

ΜΑΪΟΣ

# ΕΠΑΛΗΘΕΥΣΗ ΚΑΙ ΕΠΙΚΥΡΩΣΗ ΛΟΓΙΣΜΙΚΟΥ

GUI TESTING

Ηλίας Μπεζαντάκος, Ελευθέριος Χατζηαράπης  
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ



## ΕΙΣΑΓΩΓΗ

Πρωτού γίνει αναφορά στο GUI Testing και εμβαθύνουμε στο θέμα αυτό, είναι απαραίτητο να κατανοήσουμε την ευρύτερη έννοια του Software Testing.

Η Δοκιμή Λογισμικού (Software testing) είναι μια διαδικασία κατά την οποία αξιολογείται η λειτουργικότητα μιας εφαρμογής λογισμικού με στόχο να εξεταστεί εάν το λογισμικό αυτό ικανοποιεί τις πραγματικές απαιτήσεις. Αν δεν τις ικανοποιεί, εντοπίζει τα πιθανά errors ,bugs και τυχόν απαιτήσεις που πρέπει να συμπεριληφθούν.

Το Software Testing είναι εξαιρετικά χρήσιμο και απαραίτητο για την ανάπτυξη ενός προϊόντος λογισμικού, καθώς δίνει την δυνατότητα στον developer να εξετάζει το λογισμικό του σε πραγματικές συνθήκες και έτσι να διασφαλίσει την ποιότητα του.

### Software Testing 2

ειδών:

#### 1)Manual Testing

Ο χειροκίνητος τρόπος ελέγχου επιτρέπει σε έναν developer να επαληθεύσει όλες τις απαραίτητες λειτουργίες του λογισμικού και επίσης να πραγματοποιήσει ελέγχους και να τεστάρει το λογισμικό από την σκοπιά του τελικού χρήστη.

#### 2) Automation Testing

Ο αυτοματοποιημένος τρόπος ελέγχου είναι ο βέλτιστος τρόπος για τον εντοπισμό αποκλίσεων μεταξύ των λειτουργιών του σε πραγματικό χρόνο και των απαιτήσεων του. Ο tester εκτελεί ειδικά scripts

ελέγχου από τα οποία παράγονται αυτόματα αποτελέσματα και τα οποία αξιολογούνται. Οι δύο μέθοδοι Software Testing είναι οι εξής:

1)Static Testing (ή Επαλήθευση): Η επαλήθευση είναι μία στατική μέθοδος ελέγχου εγγράφων και αρχείων. Αποτελεί την διαδικασία κατά την οποία επαληθεύονται οι απαιτήσεις που έχει ο developer και επαληθεύεται εάν τελικά η ανάπτυξη του λογισμικού εξελίσσεται σωστά.

2)Dynamic Testing (ή Επικύρωση): Η επικύρωση είναι μία δυναμική μέθοδος ελέγχου του τελικού λογισμικού. Αποτελεί την διαδικασία κατά την οποία επικυρώνεται εάν το λογισμικό που έχει κατασκευαστεί είναι σωστό ή όχι.

## ΤΙ ΕΙΝΑΙ ΤΟ GUI

Πρωτού εξηγήσουμε τι σημαίνει το GUI Testing (Graphical User Interface Testing) οφείλουμε να εξηγήσουμε επακριβώς τη σημασία του όρου GUI.

Υπάρχουν 2 τύποι interface (=διασύνδεση ή διεπαφή ή αλλιώς περιβάλλον στη γλώσσα των ηλεκτρονικών υπολογιστών).

Ο πρώτος είναι το Command Line interface, όπου αποτελεί στην ουσία τη διεπαφή στην οποία πληκτρολογούμε εντολές ( εξ ου και το όνομα Command Line δηλαδή γραμμή εντολών) και αυτές με τη σειρά τους επεξεργάζονται από τον υπολογιστή, ο οποίος και εμφανίζει το αποτέλεσμα αυτής της διαδικασίας μέσω της οθόνης που το συνοδεύει.

Το παράθυρο, το οποίο συνήθως ονομάζεται **γραμμή**

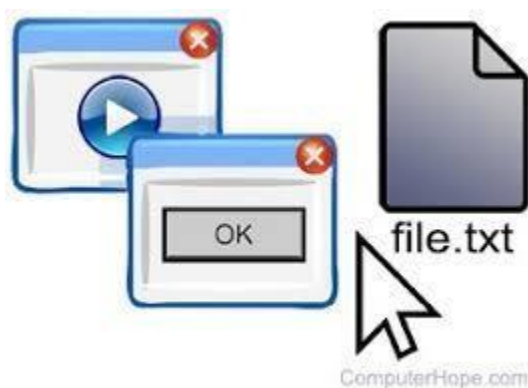
**εντολών ή περιβάλλον γραμμής εντολών**, είναι μια εφαρμογή που βασίζεται σε κείμενο για προβολή, διαχείριση και χειρισμό αρχείων στον υπολογιστή σας. Είναι κάπως σαν το Windows Explorer ή το Finder στα Mac αλλά χωρίς το γραφικό κομμάτι (κουμπιά, βελάκια, σχήματα κλπ).

Ο δεύτερος είναι το GUI, το οποίο αποτελεί πρακτικά την αλληλεπίδραση του χρήστη με τον υπολογιστή μέσω γραφικών όπως εικόνες, ειδικά μενού, κουμπιά, εικονίδια, καθώς και κουτιών διαλόγου, μπαρών μενού ή διάφορων εργαλείων αλλά και παραθύρων κλπ. Άρα, το GUI συνιστά το “συνδετικό κρίκο” του υπολογιστή και του χρήστη μέσω του οποίου αποκτά σημασία και λειτουργικότητα κάθε εντολή. Με αυτόν τον τρόπο λοιπόν, παρέχει στο χρήστη, μέσω εικόνων, ενδείξεις και «εργαλεία» προκειμένου αυτός, με απλές ενέργειες, να εκτελέσει συγκεκριμένες εργασίες. Για τον λόγο αυτό αποδέχονται ενέργειες του χρήστη και “αντιδρούν” ανάλογα στα συμβάντα που αυτός προκαλεί μέσω κάποιας συσκευής εισόδου (π.χ. πληκτρολόγιο, ποντίκι).

Τα περισσότερα σύγχρονα προγράμματα και λειτουργικά συστήματα υπολογιστών, προσφέρουν στους χρήστες τους κάποιο γραφικό περιβάλλον αποτελούμενο από μία ή περισσότερες εικονικές διεπαφές. Αυτός ο τρόπος αλληλεπίδρασης μεταξύ των χρηστών και του υπολογιστή προσεγγίζει την ανθρώπινη εμπειρία στο υπόλοιπο φυσικό περιβάλλον και συνεπώς ταιριάζει καλύτερα στην ανθρώπινη φύση. Το προσδοκώμενο αποτέλεσμα από την

υιοθέτηση ενός σωστά σχεδιασμένου γραφικού περιβάλλοντος είναι να γίνει το περιβάλλον εργασίας στον υπολογιστή όμορφο, εύχρηστο και λειτουργικό.

Πριν από την καθιέρωση των εικονικών διεπαφών η αλληλεπίδραση μεταξύ ενός χρήστη και ενός μικροϋπολογιστή γινόταν μέσω της γραμμής εντολών κάνοντας χρήση κάποιου κελύφους (shell) εντολών. Αυτό προϋπέθετε καλή γνώση των εντολών του συγκεκριμένου κελύφους και των κανόνων σύνταξής τους. Η γνώση αυτή μπορούσε να αποκτηθεί μόνο με την εξοικείωση με το συγκεκριμένο κέλυφος μετά από αρκετή εξάσκηση. Προφανώς, για τη χρήση κάποιου άλλου μικροϋπολογιστή ή κελύφους η διαδικασία εκμάθησης θα έπρεπε ως ένα βαθμό να επαναληφθεί. Το γεγονός αυτό έκανε τους μικροϋπολογιστές δύσχρηστους έως απρόσιτους για το ευρύ κοινό και συνεπαγόταν το να θεωρούνται οι χρήστες τους άτομα με σχετικά υψηλή εξειδίκευση .





