΕΧΟΜΕΝΑ.....	4
Ευρετήριο σχημάτων.....	5
Ευρετήριο πινάκων.....	8
ΕΙΣΑΓΩΓΗ.....	6
ΚΕΦΑΛΑΙΟ 1.....	8
ΕΙΣΑΓΩΓΗ.....	8
1.1 ΔΕΥΤΕΡΕΥΟΥΣΕΣ ΜΕΛΕΤΕΣ ΣΕ ΔΟΚΙΜΕΣ ΛΟΓΙΣΜΙΚΟΥ Η/Υ.....	8
1.2 ΔΕΥΤΕΡΕΥΟΥΣΕΣ ΜΕΛΕΤΕΣ ΣΕ ΔΟΚΙΜΕΣ ΕΦΑΡΜΟΓΩΝ ΔΙΑΔΙΚΤΥΟΥ.....	10
ΚΕΦΑΛΑΙΟ 2.....	12
ΕΙΣΑΓΩΓΗ.....	12
2.1 ΠΡΩΤΟΚΟΛΛΟ ΑΝΑΣΚΟΠΗΣΗΣ.....	12
2.2 ΕΡΕΥΝΗΤΙΚΕΣ ΕΡΩΤΗΣΕΙΣ.....	13
2.3 ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΠΗΓΕΣ ΚΑΙ ΚΡΙΤΗΡΙΑ ΑΝΑΖΗΤΗΣΗΣ.....	14
2.3.1 ΑΡΧΙΚΗ ΔΙΑΔΙΚΑΣΙΑ ΑΞΙΟΛΟΓΗΣΗΣ.....	15
2.3.2 ΚΡΙΤΗΡΙΑ ΣΥΜΠΕΡΙΛΗΨΗΣ.....	16
2.3.3 ΚΡΙΤΗΡΙΑ ΑΠΟΚΛΕΙΣΜΟΥ.....	16
2.4 ΕΠΙΛΟΓΗ ΑΡΧΙΚΩΝ ΜΕΛΕΤΩΝ.....	16
2.5 ΕΞΑΓΩΓΗ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΣΥΝΘΕΣΗ ΔΕΔΟΜΕΝΩΝ.....	18
2.6 ΑΠΟΤΕΛΕΣΜΑΤΑ.....	18
ΚΕΦΑΛΑΙΟ 3 ΣΥΜΠΕΡΑΣΜΑΤΑ.....	29
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	31
ΑΝΑΦΟΡΕΣ.....	33

Ευρετήριο σχημάτων

Σχήμα 1. Προτεινόμενο πρωτόκολλο ανασκόπησης.....	13
Σχήμα 2. Ομαδοποίηση πρωτοβάθμιων μελετών ανά έτος.....	17

Ευρετήριο πινάκων

Πίνακας 1 " Δευτεροβάθμιες μελέτες στον έλεγχο λογισμικού ".....	09
Πίνακας 2 " Δευτεροβάθμιες μελέτες στον έλεγχο εφαρμογών λογισμικού ".....	11
Πίνακας 3. Κατάλογος των εργαλείων δοκιμών εφαρμογών web που είναι διαθέσιμα για λήψη.....	19
Πίνακας 4. Μοντέλο σφάλματος / ταξινόμηση σφαλμάτων - ερευνητική εργασία. .	20
Πίνακας 5. Κατάλογος των εργαλείων δοκιμής εφαρμογών Η/Υ διαθέσιμων για λήψη.....	22
Πίνακας 6. Κατάλογος τεχνικών ελέγχου εφαρμογών για Η/Υ.....	25
Πίνακας 7. Εμπειρικές μελέτες σχετικά με τη δοκιμή διαδικτυακών εφαρμογών ιστού (στη πλευρά του τελικού χρήστη).....	25
Πίνακας 8. Εμπειρικές μελέτες σχετικά με τις δοκιμές web εφαρμογών (πλευρά εξυπηρετητή).....	26
Πίνακας 9. Εμπειρικές μελέτες σχετικά με τον έλεγχο εφαρμογών Η/Υ.....	27
Πίνακας 10. Έλεγχος εφαρμογών στο διαδίκτυο και για Η/Υ – τρέχουσα τεχνολογία.....	28

ΕΙΣΑΓΩΓΗ

Ο έλεγχος λογισμικού είναι μία διαδικασία που έχει σχεδιαστεί για τον προσδιορισμό της ποιότητας, της απόδοσης και της αξιοπιστίας μιας εφαρμογής ή ενός συστήματος λογισμικού πριν χρησιμοποιηθεί από τους τελικούς χρήστες.

Σήμερα, η τεχνολογία της πληροφορικής είναι αυτή που έχει αναπτυχθεί περισσότερο από οποιαδήποτε άλλη επιστήμη. Οι χρήστες έχουν αποκτήσει όλο και περισσότερο εμπειρία στην εφαρμογή της τεχνολογίας ώστε να αναζητούν συνεχώς λογισμικά υψηλότερης ποιότητας. Επίσης έχουν την εμπειρία και την γνώση να αξιολογήσουν τα διαθέσιμα προγράμματα λογισμικού με βάση τις ανάγκες τους. Για παράδειγμα, η αυτοματοποίηση του ελέγχου αποτελεί μείζονα στόχο για τις εταιρείες που ακολουθούν μοντέλα με στενούς κύκλους ανάπτυξης λογισμικού. Αυτό συμβαίνει επειδή είναι δυνατό να βελτιωθεί η ποιότητα του λογισμικού που αναπτύσσουν με δυναμικό τρόπο.

Πολλοί ερευνητές στο χώρο της τεχνολογίας λογισμικού εστιάζουν στη βελτίωση των διαδικασιών ελέγχου των εφαρμογών που χρησιμοποιούνται είτε στους προσωπικούς υπολογιστές ή στο διαδίκτυο (δηλαδή είναι προσβάσιμες μέσω ενός browser). Για παράδειγμα, ένα ερευνητικό ερώτημα που τίθεται είναι αν οι διαδικτυακές εφαρμογές παρουσιάζουν περισσότερα, λιγότερα, ή εξίσου τα ίδια σφάλματα στο επίπεδο παρουσίασης, επιχειρησιακής λογικής και στο επίπεδο δεδομένων από ότι οι παραδοσιακές εφαρμογές.

Στο παραπάνω λαμβάνεται υπόψη ότι και οι διαδικτυακές εφαρμογές χειρίζονται πλέον ευαίσθητα δεδομένα όπως και οι παραδοσιακές εφαρμογές στους Η/Υ και πολλοί άνθρωποι στηρίζονται σε αυτές για να υποστηρίξουν διάφορες καθημερινές δραστηριότητες. Σε κάθε περίπτωση τα όποια σφάλματα ενδέχεται να έχουν σοβαρές και ευρείες συνέπειες. Σε αντίθεση με τις περισσότερες εφαρμογές Η/Υ (desktop), πολλές web εφαρμογές γράφονται σε scripting γλώσσες, όπως η PHP. Τα δυναμικά χαρακτηριστικά που συνήθως υποστηρίζονται από αυτές τις γλώσσες αναστέλλουν σημαντικά τη στατική ανάλυση και η υπάρχουσα στατική ανάλυση αυτών των γλωσσών μπορεί να αποτύχει να παράγει σημαντικά αποτελέσματα.

Και ενώ στις desktop εφαρμογές εφαρμόζεται ένα πλαίσιο ελέγχων βασισμένο στη συμβολική εκτέλεση για την εύρεση σφαλμάτων, τη κάλυψη όσο το δυνατόν περισσότερο μέρους του κώδικα (πχ γραμμένο σε γλώσσα C, Java κοκ) ή ακόμη και την αυτοματοποιημένη δημιουργία περιπτώσεων ελέγχου, στη περίπτωση των web εφαρμογών οι scripting γλώσσες, όπως η PHP, προωθούν ένα τρόπο προγραμματισμού για την ανάπτυξη εφαρμογών ιστού που δίνει έμφαση στις τιμές των συμβολοσειρών, τα αντικείμενα και τους πίνακες.

Αυτή η πτυχιακή εργασία θα ασχοληθεί με τον έλεγχο λογισμικό τόσο σε εφαρμογές που χρησιμοποιούνται σε προσωπικούς Η/Υ όσο και εφαρμογές που είναι προσβάσιμες στο διαδίκτυο. Θα ασχοληθεί επίσης με τον τρόπο με τον οποίο ο έλεγχος μπορεί να γίνει πιο αποτελεσματικός, γρήγορος και αποδοτικός.

Το κεφάλαιο 1 παρουσιάζει μία σύντομη ανασκόπηση της βιβλιογραφίας και ερευνητικής δραστηριότητας σε δύο τομείς ελέγχου: εφαρμογές στο διαδίκτυο, και εφαρμογές για Η/Υ. Το κεφάλαιο 2 επιχειρεί βαθύτερη ανάλυση παρουσιάζοντας μία συστηματική μέθοδο εξέτασης της βιβλιογραφίας: γίνεται συλλογή και ταξινόμηση σχετικών επιστημονικών άρθρων σύμφωνα με συγκεκριμένο πρωτόκολλο ανασκόπησης, ερευνητικών ερωτήσεων, κριτηρίων αναζήτησης και επιλογής των τελικών αποτελεσμάτων. Το κεφάλαιο 3 ολοκληρώνει την πτυχιακή εργασία.

ΚΕΦΑΛΑΙΟ 1

ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΑΝΑΣΚΟΠΗΣΗ

ΕΙΣΑΓΩΓΗ

Η βιβλιογραφική ανασκόπηση ερευνητικών δραστηριοτήτων που σχετίζονται με το πλαίσιο της εργασίας ξεδιπλώνεται σε τρεις κατηγορίες:

1. δευτεροβάθμιες μελέτες στον τομέα του ελέγχου λογισμικού
2. ψηφιακά αποθετήρια με επιστημονικές εργασίες σχετικά με τον έλεγχο λογισμικού ή γενικότερα τη τεχνολογία λογισμικού και
3. δευτεροβάθμιες μελέτες που εστιάζουν στον έλεγχο εφαρμογών στο διαδίκτυο και τους Η/Υ.

1.1 ΔΕΥΤΕΡΕΥΟΥΣΕΣ ΜΕΛΕΤΕΣ ΣΕ ΔΟΚΙΜΕΣ ΛΟΓΙΣΜΙΚΟΥ Η/Υ

Ανακτήσαμε 18 δευτεροβάθμιες μελέτες ((Garousi V. et al, 2013); (Ferreira de Souza E. et al, 2015); (Garousi V et al, 2015); (Banerjee I. et al, 2013); (Neto et al, 2011); (Palacios M. et al, 2011); (Garousi V. and Mantyla V., 2016); (Per Lauvås jr and Andrea Arcuri, 2017); (Afzal W. et al, 2009); (André Takeshi et al, 2010); (Lee J. et al, 2012); (Prabhakar N. et al, 2018); (Jia Y. and Harman M., 2003); (Mesbah A. and Prasad M., 2011); (Bozkurt M. et al, 2010); (Memon A.M., Nguyen B.N., 2010); (Jamil M. et al, 2016); (Garousi V. and Mantyla V., 2016)) σε διάφορους τομείς δοκιμών λογισμικού (με έμφαση στις εφαρμογές Η/Υ).

Αυτές οι μελέτες παρουσιάζονται στον Πίνακα 1 σε συνδυασμό με τον αντίστοιχο τομέα της εξειδίκευσής τους. Είναι σαφές ότι το παραδοσιακό πεδίο των ερευνών, της ανασκόπησης βιβλιογραφίας και της ταξινόμησης έχει ξεπεραστεί από την αυξανόμενη ζήτηση για συστηματική χαρτογράφηση (6 μελέτες) και τη συστηματική βιβλιογραφική ανασκόπηση (4 μελέτες).

Το ερευνητικό έργο του V. Garousi κυριαρχεί τουλάχιστον σε τρεις τύπους δευτεροβάθμιας μελέτης (SM, SLR, Literature Review). Ως εκ τούτου, το βάθος της εργασίας του παρείχε μια πολύτιμη αναφορά και σε αυτή την πτυχιακή εργασία που επίσης εστιάζει στην πραγματοποίηση μίας συστηματικής ανασκόπησης τύπου SLR. Πράγματι μία σημαντική ταξινόμηση εγγράφων και εμπειρικών μελετών εντοπίστηκαν στο έργο του και μας επέτρεψε να αναλύσουμε τις περαιτέρω πτυχές του ελέγχου τόσο στις εφαρμογές Η/Υ όσο και στις web εφαρμογές.

