

Έλεγχος Λογισμικού

Software Testing

Γενικά...

- **ΣΤΑΤΙΚΟΣ ΕΛΕΓΧΟΣ**: Ο κώδικας **εξετάζεται** για τον εντοπισμό λαθών ή αδυνάμων σημείων
- **ΔΥΝΑΜΙΚΟΣ ΕΛΕΓΧΟΣ**: Ένα συστατικό κώδικα **εκτελείται** με ένα σύνολο από δεδομένα για τα οποία τα αποτελέσματα είναι γνωστά (δοκιμή, test): αποκάλυψη ύπαρξης λαθών και όχι απόδειξη απουσίας λαθών
- Δοκιμαστικά δεδομένα (test data)
- Περιπτώσεις ελέγχου (test cases)

Παράδειγμα...

Έστω ένα πρόγραμμα που δέχεται στην είσοδό του 3 πραγματικούς αριθμούς που αντιπροσωπεύουν μήκη γεωμ. τμημάτων και αποφασίζει αν μπορούν αυτά να είναι οι πλευρές ενός τριγώνου, και τι είδους είναι το τρίγωνο (ισοσκελές, ορθογώνιο, κλπ)

Να δοθεί ένα σύνολο εισόδων που θα εγγυάται την ορθή λειτουργία αυτού του προγράμματος για οποιονδήποτε συνδυασμό 3 τμημάτων

Περιπτώσεις ελέγχου

- 5, 3, 4 \Rightarrow Σκαληνό Τρίγωνο
- 3, 3, 4 \Rightarrow Ισοσκελές Τρίγωνο
- 3, 3, 3 \Rightarrow Ισόπλευρο
- 50, 50, 25 \Rightarrow Ισοσκελές Τρίγωνο
- 8, 2, 5 \Rightarrow Όχι Έγκυρο
- α , 25, 50 \Rightarrow Όχι Έγκυρο
- 10, 0, 100 \Rightarrow Όχι Έγκυρο
- 3, 3, -4 \Rightarrow Όχι Έγκυρο

Βασικές Αρχές Δυναμικού Ελέγχου (1/2)

Γενικά ισχύουν τα ακόλουθα:

- Ο κώδικας ενός προγραμματιστή πρέπει να ελέγχεται από κάποιον άλλο προγραμματιστή
- Ο κώδικας μιας εταιρείας πρέπει να ελέγχεται από άλλη εταιρεία
- Ένα συστατικό κώδικα πρέπει να δοκιμασθεί ώστε να διαπιστωθεί ότι δεν κάνει αυτό που δεν πρέπει να κάνει

Βασικές Αρχές Δυναμικού Ελέγχου (2/2)

- Οι περιπτώσεις ελέγχου καταγράφονται για μελλοντική χρήση
- Μια περίπτωση ελέγχου είναι καλή όταν έχει μεγάλη πιθανότητα ανακάλυψης λαθών
- Μια περίπτωση ελέγχου είναι επιτυχής όταν αποκαλύψει λάθη άγνωστα μέχρι τώρα
- Ο δυναμικός έλεγχος είναι μια εξαιρετικά δημιουργική και διανοητικά προκλητική και δύσκολη εργασία

Επίπεδα Δυναμικού Ελέγχου

- Έλεγχος μονάδας και συνένωσης (φάση κωδικοποίησης)
- Έλεγχος συστήματος και αποδοχής (φάση ελέγχου)

Στρατηγικές Δυναμικού Ελέγχου

- Στρατηγική μαύρου κουτιού (black box testing): είναι γνωστές οι προδιαγραφές του συστατικού αλλά άγνωστος ο τρόπος κατασκευής
- Στρατηγική άσπρου κουτιού (white box testing): είναι γνωστός και ο τρόπος κατασκευής του συστατικού

Στρατηγική Μαύρου Κουτιού

- Λειτουργικός Έλεγχος (Functional Testing)
- Θεωρητικά πρέπει να δοκιμασθούν όλα τα δυνατά δεδομένα εισόδου (έγκυρα και άκυρα)
- Αυτό είναι ανέφικτο οπότε χρησιμοποιούνται μερικές μέθοδοι απλοποίησης :
 - Προσέγγιση Ισοδύναμης Διαμέρισης
 - Προσέγγιση Συνοριακών Τιμών
 - Προσέγγιση Αιτίου - Αποτελέσματος

Ισοδύναμη Διαμέριση (1/4)