\*Παραδείγματα στοιχείων γραφικού περιβάλλοντος\*

## ΤΟ GUI ΣΤΗΝ ΚΑΘΗΜΕΡΙΝΟΤΗΤΑ ΤΟΥ ΣΥΓΧΡΟΝΟΥ ΑΝΘΡΩΠΟΥ

Στη σύγχρονη πραγματικότητα, κάθε άνθρωπος επί του πρακτέου

“συναντά” και αλληλεπιδρά με GUI και μάλιστα αυτό συμβαίνει σε καθημερινή βάση. Το γνωστότερο παράδειγμα μιας διεπαφής γραφικού περιβάλλοντος είναι το λειτουργικό σύστημα των Windows. Ο χρήστης αλληλεπιδρά με τις εφαρμογές μέσω των επιλογών που εμφανίζονται σε παράθυρα. Γραφικά περιβάλλοντα χρησιμοποιούνται επίσης σε πολλές συσκευές όπως κινητά, παιχνιδομηχανές και άλλα. Συνεπώς,

“συναντιέται” καθημερινά από έναν μέσο άνθρωπο της σύγχρονης εποχής και μάλιστα πολλές φορές.

Για παράδειγμα τα λειτουργικά συστήματα όπως Windows, Linux, Mac OS και άλλα έχουν GUI.

Μπορούμε για παράδειγμα να ανοίξουμε έναν φάκελο, απλά κάνοντας ένα κλικ με το ποντίκι επάνω στον σκληρό μας δίσκο, που κι αυτόν τον βλέπουμε σαν γραφικό (εικονίδιο). Το γραφικό περιβάλλον λοιπόν θα μας ανοίξει το φάκελο αυτόν υπό μορφή παραθύρου - γραφικού. Στην ουσία δηλαδή, όσο εμείς αλληλεπιδράμε με το γραφικό περιβάλλον, αυτό τα μεταφράζει από πίσω σε ειδικές εντολές που τις στέλνει στον υπολογιστή μας. Έπειτα ο υπολογιστής μας επιστρέφει αυτά που πρέπει βάσει των εντολών, και το GUI πάλι "τα μεταφράζει" σε κάποιο γραφικό, προβάλλοντάς το σε εμάς.

Στα κινητά, tablets και άλλα μάλιστα είναι η πιο συχνή "συναναστροφή" ανθρώπου και γραφικών περιβάλλοντων. Από το να απαντήσουμε σε μια απλή κλήση μέχρι να κάνουμε διαδικτυακές συναλλαγές. Όλα γίνονται μέσω αλληλεπίδρασης του χρήστη και των γραφικών περιβάλλοντων.

Αξιοσημείωτο είναι ότι υπάρχουν GUI ακόμα και σε άλλου είδους μηχανήματα, όπως μηχανήματα που χρησιμοποιούνται σε τομείς όπως η ιατρική, η αγροτική κτλ. Συνεπώς, είναι προφανές το συμπέρασμα ότι πλέον τέτοια γραφικά περιβάλλοντα αποτελούν αναπόσπαστο κομμάτι ολοένα και περισσότερων ανθρώπων της εποχής μας .

## ΙΣΤΟΡΙΚΗ ΕΞΕΛΙΞΗ GUI



Ας ρίξουμε ωστόσο μια γρήγορη αλλά συνάμα ουσιαστική ματιά στην ιστορική εξέλιξη των γραφικών περιβάλλοντων.

- 1) Ο πρόδρομος των γραφικών περιβάλλοντων δημιουργήθηκε στο Ίδρυμα Ερευνών του πανεπιστημίου Στάνφορντ. Εκεί αναπτύχθηκε η ιδέα της χρήσης υπερσυνδέσμων τοποθετημένων εντός κειμένων, προκειμένου να χρησιμεύσουν στη δημιουργία ενός δικτυακού συστήματος και τους οποίους επεξεργάζονταν μέσω ποντικιού.

Η ιδέα της χρήσης των υπερσυνδέσμων αναπτύχθηκε περαιτέρω και επεκτάθηκε η εφαρμογή τους και εκτός κειμένων σε εικόνες από τους ερευνητές στο σύστημα PARC της *Xerox*. Το PARC αποτελείται από εικονικά στοιχεία όπως «παράθυρα», «μενού», «κουμπιά ραδιοφώνου», «κουτί» ελέγχου και εικονίδια. Το περιβάλλον PARC χρησιμοποιούσε μία συσκευή κατάδειξης σε συνδυασμό με ένα πληκτρολόγιο. Αυτές οι πτυχές του συστήματος συνοψίζονται επιγραμματικά με το ακρωνύμιο WIMP (αγγλικά: *Window, Icon, Menu, Pointer*) το οποίο σημαίνει Παράθυρο, Εικονίδιο, Μενού και Δείκτης.

Οι ερευνητές προχώρησαν παραπέρα από τους υπερσυνδέσμους που βρίσκονταν μόνο εντός κειμένων και δημιούργησαν ένα εικονικό περιβάλλον ως το πρωταρχικό χρηστικό περιβάλλον για το λειτουργικό σύστημα του υπολογιστή Alto της *Xerox*. Τα περισσότερα σύγχρονα εικονικά περιβάλλοντα υπολογιστών γενικής χρήσης προέρχονται από αυτό το σύστημα. Ο Ivan Sutherland ανέπτυξε το 1983 ένα σύστημα

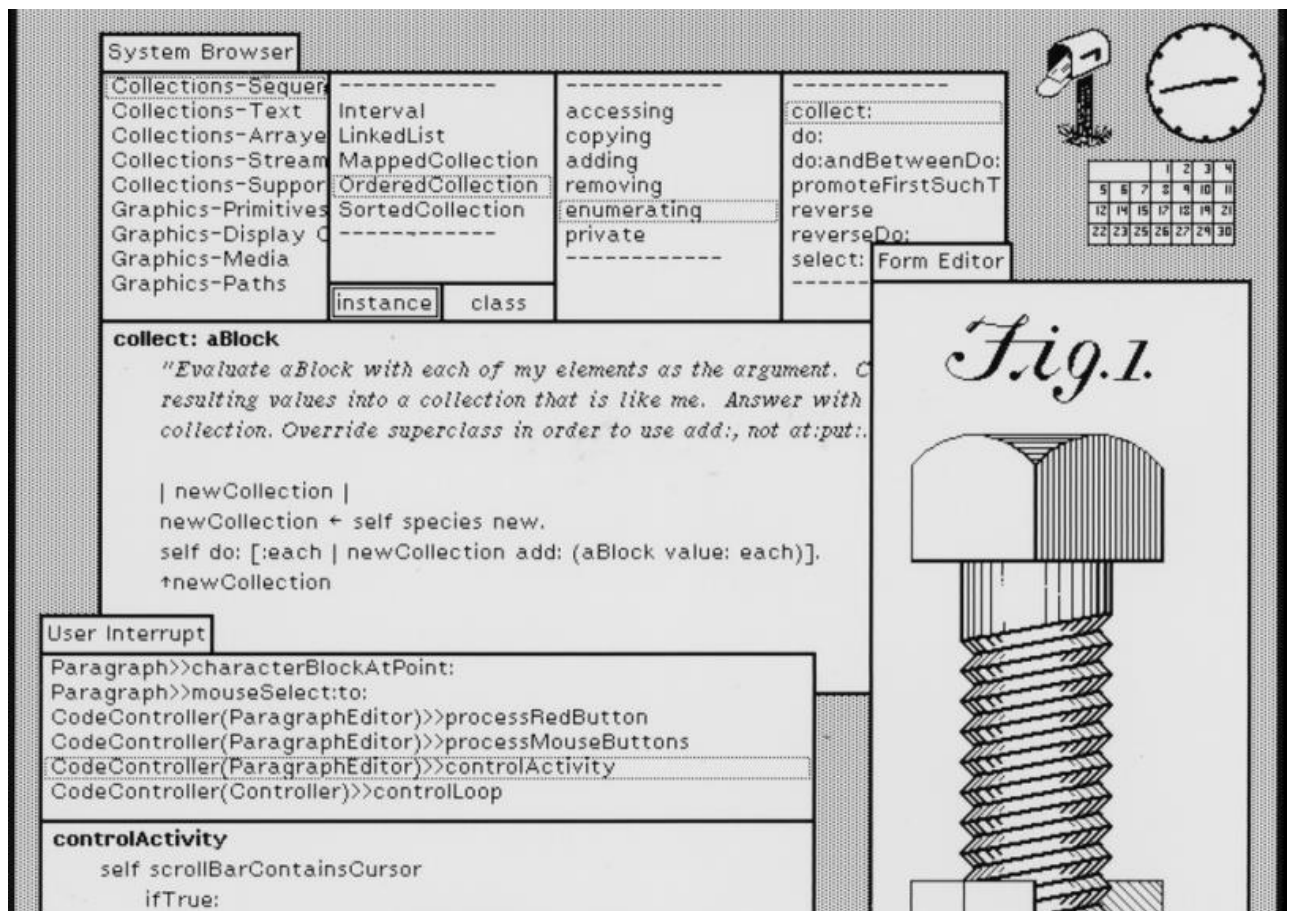
βασιζόμενο σε ένα δείκτη που ονομάστηκε Sketchpad. Χρησιμοποιούσε ένα φωτεινό-στυλό για να βοηθήσει τη δημιουργία και επεξεργασία των αντικειμένων στα σχέδια των μηχανικών.