Πίνακας 1 " Δευτεροβάθμιες μελέτες στον έλεγχο λογισμικού "

Τύπος δευτεροβάθμιας μελέτης	Περιοχή δευτεροβάθμιας μελέτης	Αριθμός πρωτογενών μελετών	Έτος	Αναφορά
SM	Web application testing	79	2013	(Garousi V. et al, 2013)
	Knowledge management initiatives	40	2015	(Ferreira de Souza E. et al, 2015)
	Code engineering	125	2015	(Garousi V et al, 2015)
	Graphical User Interface (GUI) testing	136	2013	(Banerjee I. et al, 2013)
	Product lines testing	45	2011	(Neto et al, 2011)
	Service Oriented Architecture Testing	33	2011	(Palacios M. et al, 2011)
SLR	literature reviews in software testing	29	2016	(Garousi V. and Mantyla V., 2016)
	Trends in Software Testing Education	32	2017	(Per Lauvås jr and Andrea Arcuri, 2017)
	Search-based non-functional testing	35	2009	(Afzal W. et al, 2009)
	Formal testing of web services	37	2010	(André Takeshi et al, 2010)
Survey/analysis	software testing practices	32	2012	(Lee J. et al, 2012)
	Meta-heuristic Techniques in Software Testing	28	2018	(Prabhakar N. et al, 2018)
	Mutation testing	264	2011	(Jia Y. and Harman M., 2011)
	Service Oriented Architectures Testing	64	2011	(Mesbah A. and Prasad M., 2011)
	Testing web services	86	2010	(Bozkurt M. et al, 2010)
Taxonomy	Model-based GUI testing	33	2010	(Memon A.M., Nguyen B.N., 2010)
Literature review	Software testing techniques	28	2016	(Jamil M. et al, 2016)
	Software testing automation	78	2016	(Garousi V. and Mantyla V., 2016)

1.2 ΔΕΥΤΕΡΕΥΟΥΣΕΣ ΜΕΛΕΤΕΣ ΣΕ ΔΟΚΙΜΕΣ ΕΦΑΡΜΟΓΩΝ ΔΙΑΔΙΚΤΥΟΥ

Αυτές οι μελέτες (Πίνακας 2) συνοψίζουν τρεις κύριους τύπους δευτερογενούς μελέτης (συστηματική χαρτογράφηση, συστηματική ανασκόπηση βιβλιογραφίας και μελέτες). Οι (Garousi V. et al, 2013) επιδιώκουν να παρουσιάσουν μια γενική εικόνα των τάσεων και των εμπειρικών αποδείξεων στις δοκιμές web εφαρμογών. Οι (Alal fi M.H., Cordy, J.R., Dean, T.R., 2009) επικεντρώνονται στις τεχνικές μοντελοποίησης για να καταγράψουν διαφορετικές συμπεριφορές στο σύστημα, να υποστηρίξουν την επαλήθευση και τη δοκιμή μίας διαδικτυακής εφαρμογής. Η συστηματική βιβλιογραφική ανασκόπηση του (Sabaren L.N et al, 2017) αναλύει την ερευνητική προσπάθεια σχετικά με τα εργαλεία και τις τεχνικές ελέγχου σε θέματα συμβατότητας.

Αυτή η ερευνητική δραστηριότητα σχετίζεται με πρακτικές αυτοματοποίησης δοκιμών λειτουργικότητας και διεπαφής χρήστη. Οι (Lucca, G.A.D., Fasolino, A.R., 2006) θέτουν τα πιο κρίσιμα ανοικτά ζητήματα σε εφαρμογές διαδικτύου με ανεπτυγμένη λειτουργικότητα (π.χ. ενσωματώνουν λειτουργικά χαρακτηριστικά της τεχνολογίας Ajax) και βοηθούν στη συλλογή σχετικών εργασιών για την κατασκευή ενός πλαισίου ταξινόμησης της υλοποίησης περιπτώσεων δοκιμών με αυτοματοποιημένο τρόπο. Οι (Amalfitano, D., Fasolino, A., Tramontana, P., 2010b) πραγματοποίησαν έρευνα που εστίασε στα εργαλεία, και τις τεχνικές που υποστηρίζουν την εκτέλεση ελέγχου σε διαδικτυακές εφαρμογές.

Με εξαίρεση το έργο του (Garousi V. et al, 2013), το υπόλοιπο ερευνητικό έργο (δευτερογενείς μελέτες) έχει αρκετούς περιορισμούς όσον αφορά την αναπαραγωγή, γενίκευση και χρηστικότητα. Ιδιαίτερα οι μελέτες (surveys) διεξήχθησαν με ένα πρόχειρο (adhoc) τρόπο και από τις οποίες έλειπε μια συστηματική προσέγγιση ανασκόπησης βιβλιογραφίας. Ως εκ τούτου, τα κριτήρια επιλογής δεν περιγράφονται ρητά για να συγκρίνουν τα αποτελέσματά τους με άλλες προσεγγίσεις. Επίσης, αυτές οι μελέτες θα έπρεπε να περιλαμβάνουν ένα ευρύτερο αριθμό εγγράφων για να γίνει περισσότερο κατανοητό το πεδίο και η προοπτική τους.

Πίνακας 2 " Δευτεροβάθμιες μελέτες στον έλεγχο εφαρμογών λογισμικού "

Τύπος δευτεροβάθμιας μελέτης	Περιοχή δευτεροβάθμιας μελέτης	Αριθμός πρωτογενών μελετών	Έτος	Αναφορά
SM	Web application testing	193	2013	(Garousi V. et al, 2013)
	Modeling methods for web application verification and testing: state of the art	24	2009	(Alalfi M.H., Cordy, J.R., Dean, T.R., 2009)
SLR	Cross-browser Testing	56	2017	(Sabaren L.N et al, 2017)

	Techniques and tools for rich internet applications testing	29	2010	(Lucca, G.A.D., Fasolino, A.R., 2006)
Survey	Web software testing techniques	25	2011	(Santa S., Travassos G.H., 2011)
	A survey of web applications testing	20	2009	(Kam, B., Dean, T.R., 2009)
	Testing web-based applications: the state of the art and future trends	27	2010	(Amalfitano, D., Fasolino, A., Tramontana, P., 2010b)

ΚΕΦΑΛΑΙΟ 2

ΣΥΣΤΗΜΑΤΙΚΗ ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΑΝΑΣΚΟΠΗΣΗ

ΕΙΣΑΓΩΓΗ

Η συστηματική βιβλιογραφική ανασκόπηση (Systematic Literature Review – SLR) γίνεται βάσει μεθόδου που προτείνουμε σε αυτή την εργασία. Η μέθοδος μας επικεντρώνεται σε συγκεκριμένους στόχους και ερευνητικά ζητήματα και εστιάζει στα παρακάτω:

1. επιλογή των κατάλληλων άρθρων σχετικά με τους στόχους μας
2. αξιολόγηση της τελικής λίστας άρθρων και ψηφιακών αποθετηρίων
3. εξαγωγή δεδομένων
4. σύνθεση δεδομένων

2.1 ΠΡΩΤΟΚΟΛΛΟ ΑΝΑΣΚΟΠΗΣΗΣ

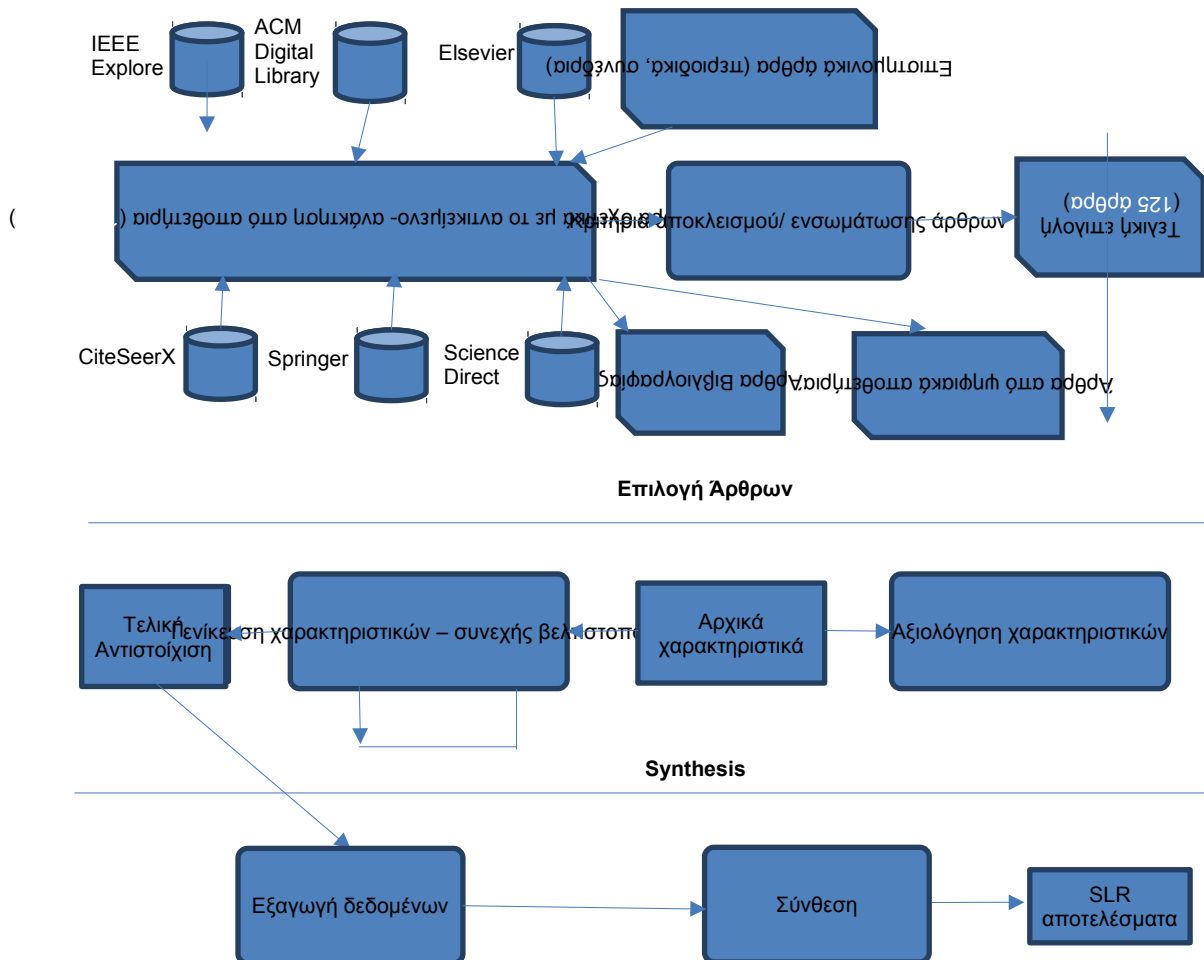
Οι (Kitchenham, B., Charters, S., 2007) προτείνουν ως κύριο κίνητρο για τη πραγματοποίηση μίας συστηματικής ανασκόπησης (SLR) τον προσδιορισμό των σωστών διαδικασιών, τεχνολογιών, μεθόδων και εργαλείων που σχετίζονται με ένα θέμα. Αυτό επιτυγχάνεται συγκεντρώνοντας πληροφορίες από πολλαπλές πηγές και συγκρίνοντας τις προτάσεις τους και τα αποτελέσματα της έρευνάς τους. Τα βήματα για την επίτευξη μίας βέλτιστης πρακτικής για αυτό το σκοπό περιλαμβάνουν τα εξής:

- Σχεδιασμός της μεθόδου ανασκόπησης:
 - Προσδιορισμός της ανάγκης ανασκόπησης
 - Καθορισμός των ερευνητικών ερωτήσεων
 - Ανάπτυξη πρωτοκόλλου ανασκόπησης
 - Αξιολόγηση του πρωτοκόλλου ανασκόπησης
- Διεξαγωγή της επανεξέτασης:
 - Προσδιορισμός του αντικειμένου της ανασκόπησης
 - Επιλογή πρωτογενών μελετών
 - Αξιολόγηση ποιότητας μελέτης
 - Εξόρυξη και παρακολούθηση δεδομένων
 - Σύνθεση δεδομένων

Αφού προσδιοριστεί η ανάγκη για την ανασκόπηση, αναπτύσσουμε ένα πρωτόκολλο μελέτης και καθορίζουμε τις ερευνητικές ερωτήσεις. Μετά τη διαδικασία σχεδιασμού, η διαδικασία καθοδηγεί την επιλογή των άρθρων και την διαμόρφωση μιας προσέγγισης ταξινόμησης. Αυτό αποτελεί τη βάση για να βελτιωθούν περαιτέρω τα κριτήρια αναζήτησης και να απαντηθούν οι στόχοι και οι ερευνητικές ερωτήσεις της συστηματικής έρευνας. Επίσης βοηθά σε ένα επόμενο

στάδιο να αντιστοιχηθούν τα επιστημονικά άρθρα στις ερευνητικές ερωτήσεις και να εξαχθούν χρήσιμα δεδομένα σχετικά με τα συστηματικά αποτελέσματα της αντιστοίχισης των εγγράφων.

Το προτεινόμενο πρωτόκολλο ανασκόπησης απεικονίζεται παρακάτω (Σχήμα 1).



Σχήμα 1. Προτεινόμενο πρωτόκολλο ανασκόπησης

2.2 ΕΡΕΥΝΗΤΙΚΕΣ ΕΡΩΤΗΣΕΙΣ

Στόχος της παρούσας εργασίας είναι η διεξαγωγή μίας εκτεταμένης συστηματικής ανασκόπησης της βιβλιογραφίας προκειμένου να απαντηθούν τα ακόλουθα ζητήματα:

1. Ποια είναι τα εργαλεία, οι διαδικασίες και οι μεθοδολογίες ελέγχου εφαρμογών στο διαδίκτυο που χρησιμοποιούνται σήμερα;

Αυτό θα βοηθήσει στον εντοπισμό των βέλτιστων πρακτικών σχετικά με τις μεθοδολογίες και τα μοντέλα δοκιμών.

2. Ποια είναι τα εργαλεία, οι διαδικασίες και οι μεθοδολογίες που χρησιμοποιούνται σήμερα στον έλεγχο εφαρμογών Η/Υ;
Αυτό θα βοηθήσει στον εντοπισμό βέλτιστων πρακτικών σχετικά με τις μεθοδολογίες και τα μοντέλα ελέγχου λογισμικού.
3. Πώς φημισμένες εμπειρικές μελέτες χειρίζονται το θέμα του ελέγχου – στο επίπεδο του κώδικα - σε εφαρμογές web και Η/Υ αντίστοιχα;
Στην περίπτωση αυτή, εξετάζουμε τις μετρήσεις που βοηθούν στην εξαγωγή δεικτών σημαντικών χαρακτηριστικών, συμπεριλαμβανομένου του κόστους και της αποτελεσματικότητας. Εξετάζονται επίσης διάφορες πτυχές σχετικές με το σχεδιασμό αυτών των μελετών και τις πηγές των στοιχείων τους και τον τρόπο σύγκρισης μεταξύ τους.
4. Ποια είναι η τρέχουσα κατάσταση της τεχνολογίας σχετικά με τον έλεγχο web εφαρμογών/ιστοσελίδων και εφαρμογών Η/Υ;
Αυτό θα βοηθήσει στην αναγνώριση της τρέχουσας κατάστασης των εργαλείων εκτέλεσης δοκιμών σε εφαρμογές διαδικτύου και Η/Υ (desktop).