- Όλοι οι δυνατοί συνδυασμοί των δεδομένων εισόδου χωρίζονται σε ένα πεπερασμένο σύνολο κλάσεων ισοδύναμων τιμών
- Οι τιμές μιας κλάσης είναι ισοδύναμες με την έννοια ότι ανακαλύπτουν το ίδιο σύνολο λαθών
- Το πλήθος των περιπτώσεων ελέγχου είναι το πολύ ίσο με το πλήθος των κλάσεων

Ισοδύναμη Διαμέριση (2/4)

- Για κάθε περίπτωση ελέγχου απαιτείται μια μόνο τιμή από κάθε κλάση
- Οι κλάσεις προσδιορίζονται από τις συνθήκες εισόδου (περιορισμοί των δεδομένων) εισόδου

Ισοδύναμη Διαμέριση (3/4)

Αν μια συνθήκη προδιαγράφει ένα διάστημα τιμών υπάρχει μια έγκυρη κλάση και δυο άκυρες

π.χ. $1.0 \leq \chi \leq 3.5$ και $\chi < 1.0$, $\chi > 3.5$

Αν μια συνθήκη προδιαγράφει ένα πλήθος τιμών με μια μέγιστη τιμή υπάρχει μια έγκυρη κλάση και δυο άκυρες

π.χ. Αριθμός παιδιών μέχρι 5 :

χ ανήκει στο $[1,2,3,4,5]$ και $\chi = 0$, $\chi > 5$

Ισοδύναμη Διαμέριση (4/4)

- Κάθε σύνολο τιμών, για τις οποίες υπάρχει υπόνοια ότι το συστατικό τις μεταχειρίζεται διαφορετικά, δημιουργεί μια έγκυρη κλάση και μια άκυρη (με μια μόνο τιμή διαφορετική από τις τιμές του συνόλου αυτού)
- Κατά τον ορισμό των περιπτώσεων ελέγχου πρέπει να επιλέγονται εκείνες οι περιπτώσεις που καλύπτουν πολλές ταυτόχρονα κλάσεις: συμπύκνωση δοκιμών (test compaction)

Ανάλυση Συνοριακών Τιμών (1/3)

- Συνοριακές τιμές σε μια κλάση ισοδύναμων τιμών είναι οι τιμές των άκρων της
- Έχει παρατηρηθεί ότι πάνω στις συνοριακές τιμές (τόσο εισόδου όσο και εξόδου) καθώς και στις τιμές αμέσως πριν και μετά γίνονται λάθη

Ανάλυση Συνοριακών Τιμών (2/3)

- Αν μια συνθήκη εισόδου προδιαγράφει ένα διάστημα τιμών απαιτούνται “έγκυρες” περιπτώσεις για τα άκρα του διαστήματος και “άκυρες” για τις τιμές έξω ακριβώς από τα άκρα
- Αν μια συνθήκη εισόδου προδιαγράφει ένα πλήθος τιμών απαιτούνται “έγκυρες” περιπτώσεις με το μικρότερο και μεγαλύτερο πλήθος καθώς και “άκυρες” με ένα μικρότερο και ένα μεγαλύτερο από αυτό το πλήθος

π.χ. Για αρχείο που περιέχει 1 με 200 εγγραφές, απαιτούνται περιπτώσεις ελέγχου με 0, 1, 200, 201 εγγραφές)

Ανάλυση Συνοριακών Τιμών (3/3)

Αν η είσοδος ή η έξοδος είναι ένα διατεταγμένο σύνολο (π.χ. ένα ακολουθιακό αρχείο, μια γραμμική λίστα, ένας πίνακας) απαιτούνται περιπτώσεις ελέγχου με το πρώτο και το τελευταίο στοιχείο ελέγχου

Προσέγγιση Αιτίου - Αποτελέσματος

- Με βάση τις προδιαγραφές κατασκευάζεται ένας γράφος που στους αρχικούς κόμβους έχει αίτια, στους τελικούς κόμβους αποτελέσματα και στους ενδιάμεσους κόμβους περιορισμούς.
- Στη συνέχεια κατασκευάζεται ένας πίνακας απόφασης από τον οποίο μπορούν να παραχθούν περιπτώσεις ελέγχου
- Η εφαρμογή της μεθόδου υποστηρίζεται από εργαλεία (π.χ. SoftTest)

Προδιαγραφή Απαίτησης

Αν ο πελάτης έχει ηλικία μικρότερη των 18 ετών και παίζει τένις, να του αποστέλλεται επιστολή σχετική με προϊόντα τένις. Αν ο πελάτης έχει ηλικία 18 ή μεγαλύτερη ή έχει άδεια οδήγησης αυτοκινήτου, να του αποστέλλεται επιστολή σχετικά με προϊόντα αυτοκινήτου. Αν συμβαίνουν και τα δυο, να προστεθεί το όνομα του στην λίστα Α.