- 2) Μετά από το PARC, το πρώτο πρότυπο διαχείρισης υπολογιστή με βάση το εικονικό περιβάλλον, ακολουθεί το 1981 το Xerox 8010 Star Information System το οποίο με τη σειρά του ακολουθήθηκε το 1983 από το Apple Lisa στο οποίο πρωτοεμφανίστηκε η γραμμή των μενού όπως επίσης και η ιδέα του «κεντρικού ελέγχου» των παραθύρων, το 1984 από το Apple Macintosh 128K και το 1985 από τα Atari ST και Commodore Amiga.

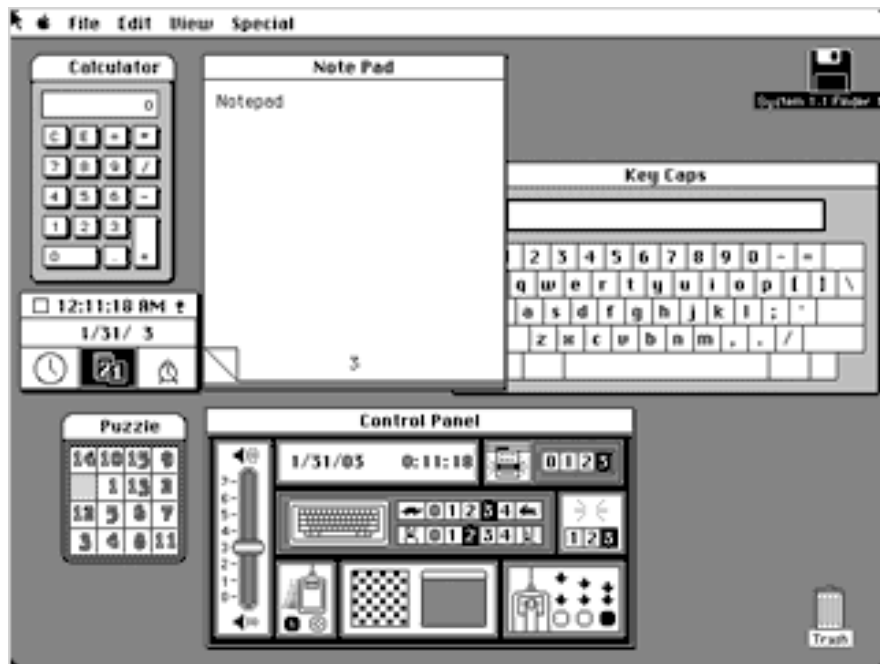
Κάποια εικονικά περιβάλλοντα που μας είναι οικεία στους περισσότερους σήμερα είναι τα περιβάλλοντα για σταθερούς και φορητούς υπολογιστές, όπως τα Microsoft Windows, το Mac OS X και το X Window System, καθώς και αυτά για συσκευές χειρός (π.χ. κινητά τηλέφωνα) όπως το Symbian, το BlackBerry OS, το Android και το iOS της Apple.

Η Apple, η IBM και η Microsoft χρησιμοποίησαν πολλές ιδέες της Xerox για να δημιουργήσουν προϊόντα και οι προδιαγραφές για την κοινή πρόσβαση χρήστη (Common User Access) της IBM απετέλεσαν τη βάση των χρηστικών περιβαλλόντων όπως τα Microsoft Windows, το IBM OS/2 Presentation Manager καθώς και το κουτί εργαλείων και ο διαχειριστής παραθύρων του UnixMotif. Αυτές οι ιδέες εξελίχθηκαν για να δημιουργήσουν το χρηστικό περιβάλλον που συναντάται στις σύγχρονες εκδοχές των Microsoft Windows, Mac OS X και σε ποικίλες επιφάνειες εργασιών σε λειτουργικά

συστήματα τύπου Unix όπως  
το Linux. Έτσι τα πιο σύγχρονα εικονικά  
περιβάλλοντα παρουσιάζουν κοινά χαρακτηριστικά  
σε μεγάλο βαθμό.



\*PARC GUI\*



\*Ένα πρώιμο γραφικό περιβάλλον ( Apple Macintosh 1984)\*

## ΤΑ “ΜΕΡΗ” ΕΝΟΣ GUI

Ένα γραφικό περιβάλλον χρησιμοποιεί ένα συνδυασμό τεχνολογιών και συσκευών για να παράσχει ένα λειτουργικό πλαίσιο με το οποίο ο χρήστης αλληλεπιδρά προκειμένου να συγκεντρώσει και να διαχειριστεί πληροφορίες.

Μια σειρά “εικονογραφικών” στοιχείων που δημιουργήθηκαν με βάση μία “εικονογραφική” γλώσσα προγραμματισμού ( *visual language*) εξελίχθηκαν για να αντιπροσωπεύσουν με ένα δομημένο και ευκολονόητο τρόπο την οργάνωση των πληροφοριών που αποθηκεύονται στους υπολογιστές. Αυτό διευκολύνει χρήστες με λίγες γνώσεις χρήσης υπολογιστών να χρησιμοποιήσουν το λογισμικό ενός

υπολογιστή. Ο πιο συνηθισμένος συνδυασμός αυτών των στοιχείων συνοψίζεται από το σύστημα τύπου WIMP, ειδικά στους προσωπικούς υπολογιστές.

Το εύρος της αλληλεπίδρασης που προσφέρει ένα εικονικό περιβάλλον που εμφανίζεται πάνω στην οθόνη του υπολογιστή με τα χαρακτηριστικά WIMP βασίζεται στη χρήση μίας συσκευής εισόδου η οποία επιτρέπει τον έλεγχο της θέσης ενός δείκτη, επιτρέπει την παρουσίαση με οργανωμένο τρόπο τόσο των πληροφοριών όσο και των λειτουργιών ανάλογα τον τύπο, οργάνωση, ομάδα ή τη δομή τους εντός ξεχωριστών πλαισίων που αποκαλούνται «παράθυρα» και την αντιπροσώπευσή τους με εικονίδια. Οι διαθέσιμες εντολές σε κάθε παράθυρο παρουσιάζονται οργανωμένα υπό τη μορφή μενού. Ο χρήστης μπορεί να μετακινήσει το δείκτη στο εικονίδιο, παράθυρο, μενού ή την εντολή από ένα μενού που τον ενδιαφέρει κάθε φορά. Κατ' αυτόν τον τρόπο μπορεί κατ' αρχήν να επιλέξει από τις εντολές που του παρουσιάζονται διαθέσιμες κάθε φορά. Επίσης μπορεί να επιλέξει από τις ενέργειες που είναι στη διάθεσή του από τα κουμπιά της συσκευής εισόδου κάθε φορά, π.χ. να εμφανίσει στο προσκήνιο το παράθυρο που τον ενδιαφέρει, να μετακινήσει ένα παράθυρο ή εικονίδιο, να αλλάξει το μέγεθός του, να ενεργοποιήσει ένα εικονίδιο κ.α.. Ένα πρόγραμμα-διαχειριστής παραθύρων διευκολύνει τις αλληλεπιδράσεις ανάμεσα στα παράθυρα, τα προγράμματα και το σύστημα παραθύρων. Το σύστημα παραθύρων είναι το πρόγραμμα εκείνο του λειτουργικού συστήματος που διαχειρίζεται αυτόματα τις διάφορες εσωτερικές ή εξωτερικές συσκευές που απαρτίζουν τον υπολογιστή, όπως οι συσκευές

ενδείξεων, το υλικό γραφικών αλλά και την θέση του δείκτη.