2.3 ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΠΗΓΕΣ ΚΑΙ ΚΡΙΤΗΡΙΑ ΑΝΑΖΗΤΗΣΗΣ

Βάσει των οδηγιών σε διάφορες μεθοδολογίες τύπου SLR για την εξαγωγή πρωτογενών μελετών και βιβλιογραφικής ανασκόπησης σχετικών με το θέμα μας, θα χρησιμοποιήσουμε ψηφιακές βιβλιοθήκες καθώς και επιστημονικά περιοδικά που είναι διαθέσιμα στο διαδίκτυο. Ο κατάλογος ψηφιακών βιβλιοθηκών, ηλεκτρονικών περιοδικών και ακαδημαϊκών μηχανών αναζήτησης για επιστημονικά άρθρα περιλαμβάνει τα παρακάτω:

- Information and Software Technology (IST)
- Journal of Systems and Software (JSS)
- IEEE Transactions on Software Engineering (TSE)
- IEEE Software (IEEE SW)
- Communications of the ACM (CACM)
- ACM Computer Surveys (ACM Sur)
- ACM Transactions on Software Engineering Methodologies (TOSEM)
- Software Practice and Experience (SPE)
- Empirical Software Engineering Journal (EMSE)

- IET Software (IET SW)
- Proceedings International Conference on Software Engineering (ICSE)
- Proceedings International Symposium of Software Metrics (Metrics)
- Proceedings International Symposium on Empirical Software Engineering (ISESE)

Επίσης, αντλούνται άρθρα από τις παρακάτω μηχανές αναζήτησης:

1. ACM Digital Library (<http://portal.acm.org>)
2. IEEEExplore (<http://ieeexplore.ieee.org>)
3. CiteseerX Library (<http://citeseerx.ist.psu.edu>)
4. ScienceDirect (<http://www.sciencedirect.com>)
5. WileyInterScience (<http://www3.interscience.wiley.com>)
6. SpringerLink (<http://www.springerlink.com>)

Όροι αναζήτησης:

Για να μειώσουμε τον κίνδυνο έλλειψης σχετικών μελετών, παρόμοιες με τις προηγούμενες μελέτες SLR, αναζητήσαμε επίσης σχετικές πηγές χρησιμοποιώντας λέξεις-κλειδιά αναζήτησης:

“Web applications testing tools OR desktop applications testing tools OR web code testing OR desktop code testing OR web testing platforms OR desktop testing platforms OR web testing empirical studies OR desktop testing empirical studies”

Επιπλέον, εξετάσαμε βιβλιογραφικές αναφορές που βρέθηκαν σε μελέτες που υπάρχουν ήδη στην ομάδα καθώς και προσωπικές ιστοσελίδες ενεργών ερευνητών στο θέμα. Εξάγοντας τα ονόματα των ενεργών ερευνητών από το αρχικό σύνολο άρθρων τα χρησιμοποιήσαμε για αναζήτηση στις παραπάνω μηχανές αναζήτησης. Όλα τα άρθρα που βρέθηκαν στις πρόσθετες πηγές που δεν ήταν ακόμη στην ομάδα των επιλεγμένων έργων αλλά φαίνονταν σχετικές με το αντικείμενο προστέθηκαν στην αρχική ομάδα.

2.3.1 ΑΡΧΙΚΗ ΔΙΑΔΙΚΑΣΙΑ ΑΞΙΟΛΟΓΗΣΗΣ

Με τα παραπάνω κριτήρια (φράσεις –κλειδιά) αναζήτησης σε συγκεκριμένες πηγές βρήκαμε 146 μελέτες που θεωρήσαμε ως την αρχική δεξαμενή των

σχετικών εργασιών. Σε ένα επόμενο βήμα, εφαρμόσαμε ένα σύνολο κριτηρίων συμπερίληψης / αποκλεισμού για να αποκλείσουμε μη σχετικές μελέτες.

Τα κριτήρια ένταξης / αποκλεισμού εφαρμόστηκαν σε όλη τη έρευνα μας. Για να αυξήσουμε την αξιοπιστία της μελέτης μας και των αποτελεσμάτων της, εφαρμόσαμε μια διαδικασία συστηματικής βαθμολόγησης στη φάση επιλογής άρθρων για ανασκόπηση με βάση μία διαδικασία βαθμολόγησης. Σύμφωνα με τα αποτελέσματα μπορούσαμε να αποφασίσουμε είτε να συμπεριλάβουμε είτε να αποκλείσουμε οποιαδήποτε από τα άρθρα για την επόμενη έκδοση της δεξαμενής των άρθρων μας. Αυτή η διαδικασία χρησιμοποιήθηκε επίσης για να αποφευχθεί οποιαδήποτε προσωπική προκατάληψη ως προς συγκεκριμένη εργασία. Αν υπήρχε οποιαδήποτε σύγκρουση λόγω της ίδιας βαθμολογίας, αυτό επιλύθηκε με περαιτέρω ανάγνωση των συγκεκριμένων εγγράφων.

2.3.2 ΚΡΙΤΗΡΙΑ ΣΥΜΠΕΡΙΛΗΨΗΣ

Τα κριτήρια ένταξης περιλάμβαναν τρία ερωτήματα:

1. Είναι η ερευνητική εργασία σχετική με τον έλεγχο λειτουργικών χαρακτηριστικών σε εφαρμογές διαδικτύου ή σχετική με τον έλεγχο λειτουργικών χαρακτηριστικών σε εφαρμογές Η/Υ;
2. Περιέχει το άρθρο ένα σχετικά αποδεκτό τρόπο αξιολόγησης της προτεινόμενης μεθοδολογίας ή μοντέλου ελέγχου;
3. Περιέχει το άρθρο μία παρουσίαση τεχνικών, εργαλείων ή μελετών περιπτώσεων / πειραμάτων;

2.3.3 ΚΡΙΤΗΡΙΑ ΑΠΟΚΛΕΙΣΜΟΥ

Τα δεδομένα αποκλεισμού περιλάμβαναν δύο ερωτήματα:

1. Έχει δημοσιευθεί το άρθρο σε κάποιο επιστημονικό περιοδικό ή συνέδριο;
2. Το άρθρο υστερεί στην ανάλυση και παρουσίαση εμπειρικών δεδομένων ή άλλων δεδομένων που θα καθιστούσαν την όποια προτεινόμενη μεθοδολογία ή μοντέλο ελέγχου κοινά αποδεκτό; Το άρθρο παρουσιάζει μόνο την άποψη των συγγραφέων;

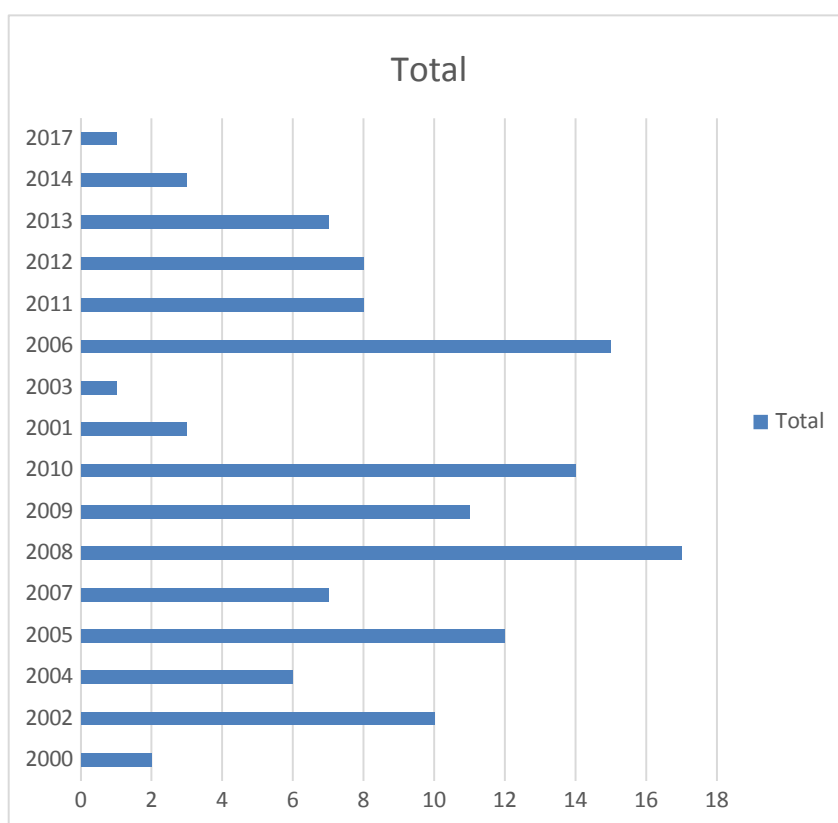
2.4 ΕΠΙΛΟΓΗ ΑΡΧΙΚΩΝ ΜΕΛΕΤΩΝ

Χρησιμοποιώντας την παραπάνω διαδικασία προσπαθήσαμε να εκτελέσουμε μια αρχική αξιολόγηση ανεξάρτητα για κάθε άρθρο. Μόνο όταν ένα δεδομένο άρθρο αντιστοιχούσε στα κριτήρια συμπερίληψης και δεν ταίριαζε με κανένα από τα κριτήρια αποκλεισμού, συμπεριλήφθηκε στην επόμενη έκδοση της δεξαμενής άρθρων. Διαφορετικά, δεν συμπεριλήφθηκε.

Εξετάσαμε κυρίως έγγραφα με βάση τον τίτλο, την περίληψη, τις λέξεις-κλειδιά και τις ενότητες αποτελεσμάτων / συμπερασμάτων. Αν δεν μπορούσαμε να αντλήσουμε επαρκείς πληροφορίες από την περίληψη, θα εξετάζαμε επίσης το περιεχόμενο για να επιβεβαιώσουμε εάν το συγκεκριμένο ερευνητικό άρθρο κάλυψε πράγματι ένα θέμα σχετικό με το αντικείμενο της έρευνας μας.

Μετά την αρχική αξιολόγηση και την επανεξέταση της λίστας για τη συμπερίληψη σχετικών άρθρων και την εξαίρεση των μη σχετικών εργασιών, η λίστα των επιλεγμένων πρωτοβάθμιων μελετών περιλάμβανε πλέον 125 άρθρα/μελέτες. Στο παράρτημα των "Αναφορών" καταγράφουμε μια πλήρη λίστα όλων των πρωτογενών μελετών. Η κατηγοριοποίηση κάθε επιλεγμένης επιστημονικής δημοσίευσης είναι σύμφωνα με το σχήμα ταξινόμησης που παρουσιάζεται στο (Garousi V. et al, 2013).

Το σχήμα 2 αναλύει το χρονοδιάγραμμα των ανακτώμενων δημοσιεύσεων. Αυτό ξεκινάει από το 2000 έως το 2017, αλλά η αύξηση των δημοσιεύσεων παρουσιάζεται από το 2008 έως το 2010 και μετά το ενδιαφέρον για το αντικείμενο αρχίζει να μειώνεται. Παρόλο που υπήρξε μια αύξηση του ενδιαφέροντος κάθε δύο έως τρία χρόνια (π.χ. 2002 έως 2005, 2005 έως 2008, 2008 έως 2010), αυτό δεν συνέχισε μετά το 2010. Η περιοδική αύξηση μπορεί να σχετίζεται με τις αλλαγές στις τεχνολογίες και τις τεχνικές ανάπτυξης λογισμικού, βελτίωση της ποιότητας του λογισμικού (όσον αφορά τις εφαρμογές διαδικτύου και Η/Υ).



Σχήμα 2. Ομαδοποίηση πρωτοβάθμιων μελετών ανά έτος

Ένας άλλος λόγος που δικαιολογεί τη μείωση των δημοσιεύσεων σχετικές με τον έλεγχο λογισμικού στους δύο τομείς, σχετίζεται με την επίτευξη υψηλότερου βαθμού αυτοματοποιημένων ελέγχων. Μετά το 2010 η ανάπτυξη λογισμικού γίνεται πιο τυποποιημένη και ο έλεγχος κώδικα επιτυγχάνεται με τη χρήση ειδικών εργαλείων/προτύπων. Το γεγονός αυτό διευκολύνει την αυτοματοποιημένη δοκιμή μέσω ειδικών εργαλείων και προτύπων.

2.5 ΕΞΑΓΩΓΗ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΣΥΝΘΕΣΗ ΔΕΔΟΜΕΝΩΝ

Έχοντας ως σημείο αναφοράς τις 125 πρωτοβάθμιες και δευτεροβάθμιες μελέτες/άρθρα προχωρήσαμε στη δημιουργία μιας φόρμας excel για την ταξινόμηση των εγγράφων και τον εντοπισμό σχετικών μελετών σύμφωνα με τα ερευνητικά κριτήρια. Αυτό μας βοήθησε να προτείνουμε νέους τύπους ταξινομήσεων σε κάθε μία από τις τρεις ερευνητικές ερωτήσεις.

Αρχικά εστιάσαμε σε μια ανάλυση και αντιστοίχιση για τους σκοπούς των δύο πρώτων κριτηρίων και στη συνέχεια εργαστήκαμε στη σύνθεση των δύο άλλων κριτηρίων. Ο βασικός στόχος της διεξαγωγής μίας διαδικασίας *σύνθεσης δεδομένων* είναι η αξιολόγηση των προσδιορισμένων μελετών για τον εντοπισμό ετερογένειας ή η αναζήτηση σημείων σύνθεσης με την ερμηνεία κοινών πτυχών. Όλα τα παραπάνω μπορούν να βασίζονται σε μία θεματική ανάλυση και σύνθετη

αφήγηση. Εάν η ετερογένεια που είχε παρατηρηθεί ήταν σε υψηλότερο βαθμό, ίσως να έπρεπε να υιοθετηθούν και στατιστικές μέθοδοι ενσωμάτωσης, αλλά είναι λιγότερο σπάνιες στη τεχνολογία λογισμικού.