Προδιαγραφή Απαίτηση - Είσοδοι

- Καταστάσεις (true/false):

U18 = The Person is Under 18

O18 = The Person is Over 18

Plays_Tennis = The Person Plays Tennis

Has_Licence = The Person has a Car
Licence

Προδιαγραφή Απαίτηση - Έξοδοι

- Καταστάσεις (true/false):

T_brochure = Send a tennis brochure

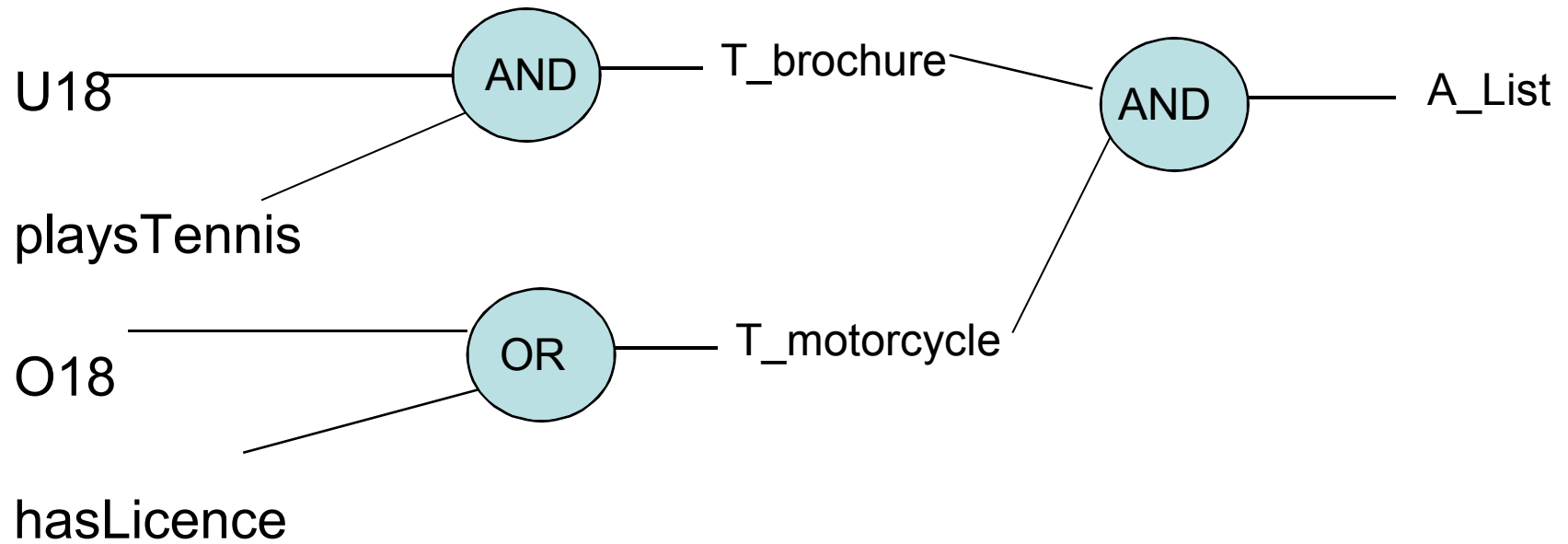
M_brochure = Send a motorcycle brochure

A_List = Put them on the "A" mailing list

Περιορισμοί – Λογικές Σχέσεις

- Περιορισμοί:
Αν $U18=true$ ΤΟΤΕ $O18=false$
Αν $hasLicence=true$ ΤΟΤΕ $O18=true$
- Σχέσεις:
Αν $U18$ ΚΑΙ $playsTennis$ ΤΟΤΕ $T_brochure$
Αν $O18$ 'Η $hasLicence$ ΤΟΤΕ $M_brochure$
Αν $T_brochure$ ΚΑΙ $M_brochure$ ΤΟΤΕ A_LIST