Στους προσωπικούς υπολογιστές όλα αυτά τα στοιχεία συνδυάζονται μέσω της “επιφάνειας εργασίας” σκοπός της οποίας είναι η αναπαράσταση στην οθόνη ενός εικονικού περιβάλλοντος που προσομοιάζει την πραγματική επιφάνεια εργασίας ενός γραφείου. Έτσι στην επιφάνεια εργασίας μπορούν να αναπαρασταθούν μέσω εικονιδίων έγγραφα, φάκελοι αλλά και προγράμματα και μπορούν να διευθετηθούν όπως ο χρήστης επιθυμεί. Τα στοιχεία αυτά βέβαια είναι αποθηκευμένα στον υπολογιστή. Τα προγράμματα μπορεί να αντικαθιστούν εντός του εικονικού περιβάλλοντος τη λειτουργία διαφόρων πραγματικών εργαλείων όπως π.χ. ένα ρολόι/ημερολόγιο/ξυπνητήρι, ένα καλάθο αχρήστων εγγράφων κλπ. ή να επιτρέπουν τον έλεγχο μιας πραγματικής συνδεδεμένης εξωτερικής συσκευής όπως π.χ. ένα εκτυπωτή, ένα σαρωτή, μία ψηφιακή φωτογραφική μηχανή ή οποιαδήποτε άλλη ψηφιακή συσκευή καταγραφής ή και αναμετάδοσης όπως ένα τηλέφωνο, ηχείο κλπ.





\*Παράδειγμα επιφάνειας εργασίας και διάφορων άλλων στοιχείων του GUI όπως εικονίδια, μπάρες αναζήτησης καθώς και widgets\*

### 3D GUI

Για τις απλές οθόνες υπολογιστών η απεικόνιση σε τρεις διαστάσεις δεν είναι εφικτή. Ωστόσο η ψευδαίσθηση της τρίτης διάστασης (δηλαδή του βάθους) μπορεί να δημιουργηθεί τοποθετώντας το ένα αντικείμενο μπροστά από το άλλο. Έτσι η αλληλεπίδραση με αυτή είναι εφικτή.

Επίσης η χρήση της σκίασης είναι αυτή που κάνει την τρισδιάστατη εικόνα να δείχνει πιο ρεαλιστική.

Τα γραφικά τριών διαστάσεων χρησιμοποιούνται κυρίως σε ταινίες του κινηματογράφου (π.χ. Άβαταρ)

και σε παιχνίδια υπολογιστών αλλά και φορητών συσκευών πχ κινητά καθώς και σε κονσόλες. Όμως η χρήση τους δεν σταματά εκεί. Πολλές εταιρίες προσπάθησαν να δημιουργήσουν τρισδιάστατη επιφάνεια εργασίας, όπως η SUN με το Looking Glass Project. Επίσης χρησιμοποιείται και στην επιστήμη και την έρευνα, όπως η τρισδιάστατη εικόνα χρησιμοποιείται και για την παρουσίαση των αποτελεσμάτων ενός αξονικού τομογράφου. Τα τρισδιάστατα γραφικά αποτελούν σήμερα το επίκεντρο των παιχνιδιών και πολλών ταινιών και για την υλοποίησή τους χρειάζονται πολλές ώρες δουλειάς από τους ειδικούς , καθώς και μεγάλα μπάτζετ για την ανάπτυξή τους.

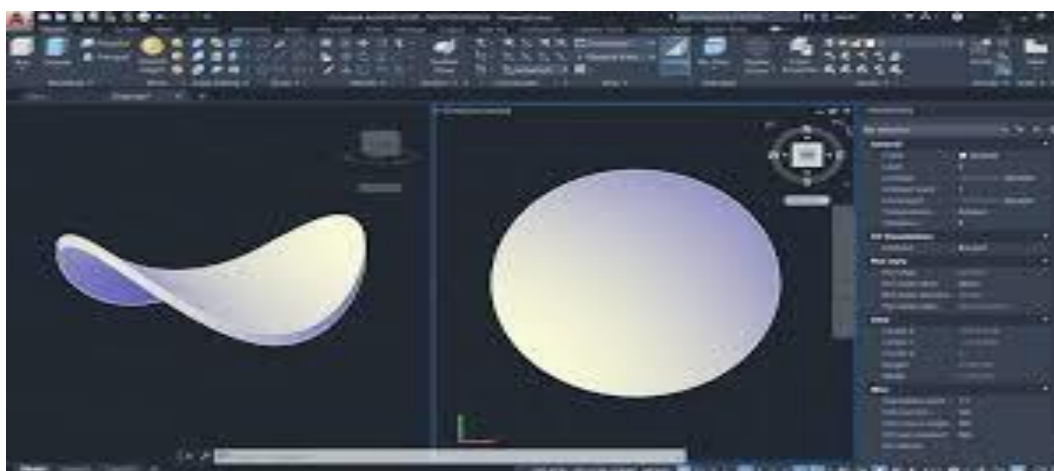


\*Σκηνή από την ταινία Avatar





\*3D γραφικά σε παιχνίδι (Minecraft)



\*3D σχεδιασμός στο AutoCAD\*

## Εικονικά Περιβάλλοντα POST-WIMP

Μικρότερες κινητές συσκευές όπως κατά κύριο λόγο τα smartphones, χρησιμοποιούν τα στοιχεία με διάφορες ενωτικές μεταφορές, εξαιτίας περιορισμών στο χώρο και σε διαθέσιμες συσκευές εισαγωγής.

Εφαρμογές για τις οποίες το WIMP δεν είναι κατάλληλο ίσως χρησιμοποιούν νεότερες τεχνικές αλληλεπίδρασης, οι οποίες είναι γνωστές σαν post-WIMP.

Κάποια λειτουργικά συστήματα που βασίζονται σε οθόνη αφής, όπως το iOS της Apple και το Android, χρησιμοποιούν τα GUI που ονομάζονται post-WIMP. Αυτά υποστηρίζουν είδη αλληλεπίδρασης που χρησιμοποιούν περισσότερα από ένα δάχτυλο στην επαφή με την οθόνη το οποίο επιτρέπει ενέργειες οι οποίες δεν μπορούν να υποστηριχθούν από ένα δείκτη και ποντίκι.

Τα Post-WIMP συμπεριλαμβάνουν διαχειριστές παραθύρων με δυνατότητες 3D compositing όπως το Compiz, το Desktop Window Manager και το LG3D. Κάποια post-WIMP περιβάλλοντα μπορεί να ταιριάζουν καλύτερα σε εφαρμογές οι οποίες συνδυάζουν 3D περιβάλλοντα, όπως το Google Earth.



## Post WIMP Interface

Post WIMP Interface's are found on operating systems used on mobile devices such as:

- # Mobile Phones
- # iPods
- # iPads
- # PDA's

These devices do not have room for a mouse and so they need different ways for the user to select icons and menus etc.

Touch screens are used so that the user's finger takes the place of the mouse and pointer. 'Fingers can be moved in different ways (called gestures) to interact with the operating system:

- # Zoom into images / documents / web pages
- # Rotate images
- # Select icons
- # etc

cs123 INTRODUCTION TO COMPUTER GRAPHICS

## Post WIMP

Then

(Keyboard + Mice) = Graphical User Interface



Now

(Touch + Voice + Gesture = Natural User Interface)

Bend Desk



Samsung Galaxy Gear Watch



Leap Motion



11/29/16

7/71

## **ΣΥΓΚΡΙΣΗ ΤΩΝ ΔΥΟ ΒΑΣΙΚΩΝ ΕΙΔΩΝ ΓΡΑΦΙΚΩΝ ΠΕΡΙΒΑΛΛΟΝΤΩΝ**

Είχαμε αναφερθεί παραπάνω στα 2 είδη interface: το Command Line interface και το GUI και είχαμε αναφέρει τα κύρια χαρακτηριστικά τους. Ας δούμε όμως τώρα τις κύριες διαφορές μεταξύ τους.