Στην προσέγγισή μας επικεντρωθήκαμε στη μέθοδο της θεματικής σύνθεσης, η οποία επιδιώκει να κατανοήσει και να συγκεντρώσει τις έννοιες πολλαπλών μελετών. Το θεματικό στάδιο ανάλυσης περιελάβανε: εξαγωγή δεδομένων, κωδικοποίηση δεδομένων, μετάφραση κωδικοποιημένων δεδομένων σε θέματα, δημιουργία μοντέλων με βάση θέματα υψηλότερης τάξης, εκτίμηση των αποτελεσμάτων της σύνθεσης.

2.6 ΑΠΟΤΕΛΕΣΜΑΤΑ

Σε αυτή την ενότητα παρουσιάζουμε τα αποτελέσματα της συστηματικής βιβλιογραφικής ανασκόπησης.

1. Ποια είναι τα εργαλεία, οι διαδικασίες και οι μεθοδολογίες δοκιμών σε εφαρμογές διαδικτύου που χρησιμοποιούνται σήμερα;

1.1 Ποια εργαλεία έχουν προταθεί και ποιες είναι οι δυνατότητές τους;

48 από τα 125 έγγραφα (38%) παρείχαν υποστήριξη εργαλείων για τον έλεγχο εφαρμογών στο διαδίκτυο. Αυτό έθεσε το ενδιαφέρον μας για τα εργαλεία που ήταν διαθέσιμα στο Διαδίκτυο για να διερευνηθούν οι δυνατότητές τους (δηλ. Ποιο μέρος της διαδικασίας δοκιμής υποστηρίζουν, ποιες τεχνολογίες ανάπτυξης διαδικτυακών εφαρμογών υποστηρίζουν). Ωστόσο, από αυτά τα 48 έγγραφα μόνο 5 παρέχουν ακόμα τα εργαλεία τους διαθέσιμα για λήψη. Αυτή η λίστα απεικονίζεται στον παρακάτω πίνακα.

Πίνακας 3. Κατάλογος των εργαλείων δοκιμών εφαρμογών web που είναι διαθέσιμα για λήψη

Εργαλείο	Αναφορά	Μέρος (εργασία) της διαδικασίας ελέγχου	Σύνδεσμος	Web τεχνολογίες
TestFabrik	Dallmeier et al. (2013)	Reverse engineering Model-based testing Cross-browser compatibility testing	https://testfabrik.com/	HTML, JavaScript, CSS
Web Tools Suite	Marchetto and Tonella (2010)	Reverse engineering	http://selab.fbk.eu/marchetto/	Ajax, DOM, JSP, Servlet

EXACT	Mirshokraie and Mesbah (2012) Amalfitano et al. (2010a)	Regression testing	http://wpage.unina.it/ptramont/downloads.htm	Javascript, DOM
Crawljax	Roest et al. (2010)	Regression testing	http://crawljax.com/	Ajax, DOM
Mutandis	Mirshokraie et al. (2013)	Mutation testing	www.github.com/saltlab/mutandis	Javascript, DOM

1.2 Ποιοι τύποι διαδικασιών / μεθοδολογιών δοκιμών έχουν προταθεί;

Ένας σημαντικός αριθμός μελετών (33%) επικεντρώνεται σε τεχνικές αντιστροφής κώδικα (reverse engineering). Αυτές οι τεχνικές αντιστρέφουν τον κώδικα σε ένα μοντέλο που βοηθά στη συνέχεια να συμπεράνει αν ο κώδικας αντιστοιχεί στον αρχικό σχεδιασμό, ή αλλιώς σε μία ροή μεταξύ των σελίδων μίας εφαρμογής web. Επίσης, μπορεί να αναστραφούν σε μοντέλα ροής δεδομένων για να βοηθήσουν στην επικύρωση της εισαγωγής και εξόδου δεδομένων μεταξύ των σελίδων. Ενώ πιο προηγμένες προσεγγίσεις εστιάζουν στο μοντέλο αντικειμένου εγγράφου (DOM), το οποίο επιτρέπει τη δοκιμή των διαφόρων αντικειμένων που συνθέτουν μια δυναμική σελίδα (π.χ. xml, javascript, xhtml, html).

Άλλες τεχνικές έδωσαν έμφαση στην ανάπτυξη διαδικασιών για την αναγνώριση σφαλμάτων. Αυτά ακολουθούν επίσης μια τεχνική αντιστροφής κώδικα για την κατασκευή μοντέλων σφαλμάτων ή ταξινόμησης σφαλμάτων όπως παρουσιάζονται παρακάτω:

Πίνακας 4. Μοντέλο σφάλματος / ταξινόμηση σφαλμάτων - ερευνητική εργασία

Μοντέλο / ταξινόμηση σφαλμάτων	Αναφορά
CSS	Dobolyi and Weimer(2010)
Code Disclosure	Dobolyi and Weimer(2010)
Scripting	Elbaum et al. (2005) and Jensen and Moller (2011)
DOM	Mesbahetal (2012), Mirshokraie and Mesbah(2012) and Marchettoetal (2008)
Cross-browser incompatibility	Marchettoetal (2007) and Mesbahetal (2012)
Plugin	Marchettoetal (2007)
Cookie manipulation	Marchettoetal (2007)
Syntactic,	Ozkinaci and Betin Can (2011), Artzi et al. (2008) and Artzi and Pistoia (2010)

Programming language	
Multi tier architectural	Luoetal (2009)
Appearance (GUI, layout, etc.)	Sprenkleetal (2007), Luoetal (2009), Dobolyi and Weimer (2010), Peng and Lu (2011) and Choudharyetal (2010)
Database, persistence	Sprenkle et al. (2007), Marchetto et al. (2007), Elbaum et al. (2005), Dobolyi and Weimer (2010) and Peng and Lu (2011)
Form construction	Marchettoetal (2008), Sprenkle et al. (2007) and Elbaumetal (2005)
Session	Marchetto et al. (2008), Marchetto et al. (2007) and Dobolyi and Weimer (2010)
Asynchronous Communication	Marchettoetal (2008) and Pattabiraman and Zorn (2010)
Navigation	Marchetto et al. (2008a), Praphamontripong and Offutt (2010), Sprenkle et al. (2005), Luo et al. (2009), Ettema and Bunch (2010), Kallepalli and Tian (2001), Dobolyi and Weimer (2010) and Peng and Lu (2011)
Portability	Marchettoetal (2008)
Functionality	Marchetto et al. (2008a), Sprenkleetal (2007),Dobolyi and Weimer (2010) and Choudhary et al (2010)
Internationalization	Marchetto et al (2008a) and Dobolyi and Weimer(2010)
Authentication, permission	Marchetto et al (2008a), Luo et al (2009), Kallepalli and Tian (2001) and Dobolyi and Weimer (2010)

2. Ποια είναι τα εργαλεία, οι διαδικασίες και οι μεθοδολογίες που χρησιμοποιούνται σήμερα στον έλεγχο εφαρμογών Η/Υ;

2.1 Ποια εργαλεία έχουν προταθεί και ποιες είναι οι δυνατότητές τους;

Το TET (Test Environment Toolkit) παρέχεται ως προϊόν γραμμής εντολών ανοιχτού κώδικα, το οποίο δεν υποστηρίζεται και το οποίο ικανοποιεί τις τρέχουσες και αναμενόμενες μελλοντικές ανάγκες δοκιμών της κοινότητας ανάπτυξης εργαλείων ελέγχου.

Το TETware είναι η υποστηριζόμενη έκδοση του Test Environment Toolkit, το οποίο επιτρέπει στους προγραμματιστές να κάνουν τη διαχείριση των ελέγχων, τον ορισμό μίας ακολουθίας δοκιμών, την αναφορά του αποτελέσματος μίας δοκιμής σε μια τυποποιημένη μορφή. Αυτό μπορεί να εφαρμοστεί τόσο στα λειτουργικά συστήματα UNIX όσο και στα Microsoft Windows, διασφαλίζοντας τη δυνατότητα μεταφοράς των δοκιμαστικών περιπτώσεων.

Αυτοματοποιημένο Δοκιμή και Retest (ATRT): ο διαχειριστής ελέγχου είναι ένα αυτοματοποιημένο εργαλείο ελέγχου λογισμικού που σχεδιάστηκε για να

υποστηρίζει τη δοκιμή ετερογενών, πολύπλοκων συστημάτων. Καλύπτει ολόκληρο τον κύκλο δοκιμών του λογισμικού και υποστηρίζει την πλήρη κάλυψη των δοκιμών, συμπεριλαμβανομένων:

- Έλεγχος λειτουργιών (functional testing)
- Έλεγχος παλινδρόμησης (regression testing)
- Δοκιμή διεπαφών (interface testing)
- Έλεγχος συγχρονισμού (concurrency testing)
- Έλεγχος αντοχής (stress testing)
- Έλεγχος απόδοσης (performance testing)

Το RTH (Requirements and Testing Hub) είναι ένα εργαλείο διαχείρισης δοκιμών ανοιχτού κώδικα, το οποίο διαθέτει επίσης δυνατότητες διαχείρισης απαιτήσεων και παρακολούθησης σφαλμάτων.

Το Ranorex Studio παρέχει στους μηχανικούς ελέγχου ένα ολοκληρωμένο σύνολο εργαλείων για δοκιμές από άκρο σε άκρο για εφαρμογές Η/Υ, διαδικτύου και την κινητή τηλεφωνία. Ελέγχει εφαρμογές από την οπτική γωνία του χρήστη, χρησιμοποιώντας τυπικές γλώσσες και κοινές τεχνικές προγραμματισμού. Δεν απαιτεί από τους προγραμματιστές να γνωρίζουν μια scripting γλώσσα. Το ίδιο το εργαλείο είναι γραμμένο σε κώδικα C # ή VB.net. Ως εκ τούτου, παρέχει τη δυνατότητα να κάνει αυτοματοποίηση δοκιμών στο περιβάλλον του πελάτη χρησιμοποιώντας τυπικές και σύγχρονες τεχνικές προγραμματισμού. Επιτρέπει τη δημιουργία δοκιμαστικών περιπτώσεων με ελάχιστες γνώσεις προγραμματισμού.

Το LoadTracer είναι ένα εργαλείο ελέγχου απόδοσης το οποίο λειτουργεί σε τρεις φάσεις. Πρώτα καταγράφει το σενάριο, μετά προσομοιώνει με εκατοντάδες εικονικούς πελάτες και στη συνέχεια αναλύει τα αποτελέσματα. Σε κάθε φάση χρησιμοποιείται ένα αντίστοιχο εργαλείο.

Πίνακας 5. Κατάλογος των εργαλείων δοκιμής εφαρμογών Η/Υ διαθέσιμων για λήψη

Εργαλείο	Αναφορά	Μέρος (εργασία) της διαδικασίας ελέγχου	Σύνδεσμος
TET(Test Environment Toolkit)	Pressman (2014)	Test management: produces a specification for a test environment	http://tetworks.opengroup.org/Products/tet.htm
TETWare	Mittal and Kumar (2014)	Test management: Test administration, test sequence specification	http://tetworks.opengroup.org/Products/tetware.htm
Automated Test and ReTest (ATRT)		Test management of: <ul style="list-style-type: none"> • Functional Testing 	http://www.testingtoolsguide.net/tools/atrt-test-manager/

		<ul style="list-style-type: none"> • Regression Testing • Interface Testing • Concurrency Testing • Endurance Testing • Performance Testing 	
RTH - Requirements and Testing Hub	Padmavathi, Saravanan (2017)	test management and requirements management tool	https://sourceforge.net/p/rth/wiki/Home/
Ranorex	Chauhan and Singh (2014)	Functional testing for graphical applications	https://www.ranorex.com/
LoadTracer	Pressman (2014)	Load (performance) simulation and testing	http://www.loadtracer.com/

2.2 Ποιοι τύποι διαδικασιών / μεθοδολογιών δοκιμών έχουν προταθεί;

Μερικές από τις πιο διαδεδομένες τεχνικές δοκιμών λογισμικού απεικονίζονται στον ακόλουθο πίνακα. Η δοκιμή ορθότητας θέτει τα ελάχιστα πρότυπα του αναμενόμενου αποτελέσματος ενός λογισμικού. Καθορίζει τη σωστή συμπεριφορά ενός συστήματος από τη λάθος. Αυτό θα περιλαμβάνει τη πραγματοποίηση white box ή black box δοκιμών, καθώς οι εσωτερικές λεπτομέρειες της ενότητας λογισμικού (που θα δοκιμαστεί) ενδέχεται να μην είναι γνωστή σε έναν μηχανικό ελέγχου.

Ο έλεγχος white box επικεντρώνεται στην επικύρωση της εσωτερικής εργασίας και της δομής μιας εφαρμογής. Παίρνει ως προϋπόθεση ότι ο μηχανικός ελέγχου έχει γνώση του πηγαίου κώδικα και εφαρμόζει τότε μία συγκεκριμένη διαδικασία ελέγχου του λογισμικού σε επίπεδο ολοκλήρωσης, μονάδας και συστήματος της.

Στη βιβλιογραφία υπάρχουν πολλές αναφορές σχετικά με τον έλεγχο τύπου Black Box. Η μεθοδολογία είναι η τυπική προσέγγιση για την επικύρωση των προδιαγραφών μιας εφαρμογής χωρίς αναφορά στην εσωτερική της εργασία. Ελέγχει αν το συγκεκριμένο λογισμικό ταιριάζει με την απαίτηση χωρίς να έχει υπόψη την εσωτερική λογική δομή του συστήματος. Εξετάζει μόνο τη βασική πτυχή του συστήματος.