- 1) Ένα χρηστικό περιβάλλον με βάση τη γραμμή εντολών απαιτεί τη γνώση πολλών εντολών, οι οποίες είναι συνδυασμός λέξεων και συμβόλων, πράγμα που είναι αρκετά δύσκολο, γιατί η αποτελεσματικότητα και η παραγωγικότητα εξαρτώνται καθαρά από το πόσες τέτοιες εντολές γνωρίζει ο χρήστης. Το να φτάσει ο χρήστης σε ένα υψηλό επίπεδο ικανότητας είναι αρκετά δύσκολο γιατί οι εντολές αυτές θέλουν αρκετό χρόνο για να τις μάθει και δεν είναι ευκολομνημόνευτες. Συχνά η γραμμή εντολών χρησιμοποιείται από τους προγραμματιστές ώστε να μπορούν να επικεντρωθούν περισσότερο στον σκοπό της δουλείας τους και να μην χάνουν χρόνο στον σχεδιασμό των WIMP. Γίνεται φανερό ότι η γραμμή εντολών αποτελεί περιβάλλον που απευθύνεται σε έμπειρους και πολλές φορές εξειδικευμένους χρήστες.

```
C:\> Διαχειριστής: C:\Windows\System32\cmd.exe

Microsoft Windows [Έκδοση 6.1.7600]
Πνευματικά δικαιώματα (c) 2009 Microsoft Corporation. Με επιφύλαξη κάθε νό  
δικαιώματος.

C:\Windows\system32>sfc /scannow

Ξεκινά η σάρωση του συστήματος. Αυτή η διεργασία θα διαρκέσει λίγη ώρα.

Εκκίνηση της φάσης επιβεβαίωσης της σάρωσης συστήματος.
Ολοκληρώθηκε η επιβεβαίωση 100%.

Η Προστασία πόρων των Windows δεν βρήκε παραβιάσεις ακεραιότητας.

C:\Windows\system32>_
```

## \*Χρήση γραμμής εντολών\*

```
Command Prompt

C:\Users\Mark Nicholls\Documents\My Files>dir
Volume in drive C has no label.
Volume Serial Number is 8CBE-2A36

Directory of C:\Users\Mark Nicholls\Documents\My Files

14/09/2012  16:10    <DIR>          .
14/09/2012  16:10    <DIR>          ..
14/09/2012  16:07                0 JUNK.txt
04/09/2012  19:26               75 osoul account.txt
14/03/2012  08:48            10,560 Southern Gothic Horrors.docx
24/02/2012  18:07            10,607 website domain and hosting.docx
11/12/2011  19:43                75 WIN 7 Pro.txt
               5 File(s)            21,317 bytes
               2 Dir(s)  35,198,140,416 bytes free

C:\Users\Mark Nicholls\Documents\My Files>del junk.txt_
```

### Command Line Interface:

Commands are entered at the prompt

```
C:\Users\Mark Nicholls>_
```

## Command Line Interface

A Command Line Interface operating system works via the user entering typed commands with a keyboard. There is no use for a mouse.

Command Line Interfaces do not use windows, icons, menus or pointers.

There are over 270 commands available for functions such as delete, open, run etc.

Commands must be entered precisely with no spelling mistakes. These commands can be difficult to remember and, as a result, this type of interface is considered more difficult to use when compared to a GUI.

In the example on the left I am using a version of MSDOS to delete a file named *JUNK.txt*



2) Αργότερα όταν εμφανίστηκε το GUI, έκανε την χρήση του υπολογιστή πιο εύκολη λόγω της χρήσης των WIMP. Τα WIMP υπερτερούν έναντι της γραμμής εντολών γιατί με τον συνδυασμό των πλήκτρων και του δείκτη μπορεί ο χρήστης να δουλεύει σε πολλά σημεία, ενώ στην γραμμή εντολών δουλεύει μόνο στο τρέχοντα κατάλογο. Στα σύγχρονα πλέον λειτουργικά συστήματα χρησιμοποιούνται και τα δύο περιβάλλοντα, αλλά το GUI λαμβάνει περισσότερη σημασία. Επίσης υπάρχουν και εφαρμογές που έχουν τον συνδυασμό και των δύο περιβαλλόντων.

## **Τάση για GUI Testing.**

Γραφικό περιβάλλον μπορούμε επίσης να βρούμε και σε πάρα πολλές εφαρμογές ενός σύγχρονου υπολογιστή και επίσης σε πάρα πολλές διαδικτυακές εφαρμογές. Τα τελευταία χρόνια όμως υπάρχει μια τάση προς τις διαδικτυακές εφαρμογές, μιας και οι περισσότεροι χρήστες τις προτιμούν. Με λίγα λόγια δημιουργείται μια μεγάλη ζήτηση για ανάλυση, σχεδίαση, υλοποίηση και προβολή ενός γραφικού περιβάλλοντος με το οποίο θα αλληλεπιδρά ο χρήστης. Όσο μεγαλώνει όμως αυτή η ζήτηση άλλο τόσο μεγαλώνει και η απαίτηση για πολύ καλά σχεδιασμένα και αντοχής γραφικά περιβάλλοντα, ικανά να τρέξουν σε ένα μεγάλος εύρος συσκευών.

Όσο μεγάλωνε όμως μια εφαρμογή γινότανε όλο και πιο δύσκολο να υπάρξει ένα συνεχές μοντέλο αξιολόγησης που να ορίζει την καταλληλότητα και τη σωστή υλοποίηση του περιβάλλοντος. Κάποτε ήταν αρκετά εύκολο ένα άτομο ή ομάδα ατόμων να ελέγχουν με διάφορες διαδικασίες τα στοιχεία

γραφικών. Όμως η πολυπλοκότητα αυξήθηκε σε βαθμό που η ανθρώπινη επέμβαση μείωνε κατά πολύ τη ποιότητα ελέγχου.

## **Λεπτομέρειες που αφορούν το Testing.**

Σε μια διαδικτυακή εφαρμογή που αποτελείται από πάρα πολλά στοιχεία όπως είναι κουμπιά, παράθυρα με πληροφορίες , εικόνες κτλ. Χρειάζεται μια πιο πρακτική προσέγγιση στο πρόβλημα του ελέγχου. Πέρα από το κομμάτι της όψης , απαιτείται και ο έλεγχος της λειτουργικότητας των στοιχείων αυτών.

Στο κομμάτι το γραφικό το GUI Testing αξιολογεί τα χαρακτηριστικά στοιχεία που διέπουν τα γραφικά στοιχεία όπως είναι για παράδειγμα αυτά που αναφέρουμε στη παρακάτω λίστα:

- Το χρώμα ενός κειμένου
- Το μέγεθος ενός πεδίου συμπλήρωσης ονόματος
- Το μέγεθος μιας εικόνας κτλ.
- Το μέγεθος γραμματοσειράς
- Λίστες
- Κουμπιά(χρώμα , γραμματοσειρά κειμένου εσωτερικά)
- Και πολλά άλλα.

## **Μεθοδολογίες για Testing**

Μια εφαρμογή διαδικτυακή ή μη περνάει από μια διαδικασία ελέγχου και QA είτε από την ίδια την εταιρία ή μέσω μιας άλλης διότι υπάρχει και αυτό το σενάριο, δηλαδή πολλές εταιρίες παρέχουν τέτοιες υπηρεσίες. Αυτή η διαδικασία είναι αρκετά σημαντική για τη παραγωγή και τη ποιότητα ενός προϊόντος.

Μεθοδολογίες ελέγχου λογισμικού αναφέρονται σε διάφορες στρατηγικές που ορίζουν πως ένα πρόγραμμα ή μια εφαρμογή συναντάει συγκεκριμένα standards και ότι όλες οι λειτουργίες του μπορούν να πραγματοποιηθούν χωρίς κανένα σφάλμα.