Οι ερευνητικές εργασίες στις μεθόδους δοκιμής Black Box υπογραμμίζουν την ελάχιστη ή μηδαμινή συμμετοχή των χρηστών στις φάσεις των λειτουργικών δοκιμών, δοκιμών αντοχής, δοκιμών φορτίου, δοκιμών ad-hoc, διερευνητικών δοκιμών, δοκιμών ευχρηστίας, δοκιμών ανάκτησης λειτουργίας μετά από σφάλμα, και δοκιμών διαχείρισης μεγάλου όγκου δεδομένων. Από την άλλη πλευρά, η έρευνα σχετικά με τις τεχνικές δοκιμής Black Box, στις οποίες απαιτείται η συμμετοχή του χρήστη, εξετάζει τις δοκιμές αποδοχής από τους χρήστες, τις δοκιμές τύπου Alpha και τις δοκιμές τύπου Beta.

Τα τελευταία χρόνια, στη βιβλιογραφία για τις τεχνικές ελέγχου Black Box έχουν δημοσιευτεί αποτελέσματα σχετικά με τεχνικές ελέγχου που βασίζονται σε γράφους, την κατανομή ισοδυναμίας, την ανάλυση οριακής αξίας, συγκριτικές δοκιμές, τις δοκιμές ορθογωνίας συστοιχίας, εξειδικευμένες δοκιμές, δοκιμές εντοπισμού ασάφειας και μετρήσεις ιχνηλασιμότητας. Επίσης, η δοκιμή Grey Box ή η αντιστροφή κώδικα στοχεύει στη βελτιστοποίηση των αποτελεσμάτων των δοκιμών έχοντας κάποιες γνώσεις σχετικά με τον κώδικα και τη ροή δεδομένων / ελέγχου.

Η δοκιμή απόδοσης πραγματοποιείται για να αξιολογηθεί η συμμόρφωση ενός συστήματος ή μίας συνιστώσας του λογισμικού του με συγκεκριμένες απαιτήσεις απόδοσης. Οι ερευνητικές εργασίες σε αυτό το θέμα προτείνουν εναλλακτικά μοντέλα και διαδικασίες για την αξιολόγηση της απόδοσης ενός συστήματος λογισμικού σε σχέση με τη χρήση πόρων, τη δυναμική και τον χρόνο απόκρισης του συστήματος ή συνιστώσας του λογισμικού. Ή στην περίπτωση ενός ιστότοπου, παρόμοιες τεχνικές επικυρώνουν την καθυστέρηση της ιστοσελίδας, τη διακίνηση δεδομένων από και προς αυτή, και τη χρήση της.

Η δοκιμή φόρτου καλείται να μετρήσει εάν το σύστημα μπορεί να υποστηρίξει έναν ελάχιστο αριθμό χρηστών ή όχι. Υπάρχουν ορισμένες προτάσεις για διαδικασίες μέτρησης φόρτου χρησιμοποιώντας ως βασικό συστατικό τη δημιουργία εικονικών χρηστών για να ελεγχθεί αν το σύστημα μπορεί να εκτελεστεί χωρίς σημεία συμφόρησης ή όχι και σε ποια τμήματα της εφαρμογής.

Οι μέθοδοι ελέγχου αντοχής εκτελούν τυχαίες επιχειρησιακές ακολουθίες σε μεγάλη κλίμακα και πάνω από την κανονική ταχύτητα (ή περισσότερο από τις κανονικές χρονικές περιόδους), για την εύρεση ελαττωμάτων. Μετρούν την ευρωστία μιας εφαρμογής ή συστήματος μέσω συγκεκριμένων κριτηρίων ή καθορίζουν εάν δεν επιτυγχάνουν το ελάχιστο. Οι μέθοδοι δοκιμών αντοχής καλούνται να αξιολογήσουν τη συμπεριφορά διαφορετικών τύπων εφαρμογών (π.χ. αυτόνομων, διανεμημένων, βάσεων δεδομένων ή δικτύων) με μεγάλη βάση χρηστών.

Η δοκιμή αξιοπιστίας παρουσιάζεται στη βιβλιογραφία ως σημαντικό στοιχείο της διαδικασίας ελέγχου για να ανακαλυφθούν οι αστοχίες του συστήματος και να τις απομακρύνει πριν από την διάθεση του στην παραγωγή. Αυτό μπορεί να

εφαρμοστεί με μια αποτελεσματική μέθοδο δειγματοληψίας για να μετρηθεί το παρόν και να προβλεφθεί η μελλοντική αξιοπιστία του λογισμικού.

Δοκιμές ασφάλειας εφαρμόζονται για να διασφαλιστεί ότι ικανοποιούνται οι απαιτήσεις λογικής πρόσβασης (δηλαδή μόνο εξουσιοδοτημένο προσωπικό μπορεί να έχει πρόσβαση στην εφαρμογή με τα κατάλληλα δικαιώματα ανά λειτουργία ή ενότητα). Με αυτόν τον τρόπο αποκαλύπτονται τυχόν σημαντικά κενά ασφάλειας και ευπάθειες της εφαρμογής που μπορεί να εκμεταλλευτεί ένας εξουσιοδοτημένος χρήστης.

Αυτόματος έλεγχος και επανέλεγχος (Automated Test and ReTest - ATRT): παρέχει δυνατότητες αυτοματοποιημένου ελέγχου τόσο στη γραφική διεπαφή χρήστη (GUI), όσο και σε διεπαφές ανταλλαγής μηνυμάτων που χρησιμοποιούν πολλαπλά πρωτόκολλα (TCP / IP, UDP, Multicast, κλπ.). Ο διαχειριστής δοκιμών παρέχει ένα κεντρικό αποθετήριο με αντικείμενα ή πρότυπα τα οποία είναι χρήσιμα σε διάφορα στάδια της διαδικασίας ελέγχου: από την ανάπτυξη δοκιμαστικών περιπτώσεων μέχρι την ανιχνευσιμότητα απαιτήσεων και την αναφορά των αποτελεσμάτων των δοκιμαστικών περιπτώσεων. Έχοντας ορίσει κάποιο μοντέλο, χωρίς συγκεκριμένες δέσμες ενεργειών, δεν επηρεάζει τη λειτουργία του συστήματος ενώ πραγματοποιεί ελέγχους σε αυτό.

Πίνακας 6. Κατάλογος τεχνικών ελέγχου εφαρμογών για Η/Υ

Τεχνική ελέγχου λογισμικού	Αναφορά
Correctness Testing (black or white box testing or greybox testing)	(Jiantao Pan, 2008), (Laurie W., 2013)
Performance Testing	(Collins et al, 2012)
Reliability Testing	(Jiantao Pan, 2008)
Security Testing	(Berner et al, 2005)
Automated Test and ReTest	(Memon et al, 2013)

3. Πώς διαδεδομένες εμπειρικές μελέτες που ασχολούνται με το αντικείμενο του ελέγχου κώδικα σε εφαρμογές web και desktop αντίστοιχα;

Ο παρακάτω πίνακας παρουσιάζει εμπειρικές μελέτες για τον έλεγχο διαδικτυακών εφαρμογών στην πλευρά του τελικού χρήστη. Οι περισσότερες από αυτές τις μελέτες επικεντρώνονται στο τμήμα HTML και λιγότερο στις τεχνολογίες DOM και Javascript.

Πίνακας 7. Εμπειρικές μελέτες σχετικά με τη δοκιμή διαδικτυακών εφαρμογών ιστού (στη πλευρά του τελικού χρήστη)

	HTML	DOM	Javascript
Validation	Artzi et al. (2011), Praphamontripong and Offutt (2010), Sprenkle et al. (2005b, 2007, 2008, 2012), Luo et al. (2009), Sampath et al. (2005), Alshahwan et al. (2012), Marchetto et al. (2007), Thummalapenta et al. (2013), Alshahwan et al. (2009), Mirshokraie and Mesbah (2012), Kallepalli and Tian (2001), Li et al. (2010), Sakamoto et al. (2013),	Mesbah and Prasad (2011), Choudhary et al. (2012), Pattabiraman and Zorn (2010), Thummalapenta et al. (2013), Mirshokraie and Mesbah (2012), Jensen and Moller (2011), Li et al. (2010), Sakamoto et al. (2013), Amalfitano et al. (2010a), Marchetto et al. (2008b), Marchetto (2008) and Choudhary et al. (2010)	Artzi et al. (2011), Saxena et al. (2010), Mesbah and Prasad (2011), Alshahwan et al. (2012), Choudhary et al. (2012), Pattabiraman and Zorn (2010), Marchetto et al. (2007), Alshahwan et al. (2009), Mirshokraie And Mesbah (2012), Jensen and Moller (2011), Marchetto et al. (2008b), Choudhary et al. (2010) and Dallmeier et al. (2013)
Evaluation	Marchetto (2008), Hao and Mendes (2006), Tian and Ma (2006), Choudhary et al. (2010) and Dallmeier et al. (2013)	-	-

Παρομοίως, υπάρχουν εμπειρικές μελέτες στην πλευρά του διακομιστή των διαδικτυακών εφαρμογών. Σε αυτή την περίπτωση, υπάρχει μια μεγαλύτερη ποικιλία λόγω των πολλαπλών τεχνολογιών που χρησιμοποιούνται. Είναι όμως προφανές ότι η Java (J2EE) κυριαρχεί στην εμπειρική έρευνα καθώς και η PHP.

Πίνακας 8. Εμπειρικές μελέτες σχετικά με τις δοκιμές web εφαρμογών (πλευρά εξυπηρετητή)

	PHP	J2EE	.Net	Perl/CGI	Άλλο
Validation	Marback et al. (2012), Dobolyi et	Sampath et al. (2007), Praphamontripon	Ozkinaci and Betin Can	Ran et al. (2009)	-

	al. (2010), Alshahwan et al. (2012), Ettema and Bunch (2010), Marchetto et al. (2007), Artzi et al. (2008), Artzi and Pistoia (2010), Marchetto (2008) and Choudhary et al. (2010)	g and Offutt(2010), Halfond and Orso (2008), Sprenkle et al. (2005b, 2007, 2012), Luo et al. (2009), Sampath et al. (2005, 2006a), Thummalapenta et al. (2013), Halfond et al. (2009), Amalfitano et al. (2010a), Marchetto et al. (2008b), Liu And Kuan Tan (2008) and Peng and Lu (2011)	(2011) and Marchetto et al. (2007)		
Evaluation	Sprenkle et al. (2005a), Alshahwan and Harman (2011) and Mesbah et al. (2012)	Sprenkle et al. (2005a, 2011), Mesbah et al. (2012), Sampath and Bryce (2008), Marchetto and Tonella (2010) and Sampath et al. (2006b)	-	Elbaum et al. (2005)	Marchetto and Tonella (2010)

Έχει επίσης υπάρξει ένα ευρύ φάσμα εμπειρικών μελετών για δοκιμές εφαρμογών σε υπολογιστές. Οι μελέτες αυτές δίνουν έμφαση κυρίως στη δημιουργία περιπτώσεων με δοκιμές ελέγχου, την επιλογή περιπτώσεων ελέγχου και το φιλτράρισμα ή αλλιώς τη μείωση του αριθμού περιπτώσεων ελέγχου (πίνακας παρακάτω). Επιπλέον, έχει σημειωθεί σημαντική δουλειά όσον αφορά την εφαρμογή μετρήσεων σε δοκιμές λογισμικού για υπολογιστές. Αυτό είναι σημαντικό για τη μέτρηση του κόστους, των χρονικών περιορισμών ή την υποστήριξη αυτοματοποίησης. Λιγότερες εμπειρικές μελέτες εντοπίζονται για τον εντοπισμό σφαλμάτων.

Πίνακας 9. Εμπειρικές μελέτες σχετικά με τον έλεγχο εφαρμογών Η/Υ

Επιλογή τεχνικής ελέγχου	Εντοπισμός σφάλματος	Μετρικές ελέγχου
Jung-Min Kim et al (2005), Graves, Todd L. et al (2001) S. Ali, L. C. Briand et al (2010), M. Harman and P. McMinn (2010), W. Masri, A. Podgurski and D. Leon (2007), G. Fraser and A. Arcuri (2012), Luo, Qingzhou et al (2014), S. Elbaum, A. G. Malishevsky and G. Rothermel (2002), L. Zhang, D. Marinov, L. Zhang and S. Khurshid (2011), J. D. McCaffrey (2010), Engstrom, E. et al (2008)	Y. Yu, J. Jones and M. J. Harrold (2008), Xiao Q. et al (2008), S. Shamsiri et al (2015)	Smith and Kapfhammer (2010) Walcott K. R. et al (2006), Do, Hyunsook and Rothermel, Gregg (2006), Jussi Kasurinen, Ossi Taipale, and Kari Smolander (2010), Kapfhammer, Gregory M. and Soffa, Mary Lou (2003), Do, Hyunsook et al (2008)

4. Ποια είναι η τρέχουσα κατάσταση της τεχνολογίας σχετικά με τον έλεγχο εφαρμογών στο διαδίκτυο και τους Η/Υ;

Από τον Πίνακα 10 γίνεται φανερό ότι μόνο λίγες δημοσιεύσεις έχουν ανακτηθεί, οι οποίες αναφέρονται σε εργασίες τελευταίας τεχνολογίας, είτε για δοκιμές διαδικτυακών εφαρμογών είτε για δοκιμές εφαρμογών Η/Υ. Ενώ τα άρθρα σχετικά με τον έλεγχο web εφαρμογών επικεντρώνονται στην τεχνολογία των εργαλείων, των διαδικασιών και των μελλοντικών τεχνολογικών τάσεων σχετικά με τον τρόπο εκτέλεσης των ελέγχων, τα τεστ ελέγχου εφαρμογών Η/Υ εξετάζουν την τρέχουσα κατάσταση της τεχνολογίας στην δημιουργία περιπτώσεων ελέγχου.