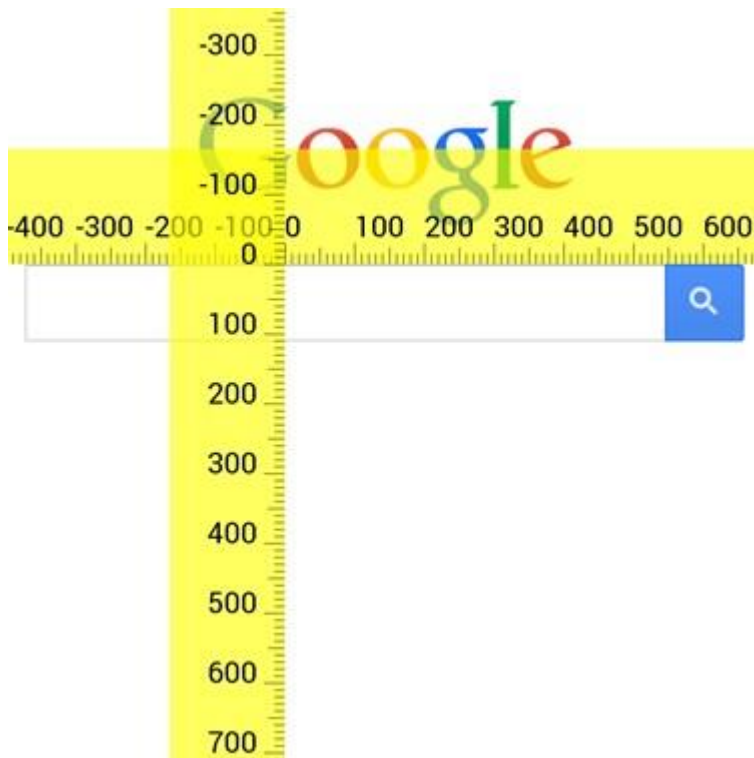
## **Χειρωνακτικός Έλεγχος(Manual)**

Αυτή η διαδικασία ελέγχου συνήθως γίνεται από τον QA (Quality Assurance) . Συνεπάγεται λοιπόν με την ύπαρξη ενός ανθρώπινου παράγοντα που θα εκτελεί ένα σύνολο λειτουργιών και βασικά ελέγχει χειροκίνητα ότι η εφαρμογή συμπεριφέρεται με τον σωστό τρόπο. Αυτή η προσέγγιση έχει αρκετά μειονεκτήματα γτ βασίζεται στον ανθρώπινο παράγοντα που πολλές φορές είναι αναξιόπιστος. Επίσης αποτελεί μια αρκετά χρονοβόρα διαδικασία , απαιτεί αρκετές προσπάθειες και η ποιότητα εξαρτάται σε μεγάλο βαθμό από τις δυνατότητες του QA.

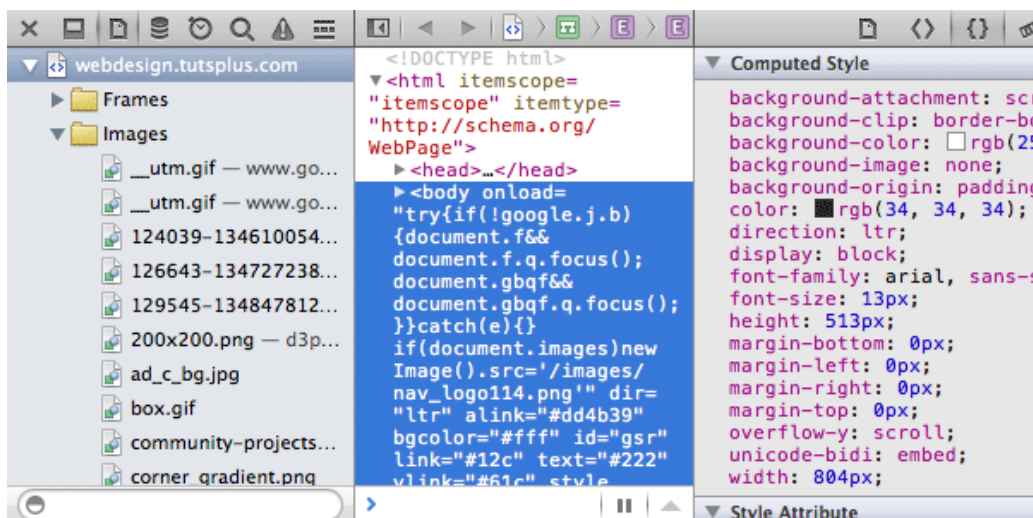
## **Εργαλεία για χειρ/κτικό γραφικό έλεγχο.**

– Page Ruler: Αυτό το εργαλείο μας βοηθάει στο να μετρήσουμε μήκος και πλάτος στα στοιχεία μιας ιστοσελίδας. Αποτελεί ένα βοηθητικό εργαλείο αλλά αποφεύγεται στις περισσότερες περιπτώσεις.



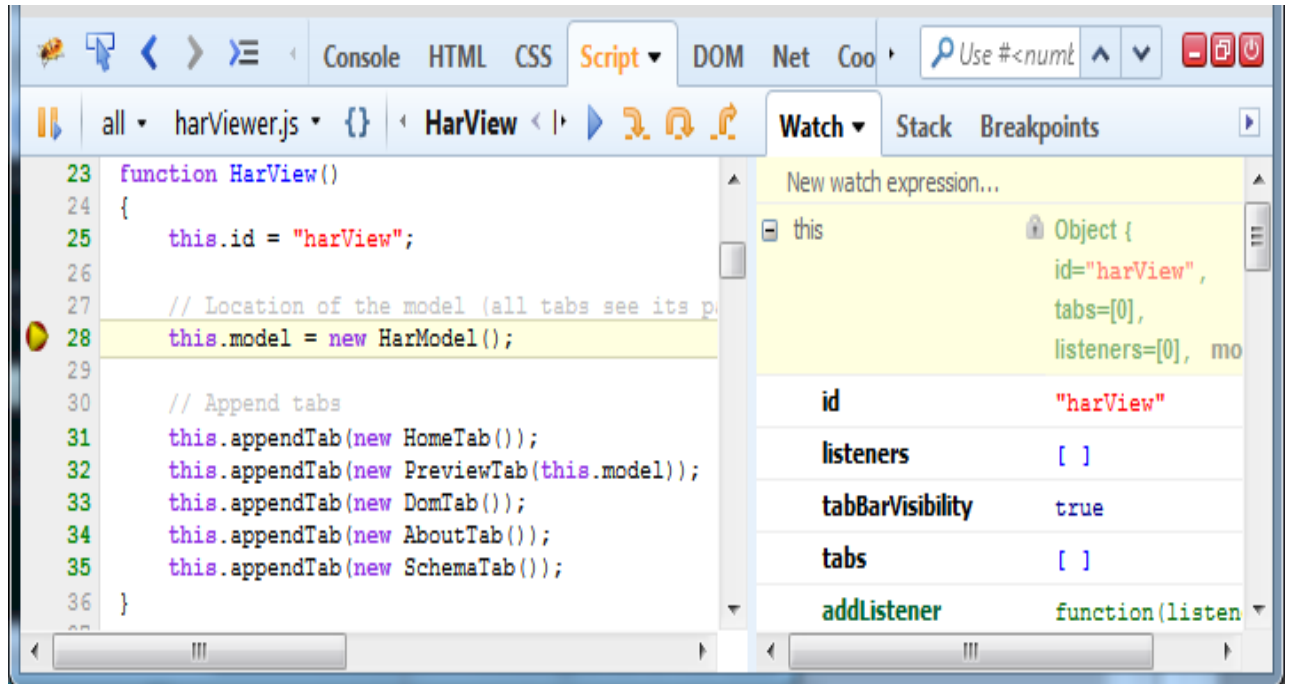


– Web Inspector: Αυτό το εργαλείο αποτελεί ένα από τα σημαντικότερα κομμάτια ενός ελέγχου διότι μας επιτρέπει με ένα διαδραστικό τρόπο να μετρήσουμε, να εξετάσουμε σε μεγάλο βαθμό όλα τα στοιχεία μιας ιστοσελίδας αλλά και να δοκιμάσουμε και τη λειτουργικότητα τους.



### – Firebug:

Αυτό το εργαλείο ανοικτού κώδικα είναι παρόμοιο με το inspect element εργαλείο και είναι μόνο συμβατό με το διακομιστή Firefox. Αποτελεί ένα από τα πολύ καλά εργαλεία για testing και χρησιμοποιείται αρκετά από την αγορά.



## Πρακτικός έλεγχος μιας φόρμας.

Υπάρχουν πολλοί τρόποι, αρκετές στρατηγικές να ακολουθήσει κανείς σχετικά με το πως θα γίνει ο έλεγχος. Μας ενδιαφέρει να καλύπτουν τα βασικά δηλαδή τις ανάγκες του χρήστη και να εκπληρώνουν τους στόχους τους.

Μια βασική στρατηγική ελέγχου για όλα τα παραπάνω στοιχεία είναι η παρακάτω:

1. Δοκιμή των μηνυμάτων σφάλματος που εμφανίζονται.
2. Δοκιμή των διαφόρων τμημάτων της οθόνης.

- 3.Έλεγχος της γραμματοσειράς αν είναι αναγνώσιμη ή όχι.
- 4.Δοκιμή της οθόνης σε διαφορετικές αναλύσεις με τη βοήθεια μεγέθυνσης και σμίκρυνσης όπως 640 x 480, 600x800 κ.λπ.
- 5.Ο έλεγχος της ευθυγράμμισης των κειμένων και άλλων στοιχείων, όπως εικονίδια, κουμπιά, κ.λπ. είναι στη σωστή θέση ή όχι.
- 6.Δοκιμή των χρωμάτων των γραμματοσειρών.
- 7.Δοκιμή των χρωμάτων των μηνυμάτων σφάλματος, των προειδοποιητικών μηνυμάτων.
- 8.Δοκιμή αν η εικόνα έχει καλή σαφήνεια ή όχι.
- 9.Δοκιμή της ευθυγράμμισης των εικόνων.
- 10.Δοκιμή της ορθογραφίας.
- 11.Ο χρήστης δεν πρέπει να απογοητευτεί κατά τη χρήση της διεπαφής συστήματος.
- 12.Δοκιμή αν η διεπαφή είναι ελκυστική ή όχι.
- 13.Δοκιμή των γραμμών κύλισης ανάλογα με το μέγεθος της σελίδας, εάν υπάρχει.
- 14.Έλεγχος των απενεργοποιημένων πεδίων εάν υπάρχουν.

15.Δοκιμή του μεγέθους των εικόνων.

16.Έλεγχος των επικεφαλίδων εάν είναι σωστά ευθυγραμμισμένο ή όχι.

17.Δοκιμή του χρώματος του υπερσυνδέσμου.

## Αυτοματοποιημένος Έλεγχος

Αυτό που συνήθως εφαρμόζεται στις περισσότερες εταιρίες είναι η δοκιμή βάσει μοντέλου. Αφορά κυρίως τη δημιουργία ενός μοντέλου ( μια γραφική αναπαράσταση της συμπεριφοράς του συστήματος) προκειμένου να κατανοηθεί βαθύτερα και να δημιουργηθούν πιο αποτελεσματικές περιπτώσεις δοκιμής. Μετά τον προσδιορισμό των εισόδων(πληροφορία που εισάγει ο χρήστης) και των υπολογισμό των αναμενόμενων εξόδων , εκτελούνται οι δοκιμές. Μετά από αυτό γίνεται η σύγκριση του αποτελέσματος με αυτό που αναμενόταν. Αυτή η προσέγγιση είναι φοβερά αξιόπιστη και αποδοτική αφού βοηθάει σε μεγάλο βαθμό στον καθορισμό καταστάσεων του GUI που δεν είναι επιθυμητές και το αυξάνει επίσης και το επίπεδο αυτοματισμού κάτι που προσφέρει αρκετά θετικά.

## Σημαντικότητα ενός Αυτομ/μενου GUI Testing

Με τον έλεγχο μιας εφαρμογής ή μιας ιστοσελίδας μπορεί να μας εξασφαλίσει αρκετά θετικά. Κάποια από αυτά είναι τα εξής:

→ **Βελτιωμένη ταχύτητα και απόδοση:** Ένα σενάριο μπορεί να εκτελεστεί 24/7 - κάτι που είναι αδύνατο να φανταστεί κανείς όταν πρόκειται για μη αυτόματες δοκιμές. Εν τω μεταξύ, μπορείτε να εμπλέξετε τους δοκιμαστές σε πιο δημιουργικές εργασίες όπου η ανθρώπινη νοημοσύνη είναι ζωτικής σημασίας

- με αυτόν τον τρόπο, ένας διευθυντής μπορεί να βελτιώσει σημαντικά την αποτελεσματικότητα της ομάδας.

→ **Μείωση του περιθωρίου σφάλματος:** Το ανθρώπινο σφάλμα είναι ένας σημαντικός παράγοντας που επηρεάζει τη χειροκίνητη δοκιμή. Η διατήρηση της εστίασης και της συγκέντρωσης κατά τον έλεγχο της απόδοσης των λειτουργιών του ιστότοπου για ώρες είναι σχεδόν αδύνατη - εκεί είναι που οι δοκιμαστές αρχίζουν να κάνουν λάθη. Η αυτοματοποίηση σεναρίων δοκιμών GUI, από την άλλη πλευρά, μειώνει το περιθώριο σφάλματος, επειδή όσο το σενάριο γράφεται σωστά, τα αποτελέσματα των δοκιμών θα συμμορφώνονται με αυτό.

→ **Παρέχει συνέπεια:** Τα αυτοματοποιημένα σενάρια δοκιμών είναι φορητά και επαναχρησιμοποιήσιμα - θα μπορείτε να εφαρμόσετε μια τυπική πρακτική ακόμα κι αν η ομάδα δοκιμών δεν έχει τα παλιά μέλη της. Έτσι, μια τέτοια προσέγγιση παρέχει στον ιδιοκτήτη μιας επιχείρησης συνέπεια και μπορεί να χρησιμοποιηθεί για πολλά προϊόντα.

→ **Πρακτικό στην εύρεση σφαλμάτων παλινδρόμησης:** Οι αυτοματοποιημένες δοκιμές επιτρέπουν στους προγραμματιστές να διευρύνουν την ποικιλία των δοκιμασμένων συμβάντων και να μάθουν εάν μια λειτουργία λειτουργεί σωστά σε όλες τις συνθήκες. Μια τόσο βαθιά εξερεύνηση θα απαιτούσε πολύ περισσότερο χρόνο και εργατικό δυναμικό εάν διεξαχθεί χειροκίνητα.

→ **Εξοικονομεί χρόνο , εργατικό δυναμικό και χρήματα:** Οι αυτοματοποιημένες δοκιμές είναι μια εναλλακτική επιλογή τόσο για νεοσύστατες εταιρείες όσο και για μεγάλες εταιρείες. Τα πρώτα δεν θα χρειαζόταν να διατηρήσουν ένα μεγάλο τμήμα δοκιμών για να φέρουν ένα προϊόν στην αγορά, ενώ το δεύτερο θα ήταν ευκολότερο να κλιμακωθεί σε νέες αγορές εάν όλα τα σενάρια δοκιμών είναι επαναχρησιμοποιήσιμα και μπορούν

εύκολα να τροποποιηθούν για να δοκιμάσουν ένα νέο χαρακτηριστικό.

## Εργαλεία Αυτομ/μενου GUI Testing

1) Watir: αποτελεί μια βιβλιοθήκη ανοικτού κώδικα που αλληλεπιδρά με ένα πρόγραμμα περιήγησης με τον ίδιο τρόπο που κάνουν οι άνθρωποι( κάνοντας κλικ σε συνδέσμους, αυτόματη συμπλήρωση φόρμας και επικύρωση κειμένου). Υποστηρίζει όλους τους browsers και διαθέτει ένα αρκετά δυνατό API.

2) AutoIT: αρκετά ενδιαφέρον εργαλείο το οποίο αυτοματοποιεί εργασίες με τρόπο που διαφέρει από αρκετά άλλα εργαλεία όπως για παράδειγμα τη κίνηση του ποντικιού σε πραγματικό χρόνο που επιλέγει ένα στοιχείο. Είναι μικρό και αυτόνομο και εκτελείται σε όλες τις εκδόσεις των Windows και μόνο.

3) AbbotJava: Το AbbotJava είναι ένα εργαλείο για δοκιμές UI Java που επιτρέπει στους δοκιμαστές να οδηγούν στοιχεία UI τόσο με σενάρια όσο και με μεταγλωττισμένο κώδικα. Είναι ένα λογισμικό μετάβασης για την εξέταση μιας κατάστασης στοιχείων και τη δημιουργία ενεργειών χρήστη. Υποστηρίζονται και η αναπαραγωγή και η εγγραφή εφαρμογών Java.