Πίνακας 10. Έλεγχος εφαρμογών στο διαδίκτυο και για Η/Υ – τρέχουσα τεχνολογία

Έλεγχος διαδικτυακών εφαρμογών – τρέχουσες τεχνολογικές εξελίξεις	Έλεγχος εφαρμογών Η/Υ – τρέχουσες τεχνολογικές εξελίξεις
Giuseppe A. et al (2006) Arora A., Sinha M. (2012)	Kull (2012), Parhizgar (2012)

ΚΕΦΑΛΑΙΟ 3

ΣΥΜΠΕΡΑΣΜΑΤΑ

Τόσο οι διαδικτυακές εφαρμογές όσο και οι εφαρμογές των Η/Υ είναι σήμερα ισχυρά κανάλια για την ικανοποίηση των αναγκών των χρηστών αντίστοιχα. Και στους δύο τομείς (web desktop) η πολυπλοκότητα αυξάνεται λόγω της δυναμικής που επιβάλλουν οι λειτουργίες των σύγχρονων εφαρμογών. Ειδικά στην περίπτωση που η χρήση των διαδικτυακών εφαρμογών αυξάνεται ολοένα και περισσότερο δημιουργούν πρόσθετες προκλήσεις για τους προγραμματιστές τέτοιων εφαρμογών.

Σε αυτή τη συστηματική εργασία ανασκόπησης της βιβλιογραφίας με τη μέθοδο SLR παρουσιάσαμε μια σειρά άρθρων που δημοσιεύθηκαν μεταξύ του 2000 και του 2017 στον τομέα των εφαρμογών web και desktop αντίστοιχα. Με βάση τη στρατηγική της επιλογής άρθρων που ακολουθήσαμε, 125 μελέτες συμπεριλήφθηκαν στη μελέτη μας. Όσον αφορά τα τέσσερα ερευνητικά ερωτήματα, αναλύσαμε τα σχετικά δεδομένα, τις εμπειρικές μελέτες και τη πιο πρόσφατη εξέλιξη της τεχνολογίας προκειμένου να διαμορφώσουμε ένα πρωτόκολλο ανασκόπησης και να απαντήσουμε σε αυτά τα ερευνητικά ερωτήματα.

Αναφορικά με το πρώτο (1^ο) ερευνητικό ερώτημα της πτυχιακής εργασίας (βλ. §2.2 Ερευνητικές Ερωτήσεις και §2.6 Αποτελέσματα) μόλις 48 από τα 125 έγγραφα (38%) σημείωναν ότι συγκεκριμένα εργαλεία έχουν αναπτυχθεί για τον έλεγχο εφαρμογών και μόνο 5 από τα 48 ήταν διαθέσιμα για λήψη. Και στους δύο τομείς έχουν γίνει μελέτες για εφαρμογή αντιστροφής κώδικα σε κάποιο μοντέλο που να προσεγγίζει τον αρχικό σχεδιασμό (33%). Οι τεχνικές αυτές συμβάλλουν στην ανάπτυξη διαδικασιών ή μοντέλων για την αναγνώριση και ταξινόμηση σφαλμάτων.

Στο δεύτερο ερώτημα, περισσότερο αναδιπλώνεται μία ευρύτερη γκάμα εξειδικευμένων τεχνικών ελέγχου σε εφαρμογές Η/Υ (λειτουργιών, διεπαφών, αντοχής, απόδοσης) παρά σε διαδικτυακές εφαρμογές.

Στο τρίτο ερώτημα διαπιστώνεται ότι εμπειρικές μελέτες έχουν πραγματοποιηθεί και στους δύο τομείς. Στη περίπτωση των web εφαρμογών ο έλεγχος εστιάζει περισσότερο στη πλευρά του διακομιστή των εφαρμογών καθώς εκεί παρουσιάζεται και ποικιλομορφία στη χρήση τεχνολογιών. Έχει επίσης υπάρξει ένα ευρύ φάσμα εμπειρικών μελετών για δοκιμές εφαρμογών σε υπολογιστές. Οι μελέτες αυτές δίνουν έμφαση κυρίως στη δημιουργία περιπτώσεων ελέγχου και τον ορισμό μετρικών για τη μέτρηση του κόστους, των χρονικών περιορισμών ή

την υποστήριξη αυτοματοποίησης. Λιγότερες εμπειρικές μελέτες εντοπίζονται για τον εντοπισμό σφαλμάτων.

Αναφορικά με το τέταρτο ερευνητικό ερώτημα διαπιστώνεται ότι λίγες είναι αυτές οι έρευνες που εστιάζουν στην περιγραφή των τεχνολογικών εξελίξεων σχετικά με τον έλεγχο λογισμικού. Παρατηρείται επίσης σημαντική απόκλιση μεταξύ των στόχων στους δύο τομείς (πχ εργαλεία, διαδικασίες και μελλοντικές τεχνολογικές προκλήσεις στις web εφαρμογές έναντι δημιουργία περιπτώσεων ελέγχου στις desktop εφαρμογές).

Συνολικά, η έρευνα μας αποκάλυψε ότι οι δοκιμές στις web εφαρμογές είναι πιο ενεργός τομέας έρευνας παρά οι δοκιμές σε desktop εφαρμογές. Ίσως οι δοκιμές εφαρμογών για Η/Υ να ξεπεραστούν επίσης από άλλους πιο ενεργούς ερευνητικούς τομείς, όπως ο έλεγχος εφαρμογών για κινητά τηλέφωνα ή στο υπολογιστικό νέφος (cloud).

Και στις δύο περιπτώσεις, αντλήσαμε χρήσιμα διδάγματα από την εξέταση των εξαγόμενων δεδομένων και άρθρων:

1. τους τύπους των μοντέλων ελέγχου
2. τα μοντέλα σφαλμάτων / ταξινόμησης σφαλμάτων για τις web εφαρμογές
3. εργαλεία ελέγχου για web εφαρμογές, καθώς και εργαλεία ελέγχου για desktop εφαρμογές και τις δυνατότητές τους
4. διαδικασίες και μεθοδολογίες για την εκτέλεση εφαρμογών Η/Υ
5. μετρήσεις που χρησιμοποιούνται για την αξιολόγηση των περιορισμών κόστους και χρόνου και στους δύο τύπους δοκιμών
6. τρέχουσες εξελίξεις στην τεχνολογία και μελλοντικές τάσεις και στους δύο τύπους

Πολύ ενδιαφέρουσα παρατήρηση είναι ότι:

- a. Αν και μια ταξινόμηση σφαλμάτων θα ήταν επίσης χρήσιμη στον έλεγχο εφαρμογών για υπολογιστές, δεν βρήκαμε μια τέτοια σχετική δημοσίευση
- b. Παρόλο που μια αυτοματοποιημένη τεχνική και μοντέλα δημιουργίας περιπτώσεων ελέγχου θα ήταν επίσης χρήσιμα στον έλεγχο διαδικτυακών εφαρμογών, δεν ανακτήσαμε μια τέτοια πρόταση.
- c. Πολύ λίγα άρθρα ανακτήθηκαν σε σχέση με την κατάσταση της τεχνολογίας για τη διεξαγωγή ελέγχων στο διαδίκτυο και τους Η/Υ.

Μία μελλοντική έρευνα – ως συνέχεια της τωρινής - θα ήταν η δημιουργία ενός ψηφιακού αποθετηρίου βιβλιογραφικών αναφορών και στους δύο τύπους δοκιμών και τη χρήση του ως βάση για την πραγματοποίηση μίας συστηματικής χαρτογράφησης (systematic mapping).

ΒΙΒΛΙΟΓΡΑΦΙΑ

- Afzal W. et al. (2009). A systematic review of search-based testing for non-functional system properties. *Elsevier Information and Software Technology*, Volume 51, Issue 6, June 2009, Pages 957-976.
- Alalfi M.H., Cordy, J.R., Dean, T.R. (2009). Modelling methods for web application verification and testing: state of the art. . *Elsevier Journal of Software Testing, Verification & Reliability*, vol. 19.
- Amalfitano, D., Fasolino, A., Tramontana, P. (2010b). Techniques and tools for rich internet applications testing. *Proceedings IEEE International Symposium on Web Systems Evolution*, 63-72.
- André Takeshi et al. (2010). A systematic review on formal testing approaches for web services. *4th Brazilian Workshop on Systematic and Automated Software Testing (SAST)*.
- Banerjee I. et al. (2013). Graphical user interface (GUI) testing: Systematic mapping and repository. *Elsevier Information and Software Technology*.
- Bozkurt M. et al. (2010). Testing web services: a survey. *Technical report TR-10-01, Department of Computer Science, King's College, London*.
- Ferreira de Souza E. et al. (2015). Knowledge management initiatives in software testing: A mapping. *Elsevier Information and Software Technology 57 (2015) 378–391*.
- Garousi V et al. (2015). Software test-code engineering: A systematic mapping . *Elsevier Information and Software Technology*.
- Garousi V. and Mantyla V. (2016). A systematic literature review of literature reviews in software testing. *Elsevier Information and Software Technology*, Volume 80, December 2016, Pages 195-216.
- Garousi V. et al. (2013). A Systematic Mapping Study of Web Application Testing. *Elsevier Information and Software Technology*, Volume 55, Issue 8, August 2013, Pages 1374-1396.
- Garoussi V and Mantyla M. (2016). When and what to automate in software testing? A multi-vocal literature review. *Information and Software Technology*.
- Jamil M. et al. (2016). Software Testing Techniques: A Literature Review. *2016 6th International Conference on Information and Communication Technology for The Muslim World*.
- Jia Y. and Harman M. (2003). Mutation Testing. <http://www.dcs.kcl.ac.uk/pg/jiayue/repository>.

- Kam, B., Dean, T.R., (2009). Lessons learned from a survey of web applications testing. *In Proceedings of the International Conference on Information Technology: New Generations*.
- Lee J. et al. (2012). Survey on software testing practices. *IET Software* 6(3):275-282 · June 2012 .
- Lucca, G.A.D., Fasolino, A.R. (2006). Testing web-based applications: the state of the art and future trends. *Information and Software Technology*, vol. 48.
- Memon A.M., Nguyen B.N. (2010). Advances in Automated Model-Based System Testing of Software Applications with a GUI Front-End. *Elsevier Advances in Computers*, Volume 80, 2010, Pages 121-162.
- Mesbah A. and Prasad M. (2011). Automated Cross-browser Compatibility Testing. *Proceedings of the 33rd International Conference on Software Engineering*.
- Neto et al. (2011). A systematic mapping study of software product lines testing. *Elsevier Information and Software Technology*, Volume 53, Issue 5, May 2011, Pages 407-423.
- Palacios M. et al. (2011). Testing in Service Oriented Architectures with dynamic binding: A mapping study. *Elsevier Information and Software Technology*, Volume 53, Issue 3, March 2011, Pages 171-189.
- Per Lauvås jr and Andrea Arcuri. (2017). Recent Trends in Software Testing Education: A Systematic Literature Review. *NIK-2018 conference*.
- Prabhakar N. et al. (2018). A Literature Survey of Applications of Meta-heuristic Techniques in Software Testing. *Part of the Advances in Intelligent Systems and Computing book series (AISC, volume 731)*.
- Sabaren L.N et al. (2017). A Systematic Literature Review in Cross-browser Testing. *Journal of Computer Science and Technology*, vol. 18, no. 1, 2018 .
- Santa S., Travassos G.H. (2011). Características de Técnicas de Teste de Software para Uso em Projetos WEB. *Congresso Ibero-Americano em Engenharia de Software*.

ΑΝΑΦΟΡΕΣ

- Graves, Todd L. et al, 2001. An Empirical Study of Regression Test Selection Techniques. *ACM Trans. Software Engineering Methodology*. Vol. 10, no. 2
- Giuseppe A. Di Lucca, Anna Rita Fasolino, 2006. Testing Web-based applications: The state of the art and future trends, *Information and Software Technology*, Volume 48, Issue 12, 2006, Pages 1172-1186, ISSN 0950-5849
- Arora A., Sinha M. 2012. Web Application Testing: A Review on Techniques, Tools and State of Art. *International Journal of Scientific & Engineering Research*, Volume 3, Issue 2, February-2012
- Jung-Min Kim et al, 2005. An empirical study of regression test application frequency. *Journal of Software: Testing, Verification and Reliability*, vol. 15, issue: 4
- Salma Parhizgar, 2012. Testing and Technology: Past, Present and Future. *Theory and Practice in Language Studies*, Vol. 2, No. 1, pp. 174-178, January 2012
- A. Kull, 2012. "Automatic GUI Model Generation: State of the Art," *2012 IEEE 23rd International Symposium on Software Reliability Engineering Workshops*, Dallas, TX, 2012, pp. 207-212.
doi: 10.1109/ISSREW.2012.23
- Do, Hyunsook et al, 2008. An Empirical Study of the Effect of Time Constraints on the Cost-benefits of Regression Testing. *Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of Software Engineering*
- J. D. McCaffrey, 2010. "An Empirical Study of Pairwise Test Set Generation Using a Genetic Algorithm," *2010 Seventh International Conference on Information Technology: New Generations*, Las Vegas, NV, 2010, pp. 992-997. doi: 10.1109/ITNG.2010.93
- Engstrom, E. and Skoglund, Mats and Runeson, Per, 2008. Empirical Evaluations of Regression Test Selection Techniques: A Systematic Review. *Proceedings of the Second ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*
- Jussi Kasurinen, Ossi Taipale, and Kari Smolander, 2010. "Software Test Automation in Practice: Empirical Observations," *Advances in Software Engineering*, vol. 2010, Article ID 620836, 18 pages, 2010.
<https://doi.org/10.1155/2010/620836>.

- L. Zhang, D. Marinov, L. Zhang and S. Khurshid, 2011. "An Empirical Study of JUnit Test-Suite Reduction," *2011 IEEE 22nd International Symposium on Software Reliability Engineering*, Hiroshima, 2011, pp. 170-179. doi: 10.1109/ISSRE.2011.26
- Kapfhammer, Gregory M. and Soffa, Mary Lou, 2003. A Family of Test Adequacy Criteria for Database-driven Applications. Proceedings of the 9th European Software Engineering Conference Held Jointly with 11th ACM SIGSOFT International Symposium on Foundations of Software Engineering
- S. Elbaum, A. G. Malishevsky and G. Rothermel, 2002. "Test case prioritization: a family of empirical studies," in *IEEE Transactions on Software Engineering*, vol. 28, no. 2, pp. 159-182, Feb. 2002. doi: 10.1109/32.988497
- Luo, Qingzhou et al, 2014. An Empirical Analysis of Flaky Tests. Proceedings of the 22Nd ACM SIGSOFT International Symposium on Foundations of Software Engineering
- Walcott K.R. et al, 2006. TimeAware test suite prioritization. Proceedings of the 2006 international symposium on Software testing and analysis
- G. Fraser and A. Arcuri, 2012. "Sound empirical evidence in software testing," *2012 34th International Conference on Software Engineering (ICSE)*, Zurich, 2012, pp. 178-188. doi: 10.1109/ICSE.2012.6227195
- W. Masri, A. Podgurski and D. Leon, 2007. "An Empirical Study of Test Case Filtering Techniques Based on Exercising Information Flows," in *IEEE Transactions on Software Engineering*, vol. 33, no. 7, pp. 454-477, July 2007. doi: 10.1109/TSE.2007.1020
- Do, Hyunsook and Rothermel, Gregg, 2006. An Empirical Study of Regression Testing Techniques Incorporating Context and Lifetime Factors and Improved Cost-benefit Models. Proceedings of the 14th ACM SIGSOFT International Symposium on Foundations of Software Engineering
- Xiao Q. et al, 2008. Configuration-aware Regression Testing: An Empirical Study of Sampling and Prioritization. Proceedings of the 2008 International Symposium on Software Testing and Analysis
- S. Shamshiri et al, 2015. Do Automatically Generated Unit Tests Find Real Faults? An Empirical Study of Effectiveness and Challenges (T). *2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pp. 201-211. doi: 10.1109/ASE.2015.86
- Y. Yu, J. Jones and M. J. Harrold, 2008. An empirical study of the effects of test-suite reduction on fault localization. *2008 ACM/IEEE 30th International*

Conference on Software Engineering, Leipzig, 2008, pp. 201-210. doi: 10.1145/1368088.1368116

S. Ali, L. C. Briand, H. Hemmati and R. K. Panesar-Walawege, 2010. A Systematic Review of the Application and Empirical Investigation of Search-Based Test Case Generation, in *IEEE Transactions on Software Engineering*, vol. 36, no. 6, pp. 742-762, Nov.-Dec. 2010. doi: 10.1109/TSE.2009.52

M. Harman and P. McMinn, 2010. A Theoretical and Empirical Study of Search-Based Testing: Local, Global, and Hybrid Search, in *IEEE Transactions on Software Engineering*, vol. 36, no. 2, pp. 226-247, March-April 2010. doi: 10.1109/TSE.2009.71

Smith A.S, Kapfhammer G.M., 2009. An empirical study of incorporating cost into test suite reduction and prioritization. Proceedings of the 2009 ACM symposium on Applied Computing.

Pressman R. S, 2014. *Software Engineering: A Practitioner's Approach*. McGraw-Hill Education; 8 edition (January 23, 2014)

Mittal D. and Kumar K., 2014. Performance Analysis of TETware and TestKit for Validating Tizen Framework. [2014 International Conference on Devices, Circuits and Communications \(ICDCCom\)](#)

Padmavathi R., Saravanan P., 2017. New Frameworks for Automated Test and Retest (ATRT) Test Case Requirement Traceability Matrix. *International Journal of Innovative Research in Applied Sciences and Engineering (IJIRASE)*, Volume 1, Issue 2, August 2017

R. Kaur Chauhan and I. Singh, 2014. Latest Research and development on software testing techniques and tools, *International Journal of Current Engineering and Technology*, vol. 4, no. 4, 2014.

Jiantao Pan, 2008. Software testing. Available at <https://www.semanticscholar.org/paper/Software-Testing-Pan/4cdec822cec1a9f42d938cc3a9b972610dab16b2>

Laurie W., 2013. *White-Box Testing: 60–61, 69*. <http://www.chaudhary.org/WhiteBox.pdf>

Alalfi, M.H., Cordy, J.R., Dean, T.R., 2010. Automating coverage metrics for dynamic web applications. In: *European Conference on Software Maintenance and Reengineering*, pp. 51–60.

Alshahwan, N., Harman, M., 2011. Automated web application testing using search based software engineering. In: *IEEEACM International Conference on Auto- mated Software Engineering*, pp. 3–12.

- Alshahwan, N., Harman, M., Marchetto, A., Tonella, P., 2009. Improving web application testing using testability measures. In: IEEE International Symposium on Web Systems Evolution, pp. 49–58.
- Alshahwan, N., Harman, M., Marchetto, A., Tiella, R., Tonella, P., Kessler, F.B., 2012.
- Crawlability metrics for web applications. In: IEEE International Conference on Software Testing, Verification and Validation, pp. 151–160.
- Amalfitano, D., Fasolino, A.R., Tramontana, P., 2010, April. Rich internet application testing using execution trace data. In: International Conference on Software Testing, Verification, and Validation Workshops, pp. 274–283.
- Amyot, D., Roy, J., Weiss, M., 2005. UCM-driven testing of web applications. In: Proceedings of the International SDL Forum, pp. 247–264.
- Andrews, A.A., Offutt, J., Alexander, R.T., 2005. Testing web applications by modeling with FSMs. *Software Systems Modeling* 4 (3), 326–345.
- Andrews, A.A., Offutt, J., Dyreson, C., Mallery, C.J., Jerath, K., Alexander, R., 2010. Scalability issues with using FSMWeb to test web applications. *Information and Software Technology* 52 (January (1)), 52–66.
- Artzi, S., Pistoia, M., 2010, May. Practical fault localization for dynamic web applications. In: Proceedings of the 2nd ACM/IEEE International Conference on Software Engineering, vol. 1, pp. 265–274.
- Artzi, S., Kiezun, A., Dolby, J., Tip, F., Dig, D., Paradkar, A., Ernst, M.D., 2008. Finding bugs in dynamic web applications. In: Proceedings of the International Symposium on Software Testing and Analysis, p. 261.
- Artzi, S., Dolby, J., Jensen, S.H., Moller, A., Tip, F., 2011. A framework for automated testing of JavaScript web applications. In: International Conference on Software Engineering, pp. 571–580.
- Bellettini, C., Marchetto, A., Trentini, A., Comelico, V., 2005. TestUml: user-metrics driven web applications testing. *Proceedings of the ACM Symposium on Applied Computing*, 1694–1698.
- Benedikt, M., Freire, J., Godefroid, P., 2002. VeriWeb: automatically testing dynamic
- Bordbar, B., Anastasakis, K., 2006. MDA and analysis of web applications. In: *Trends in Enterprise Application Architecture*. Springer, Berlin, Heidelberg, pp. 44–55.

- Castelluccia, D., Mongiello, M., Ruta, M., Totaro, R., 2006. WAVer: a model checking- based tool to verify web application design. *Electronic Notes in Theoretical Computer Science* 157 (May (1)), 61–76.
- Choudhary, S.R., Versee, H., Orso, A., 2010. WEBDIFF: automated identification of cross-browser issues in web applications. In: *IEEE International Conference on Software Maintenance*, pp. 1–10.
- Choudhary, S.R., Prasad, M.R., Orso, A., 2012. Crosscheck: combining crawling and differencing to better detect cross-browser incompatibilities in web applications. In: *International Conference on Software Testing, Verification and Validation*, April, pp. 171–180.
- Dallmeier, V., Burger, M., Orth, T., Zeller, A., 2013. WebMate: generating test cases for web 2.0. In: *Software Quality. Increasing Value in Software and Systems Development*, pp. 55–69.
- Di Lucca, G.A., Fasolino, A.R., Tramontana, P., 2006. A technique for reducing user session data sets in web application testing. In: *IEEE International Symposium on Web Site Evolution*, pp. 7–13.
- Di Sciascio, E., Donini, F.M., Mongiello, M., Totaro, R., Castelluccia, D., 2005. Design verification of web applications using symbolic model checking. In: *Web Engineering*. Springer, Berlin, Heidelberg, pp. 69–74.
- Dobolyi, K., Weimer, W., 2010. Modeling consumer-perceived web application fault severities for testing. In: *Proceedings of the International Symposium on Software Testing and Analysis*, p. 97.
- Dobolyi, K., Soechting, E., Weimer, W., 2010. Automating regression testing using web-based application similarities. *International Journal on Software Tools for Technology Transfer* 13 (2), 111–129.
- Elbaum, S., Rothermel, G., Karre, S., li, M.F., 2005. Leveraging user-session data to support web application testing. *IEEE Transactions on Software Engineering* 31 (3), 187–202.
- Ernits, J., Roo, R., Jacky, J., Veanes, M., 2009. Model-based testing of web applications using NModel. In: *Proceedings of the IFIP WG International Conference on Testing of Software and Communication Systems and International FATES Workshop*, vol. 5826, pp. 211–216.
- Ettema, T., Bunch, C., 2010. Eliminating navigation errors in web applications via model checking and runtime enforcement of navigation state machines. In: *Proceedings of the IEEE/ACM International Conference on Automated Software Engineering ACM*, pp. 235–244.

Gerlits, Y., 2010. Testing AJAX functionality with UniTESK. In: Proceedings of the 4th Spring/Summer Young Researchers' Colloquium on Software Engineering, pp. 50–57.

Halfond, W.G., Orso, A., 2006. Command-form coverage for testing database applications. In: 21st IEEE/ACM International Conference on Automated Software Engineering ASE'06, pp. 69–80.

Halfond, W.G.J., Orso, A., 2007. Improving test case generation for web applications using automated interface discovery. In: Proceedings of the Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on The Foundations of Software Engineering, p. 145.

Halfond, W.G.J., Orso, A., 2008. Automated identification of parameter mismatches in web applications. In: Proceedings of the ACM SIGSOFT International Symposium on Foundations of Software Engineering, p. p181.

Halfond, W.G.J., Anand, S., Orso, A., 2009. Precise interface identification to improve testing and analysis of web applications. In: Proceedings of the International Symposium on Software Testing and Analysis, p. 285.

Hao, J., Mendes, E., 2006. Usage-based statistical testing of web applications. In: Proceedings of the International Conference on Web Engineering (ICWE), pp. 17–24.

Harman, M., Alshahwan, N., 2008. Automated session data repair for web application regression testing. In: 2008 International Conference on Software Testing, Verification, and Validation, pp. 298–307.

Hu, W.I.C.C., 2002. Constructing an object-oriented architecture for web application testing. *Journal of Information Science and Engineering* 18 (1), 59–84.

Jensen, S.H., Møller, A., 2011. Modeling the HTML DOM and browser API in static analysis of JavaScript web applications. In: Proceedings of the ACM SIGSOFT Symposium and the European Conference on Foundations of Software Engineering, pp. 59–69.

Kallepalli, C., Tian, J., 2001. Measuring and modeling usage and reliability for statistical web testing. *IEEE Transactions on Software Engineering* 27 (11), 1023–1036. Koopman, P., Achten, P., Plasmeijer, R., 2007. Model-based testing of thin-client web applications and navigation input. *Lecture Notes in Computer Science including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics* 4902 LNCS, 299–315.

Li, N., Xie, T., Jin, M., Liu, C., 2010. Perturbation-based user-input-validation testing of web applications. *Journal of Systems and Software* 83 (November (11)), 2263–2274.

- Licata, D., Krishnamurthi, S., 2004. Verifying interactive web programs. In: Proceedings of the International Conference on Automated Software Engineering, pp. 164–173.
- Liu, C.-H., 2006. Data flow analysis and testing of JSP-based web applications. *Information and Software Technology* 48 (December (12)), 1137–1147.
- Liu, H., Kuan Tan, H.B., 2008. Testing input validation in web applications through automated model recovery. *Journal of Systems and Software* 81 (2), 222–233.
- Liu, C.-H.L.C.-H., Kung, D.C., Hsia, P.H.P., Hsu, C.-T.H.C.-T., 2000. Structural testing of web applications. Proceedings of the International Symposium on Software Reliability Engineering, ISSRE 2000, 84–96.
- Lucca, G.D., Fasolino, A., Faralli, F., 2002. Testing web applications. In: Proceedings of the International Conference on Software Maintenance, pp. 310–319.
- Luo, X., Ping, F., Chen, M.-H., 2009. Clustering and tailoring user session data for testing web applications. In: International Conference on Software Testing Verification and Validation, pp. 336–345.
- Mansour, N., Hourri, M., 2006. Testing web applications. *Information and Software Technology* 48 (January (1)), 31–42.
- Marback, A., Do, H., Ehresmann, N., 2012. An effective regression testing approach for PHP web applications. In: IEEE International Conference on Verification and Validation, pp. 221–230.
- Marchetto, A., 2008. Talking about a mutation-based reverse engineering for web testing: a preliminary experiment. In: International Conference on Software Engineering Research, Management and Applications, pp. 161–168.
- Marchetto, A., Tonella, P., 2010. Using search-based algorithms for Ajax event sequence generation during testing. *Empirical Software Engineering* 16 (December (1)), 103–140.
- Marchetto, A., Ricca, F., Tonella, P., 2007. Empirical validation of a web fault taxonomy and its usage for fault seeding. In: IEEE International Workshop on Web Site Evolution, pp. 31–38.
- Marchetto, A., Ricca, F., Tonella, P., 2008a. A case study-based comparison of web testing techniques applied to AJAX web applications. *International Journal on Software Tools for Technology Transfer* 10 (October (6)), 477–492.
- Marchetto, A., Tonella, P., Ricca, F., 2008b. State-based testing of Ajax web applications. In: International Conference on Software Testing, Verification, and Validation, pp. 121–130.