## Παραδείγματα Αυτ/πημενου Ελέγχου

Υπάρχουν αρκετές περιπτώσεις που απαιτείται η χρήση του αυτοματοποιημένου ελέγχου. Η λίστα των πιθανών σεναρίων είναι βασικά απεριόριστη – ωστόσο εδώ είναι μερικά από τα πιο κοινά.

→ **Δοκιμή σφάλματος τύπου δεδομένων** : Κατά τη διάρκεια αυτού του τύπου δοκιμής θα θέλαμε να διασφαλίσουμε πως

ένας χρήστης μπορεί να συμπληρώσει τη φόρμα μόνο με δεδομένα σε έγκυρη μορφή(κείμενο , ημερομηνία κτλ.)

→ **Δοκιμή γραμμών προόδου (Progress bar)** : Για παράδειγμα εάν χρειάζονται περισσότερα από πέντε δευτερόλεπτα για την ολοκλήρωση μιας διαδικασίας, η σελίδα θα πρέπει να περιέχει μια γραμμή προόδου για να ειδοποιεί έναν χρήστη σχετικά με την κατάσταση φόρτωσης.

→ **Αποθήκευση επιβεβαιώσεων** : Σε περίπτωση που ένας χρήστης θέλει να αλλάξει μια οθόνη αφού έχει εισαγάγει τα δεδομένα σε προηγούμενη, ένα σύστημα θα πρέπει να ενθαρρύνει έναν επισκέπτη να αποθηκεύσει αυτήν την είσοδο.

→ **Έλεγχος ευθυγράμμισης εικόνας** : πραγματοποιείται για να διασφαλιστεί ότι μια εικόνα εμφανίζεται όπως προβάλλεται στη λίστα απαιτήσεων ανεξάρτητα από το πρόγραμμα περιήγησης ή τη συσκευή.

→ **Ορθογραφικός έλεγχος** : στην ουσία εδώ δημιουργούνται περιπτώσεις – σενάρια για γραμματική και ορθογραφικό έλεγχο.

→ **Έλεγχος χρώματος υπερσύνδεσης** : διασφαλίζει ότι όλοι οι υπερσύνδεσμοι επισημαίνονται με μπλε υπογραμμισμένη γραμματοσειρά και διακρίνονται από το μπλοκ κείμενο.

→ **Οδηγίες** : επί της οθόνης - ένας ελεγκτής διασφαλίζει ότι όλες οι οθόνες (εκτός από αυτονόητες) περιέχουν οδηγίες για την καθοδήγηση του χρήστη.



→ **Μη έγκυρη επιλογή** : διασφαλίζει ότι μια οθόνη δεν προσφέρει τις διαθέσιμες επιλογές αυτήν τη στιγμή.

## **Συμπεριφορά και χρησιμότητα**

Όπως είναι γνωστό, ο έλεγχος δε πρέπει να γίνεται μόνο στα γραφικά αλλά και στη λειτουργικότητα. Μια κατευθυντήρια γραμμή το τι θα πρέπει να ελέγχουμε σε θέματα λειτουργικότητας είναι τα εξής:

1. Η εφαρμογή μας ανοίγει με διπλό κλικ.
2. Ελέγχουμε αν μπορούμε να ανοίξουμε πολλαπλά instances της εφαρμογής.
3. Το πρόγραμμα μας εμφανίζει το όνομα, την έκδοση και άλλες σχετικές πληροφορίες.
4. Δυναμικά στοιχεία όπως για παράδειγμα μια μπάρα φόρτωσης πρέπει να υποδεικνύουν με σωστό τρόπο το χρόνο που χρειάζεται μια διεργασία να τελειώσει.
5. Σημαντικά πεδία συμπλήρωσης να χρωματίζονται κτλ.

## **Συμβουλές για επιτυχημένο αυτ/ποιημένο GUI Testing.**

Η δημιουργία ενός πλαισίου δοκιμών αυτοματοποίησης GUI είναι δύσκολη λόγω του τεράστιου εύρους εξαιρέσεων σε κάθε κανόνα που ορίζει ένας επικεφαλής ομάδας. Ωστόσο, όσον αφορά τα πρότυπα και τις βέλτιστες πρακτικές, ακολουθούν οι συμβουλές που ακολουθούν οι δοκιμαστές για να δημιουργήσουν ένα καλά συγχρονισμένο και λειτουργικό πλαίσιο δοκιμών.

→ **Τήρηση προτύπων σχεδιασμού και των αρχών:**  
ανεξάρτητα από το πόσο δημιουργική είναι η διεπαφή ενός



ιστότοπου, πρέπει να συμμορφώνεται με μια λίστα μοτίβων για να μπορεί ο χρήστης να περιηγείται άνετα μεταξύ των σελίδων. Τα μοτίβα σχεδιασμού είναι συγκεκριμένες λύσεις σε ένα καθορισμένο πρόβλημα (όπως το πρότυπο PageObject), ενώ οι αρχές είναι ένα σύνολο οδηγιών και κανόνων που πρέπει να βασίζεστε κατά τη διάρκεια της δοκιμής. Περιλαμβάνουν την αρχή της απλότητας, την αρχή της ορατότητας, την αρχή της δομής, την αρχή της ανατροφοδότησης και ούτω καθεξής. Μόλις ο υπεύθυνος δοκιμών συμμορφωθεί με τα δύο, μπορεί να είναι σίγουρος ότι ένας χρήστης δεν θα ενοχληθεί κατά την πλοήγηση στον ιστότοπο.

→ **Δημιουργία ενός φορητού πλαισίου αυτοματισμού GUI:**

Οι υπεύθυνοι δοκιμών συχνά πρέπει να εκτελούν ελέγχους σε διαφορετικές πλατφόρμες και διακομιστές - σε περίπτωση που ένα πλαίσιο δεν είναι φορητό, μπορεί να χρειαστεί πολύς χρόνος για να μεταφερθεί μια δοκιμή από έναν διακομιστή σε έναν άλλο. Για να αποφύγετε τέτοια προβλήματα, φροντίστε να μην αποθηκεύετε δεδομένα δοκιμών στον τοπικό διακομιστή. Αντ' αυτού, χρησιμοποιήστε έναν εξωτερικό πάροχο αποθήκευσης. Είναι πιθανό να χρειαστείτε ένα πρόγραμμα οδήγησης ιστού για να εκτελέσετε αυτοματοποιημένες δοκιμές GUI - τα καλά νέα είναι ότι υπάρχουν πολλά διαδικτυακά σεμινάρια σχετικά με τον τρόπο εγκατάστασης ενός.

→ **Διασφάλιση της ανεξαρτησίας κάθε δοκιμής:** Οι δοκιμές που χρησιμοποιούν εξαρτήσεις είναι κατά κανόνα δυσκολότερες τόσο για ανάγνωση όσο και για συντήρηση. Προκειμένου να αποφευχθούν προκλήσεις κατά τον παράλληλο αυτοματισμό, είναι καλύτερο να εκτελέσετε κάθε δοκιμή ανεξάρτητα.

→ **Δημιουργία ενός λεπτομερούς συστήματος αναφοράς:** Η ανάλυση των αποτελεσμάτων των δοκιμών είναι ένα κρίσιμο κομμάτι της εργασίας για έναν δοκιμαστή QA. Οι περισσότεροι

προγραμματιστές δημιουργούν αναφορές με τη λεπτομερή περιγραφή ενός αποτελέσματος, του τύπου δοκιμής, της λειτουργικότητας και ούτω καθεξής. Με τα αυτοματοποιημένα εργαλεία θα χρειαζόταν λιγότερος χρόνος για τον έλεγχο όλων των εκτελέσεων.

→ **Ξεχωριστές χειροκίνητες και αυτοματοποιημένες δοκιμές:** Κατά την οργάνωση του κωδικού σας, φροντίστε να διαχωρίσετε τις μη αυτόματες δοκιμές από το αυτοματοποιημένο πλαίσιο δοκιμών. Η δημιουργία μιας ξεχωριστής λειτουργικής μονάδας για ένα αυτοματοποιημένο πλαίσιο δοκιμών GUI σάς επιτρέπει να συνδέσετε το ίδιο σενάριο με διαφορετικά έργα σε περίπτωση που θέλετε να δοκιμάσετε ταυτόχρονα πολλούς ιστότοπους ή εφαρμογές.