- Matos, E.C.B., Sousa, T.C., 2010. From formal requirements to automated web testing and prototyping. *Innovations in Systems and Software Engineering 6* (January (1–2)), 163–169.
- Mesbah, A., Prasad, M.R., 2011. Automated cross-browser compatibility testing. *In: International Conference on Software Engineering*, pp. 561–570.
- Mesbah, A., Van Deursen, A., 2009. Invariant-based automatic testing of AJAX user interfaces. *In: Proceedings of IEEE International Conference on Software Engineering*, pp. 210–220.
- Mesbah, A., Van Deursen, A., Roest, D., 2012. Invariant-Based Automatic Testing of Modern Web Applications. *IEEE Transactions on Software Engineering 38* (1), 35–53.
- Minamide, Y., Engineering, D.S., 2005. Static approximation of dynamically generated web pages. *In: Proceedings of International Conference on World Wide Web*, pp. 432–441.
- Mirshokraie, S., Mesbah, A., 2012. JSART: JavaScript assertion-based regression testing. *Web Engineering*, vol. 1. Springer, Berlin, Heidelberg, pp. 238–252.
- Mirshokraie, S., Mesbah, A., Pattabiraman, K., 2013. Efficient JavaScript mutation testing. *In: IEEE International Conference on Software Testing, Verification and Validation*.
- Offutt, J., Wu, Y., 2009. Modeling presentation layers of web applications for testing. *Software & Systems Modeling 9* (2), 257–280.
- Offutt, J., Wu, Y., Du, X., Huang, H., 2004. Bypass testing of web applications. *International Symposium on Software Reliability Engineering*, 187–197.
- Ozkinaci, M., Betin Can, A., 2011. Detecting execution and HTML errors in ASP.Net web applications. *In: Proceedings of the 6th International Conference on Software and Data Technologies*, pp. 172–178.
- Pattabiraman, K., Zorn, B., 2010, November. DoDOM: leveraging DOM invariants for web 2.0 application robustness testing. *In: IEEE International Symposium on Software Reliability Engineering*, pp. 191–200.
- Peng, X., Lu, L., 2011. User-session-based automatic test case generation using GA. *Journal of Physical Sciences 6* (13), 3232–3245.
- Praphamontripong, U., Offutt, J., 2010. Applying mutation testing to web applications. *In: International Conference on Software Testing Verification and Validation Workshops*, pp. 132–141.
- Qi, Y., Kung, D., Wong, E., 2006. An agent-based data-flow testing approach for Web applications. *Information and Software Technology 48* (12), 1159–1171.

- Raffelt, H., Margaria, T., Steffen, B., Merten, M., 2008. Hybrid test of web applications with webtest. In: Proceedings of Workshop on Testing Analysis and Verification of Web Services and Applications, pp. 1–7.
- Ran, L., Dyreson, C., Andrews, A., Bryce, R., Mallery, C., 2009. Building test cases and oracles to automate the testing of web database applications. *Information and Software Technology* 51 (2), 460–477.
- Ricca, F., Tonella, P., 2001. Analysis and testing of Web applications. *Proceedings of the 23rd International Conference on Software Engineering* 47 (6), 25–34.
- Ricca, F., Tonella, P., 2002a. Construction of the system dependence graph for web application slicing. In: Proceedings of the IEEE International Workshop on Source Code Analysis and Manipulation, pp. 123–132.
- Ricca, F., Tonella, P., 2002b. Testing processes of web applications. *Annals of Software Engineering* 14 (1–4), 93–114.
- Roest, D., Mesbah, A., Van Deursen, A., 2010. Regression testing Ajax applications: coping with dynamism. In: International Conference on Software Testing, Verification and Validation, pp. 127–136.
- Sakamoto, K., Tomohiro, K., Hamura, D., Washizaki, H., Fukazawa, Y., 2013. POGen: a test code generator based on template variable coverage in gray-box integration testing for web applications. In: *Fundamental Approaches to Software Engineering*. Springer, Berlin, Heidelberg, pp. 343–358.
- Sampath, S., Bryce, C., 2008. Prioritizing user-session-based test cases for web applications testing. In: *International Conference on Software Testing, Verification, and Validation*, vol. 1, pp. 141–150.
- Sampath, S., Mihaylov, V., Souter, A., Pollock, L., 2004. Composing a framework to automate testing of operational web-based software. In: *Proceedings of the 20th IEEE International Conference on Software Maintenance*, pp. 104–113.
- Sampath, S., Gibson, E., Sprenkle, S., Pollock, L., 2005. Coverage criteria for testing web applications. In: *Computer and Information Sciences*. University of Delaware (Tech. Rep., 2005-017).
- Sampath, S., Sprenkle, S., Gibson, E., Pollock, L., 2006a. Integrating customized test requirements with traditional requirements in web application testing. In: *Proceedings of the Workshop on Testing, Analysis, and Verification of Web Services and Applications*, pp. 23–32.
- Sampath, S., Sprenkle, S., Gibson, E., Pollock, L., 2006b. Web application testing with customized test requirements—an experimental comparison study. In: *International Symposium on Software Reliability Engineering*, pp. 266–278.

- Sampath, S., Sprenkle, S., Gibson, E., Pollock, L., Greenwald, A.S., 2007. Applying concept analysis to user-session-based testing of web applications. *IEEE Transactions on Software Engineering* 33 (10), 643–658.
- Saxena, P., Akhawe, D., Hanna, S., Mao, F., McCamant, S., Song, D., 2010. A symbolic execution framework for JavaScript. In: *IEEE Symposium on Security and Privacy*, pp. 513–528.
- Song, B., Miao, H., 2009. Modeling web applications and generating tests: a combination and interactions-guided approach. In: *IEEE International Symposium on Theoretical Aspects of Software Engineering*, vol. 2007, pp. 174–181.
- Sprenkle, S., Sampath, S.S.S., Gibson, E., Pollock, L., Souter, A., 2005a. An empirical comparison of test suite reduction techniques for user-session-based testing of Web applications. In: *IEEE International Conference on Software Maintenance*, pp. 587–596.
- Sprenkle, S., Gibson, E., Sampath, S., Pollock, L., 2005b. Automated replay and failure detection for web applications. In: *Proceedings of the IEEEACM International Conference on Automated Software Engineering*, pp. 253–262.
- Sprenkle, S., Pollock, L., Esquivel, H., Hazelwood, B., Ecott, S., 2007. Automated Oracle comparators for testing web applications. In: *IEEE International Symposium on Software Reliability*, pp. 117–126.
- Sprenkle, S., Esquivel, H., Hazelwood, B., Pollock, L., 2008. WEBVIZOR: A Visualization Tool for Applying Automated Oracles and Analyzing Test Results of Web Applications , pp. 89–93.
- Sprenkle, S., Pollock, L., Simko, L., 2011. A study of usage-based navigation models and generated abstract test cases for web applications. In: *IEEE International Conference on Software Testing Verification and Validation*, pp. 230–239.
- Sprenkle, S., Cobb, C., Pollock, L., 2012. Leveraging user-privilege classification to customize usage-based statistical models of web applications. In: *IEEE International Conference on Software Testing, Verification and Validation*, pp. 161–170.
- Stepien, B., Peyton, L., Xiong, P., 2008. Framework testing of web applications using TTCN-3. *International Journal on Software Tools for Technology Transfer* 10 (April (4)), 371–381.
- Thummalapenta, S., Lakshmi, K.V., Sinha, S., Sinha, N., Chandra, S., 2013. Guided test generation for web applications. *Proceedings of the International Conference on Software Engineering*, 162–171.

- Tian, J., Ma, L.I., 2006. Web testing for reliability improvement. *Advances in Computers* 67, 178–225.
- Tonella, P., Ricca, F., 2002. Dynamic model extraction and statistical analysis of Web applications. In: *Proceedings International Workshop on Web Site Evolution*, pp. 43–52.
- Tonella, P., Ricca, F., 2004a. A 2-layer model for the white-box testing of Web applications. In: *IEEE International Workshop on Web Site Evolution*, pp. 11–19.
- Tonella, P., Ricca, F., 2004b. Statistical testing of web applications. *Journal of Software Maintenance and Evolution: Research and Practice* 16 (12), 103–127.
- Tonella, P., Ricca, F., 2005. Web application slicing in presence of dynamic code generation. *Automated Software Engineering* 12 (2), 259–288.
- Törsel, A.M., 2013. A tool for web applications using a domain-specific modelling language and the NuSMV model checker. In: *IEEE International Conference on Software Testing, Verification and Validation*.
- Xiong, W., Bajwa, H., Maurer, F., 2005. WIT: A Framework for In-container Testing of Web-Portal Applications 2 Problems Needed to Be Addressed by ICT, vol. 1, no. 403, pp. 87–97.
- Zheng, Y., Bao, T., Zhang, X., Lafayette, W., 2011. Statically locating web application bugs caused by asynchronous calls. *Distribution* 195 (6), 805–814.
- Craig, Rick D. and Jaskiel, Stefan P, 2002. *Systematic Software Testing*. Artech House, Inc. isbn: 1580535089
- Memon, Atif M. and Cohen, Myra B., 2013. Automated Testing of GUI Applications: Models, Tools, and Controlling Flakiness. *Proceedings of the 2013 International Conference on Software Engineering (ICSE '13)*.
- [Eliane Collins](#) ; [Arilo Dias-Neto](#) ; Vicente F. de Lucena Jr., 2012. Strategies for Agile Software Testing Automation: An Industrial Experience. [2012 IEEE 36th Annual Computer Software and Applications Conference Workshops](#)
- R. L. Glass, R. Collard, A. Bertolino, J. Bach, and C. Kaner, 2006. Software testing and industry needs, *IEEE Software*, vol. 23, no. 4, pp. 55–57, 2006.
- A. Bertolino, 2005. Software testing research: Achievements, challenges, dreams, 2005. in *Proceedings of the Workshop on the Future of Software*
- S. Berner, R. Weber, and R. K. Keller, 2005. Observations and lessons learned from automated testing, in *Proceedings of the 27th International Conference on Software Engineering (ICSE 2005)*, 2005, pp. 571–579.

B. Kitchenham and S. Charters, 2007. Guidelines for performing systematic literature reviews in software engineering, Software Engineering Group, School of Computer Science and Mathematics, Keele University, Tech. Rep. EBSE-2007-01, July 2007.

F. Saglietti and F. Pinte, 2010. Automated unit and integration testing for component-based software systems, in Proceedings of the International Workshop on Security and Dependability for Resource Constrained Embedded Systems (D4RCES'10), 2010, pp. 5:1–5:6.

J. A. Dallal, 2007. Automation of object-oriented framework application testing, in Proceedings of the 5th IEEE GCC Conference and Exhibition 2009), 2007, pp. 425–434

R. P. Tan and S. Edwards, 2008. Evaluating automated unit testing in sulu, in Proceedings of the First International Conference on Software Testing, Verification, and Validation (ICST 2008), 2008, pp. 62–71.

M. Alshraideh, 2008. A complete automation of unit testing for javascript programs, Journal of Computer Science, vol. 4, no. 12, pp. 1012–1019, 2008.

M. Malekzadeh and R. Ainon, 2010. An automatic test case generator for testing safety-critical software systems, in Proceedings of the 2nd International Conference on Computer and Automation Engineering (ICCAE), 2010, pp. 136–167.

A. Leitner, I. Ciupa, B. Meyer, and M. Howard, 2007. Reconciling manual and automated testing: The autotest experience, in Proceedings of the 40th Hawaii International International Conference on Systems Science (HICSS-40 2007), 2007, p.261.

S. Kansomkeat and W. Rivepiboon, 2003. Automated-generating test case using uml statechart diagrams, in Proceedings of SAICSIT 2003), 2003, pp. 296–300.

J. Burnim and K. Sen, 2008. Heuristics for scalable dynamic test generation, in Proceedings of the 23rd IEEE/ACM International Conference on Automated Software Engineering (ASE 2008), 2008, pp. 443–446.

B. Haugset and G. K. Hanssen, 2008. Automated acceptance testing: A literature review and an industrial case study, in Proceedings of the Agile Development Conference (AGILE 2008), 2008, pp. 27–38.

C. Liu, 2000. Platform-independent and tool-neutral test descriptions for automated software testing, in Proceedings of the 22nd International Conference on Software Engineering (ICSE 2000), 2000, pp. 713–715.

- D. Hao, L. Zhang, M.-H. Liu, H. Li, and J. Sun, 2009. Test-data generation guided by static defect detection, *J. Comput. Sci. Technol.*, vol. 24, no. 2, pp. 284–293, 2009.
- M. F. Bashir and S. H. K. Banuri, 2008. Automated model based software test data generation system, in *Proceedings of the 4th International Conference on Emerging Technologies (ICET 2008)*, 2008, pp. 275–279.
- L. Shan and H. Zhu, 2009. Generating structurally complex test cases by data mutation: A case study of testing an automated modelling tool, *Computing Journal*, vol. 52, no. 5, pp. 571–588, 2009.
- K. Karhu, T. Repo, O. Taipale, and K. Smolander, 2009. Empirical observations on software testing automation, in *Proceedings of the Second International Conference on Software Testing Verification and Validation (ICST 2009)*, 2009, pp. 201–209.
- T. Wissink and C. Amaro, 2006. Successful test automation for software maintenance, in *Proceedings of the 22nd IEEE International Conference on Software Maintenance (ICSM 2006)*, 2006, pp. 265–266.
- P. Pocatilu, 2002. Automated software testing process, *Economy Informatics*, vol. 1, 2002.
- M. A. Fecko and C. M. Lott, 2002. Lessons learned from automating tests for an operations support system, *Software: Practice and Experience*, vol. 32, no. 15, pp. 1485–1506, 2002.
- R. Coelho, E. Cirilo, U. Kulesza, A. von Staa, A. Rashid, and C. J. P. de Lucena, Jat, 2007. : A test automation framework for multi-agent systems, in *Proceedings of the 23rd IEEE International Conference on Software Maintenance (ICSM 2007)*, 2007, pp. 425–434.
- C. Persson and N. Yilmazturk, 2004. Establishment of automated regression testing at abb: Industrial experience report on 'avoiding the pitfalls', in *Proceedings of the 19th IEEE International Conference on Automated Software Engineering (ASE 2004)*, 2004, pp. 112–121.