Γράφος Αιτίου - Αποτελέσματος



Δοκιμές

- Αρκούν 3 δοκιμές για την κάλυψη όλων των πιθανών λαθών.
Σύνολο 16 περιπτώσεις χ 3 δοκιμές
- TEST#1
Under18, Plays_tennis, not Over18, not Has_license =>
Send a tennis brochure, Do not send a motorcycle license, Do not put them on the "A" mailing list
- TEST#2
not Under18, Plays_tennis, Over18, not Has_license =>
Do not send a tennis brochure, Send a motorcycle license, Do not put them on the "A" mailing list
- TEST#3
Under18, does not Plays_tennis, not Over18, not Has_license =>
Do not send a tennis brochure, Do not send a motorcycle license, Do not put them on the "A" mailing list

Προσέγγιση Αιτίου - Αποτελέσματος

- Στην πράξη η κατασκευή του γράφου δεν είναι εύκολη και ο αριθμός περιπτώσεων ελέγχου είναι πολύ μεγάλος
- Με τη μέθοδο όμως αυτή είναι δυνατό να ελεγχθούν συνδυασμοί συνθηκών εισόδου
- Ο ελεγκτής πρέπει να χρησιμοποιεί τη διαίσθηση του για να διστανθεί και να ανακαλύψει σχεδιαστικά λάθη

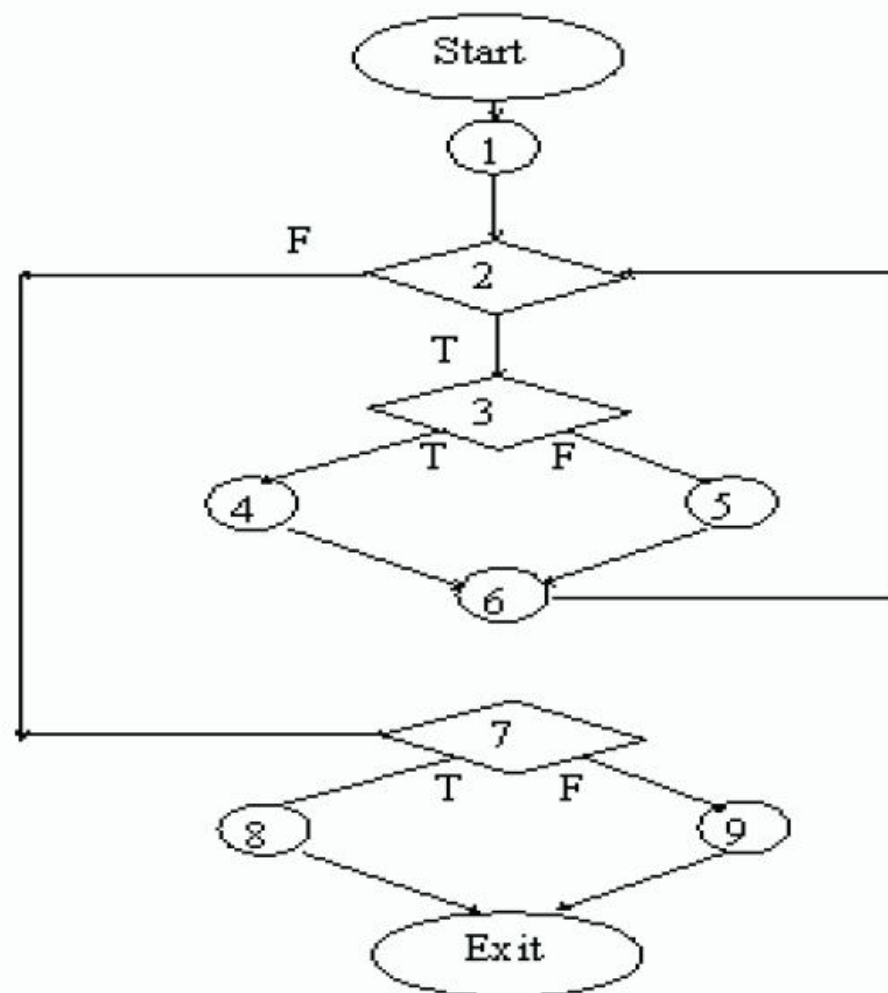
Στρατηγική του Διαφανούς Κουτιού

- Δομικός Έλεγχος (Structural Testing)
- Ο καθορισμός των περιπτώσεων ελέγχου γίνεται με βάση το σχέδιο ή / και τον κώδικα
- Συνήθως επιδιώκεται να καλυφθούν όλα τα δυνατά μονοπάτια εκτέλεσης στο γράφο ροής του προγράμματος

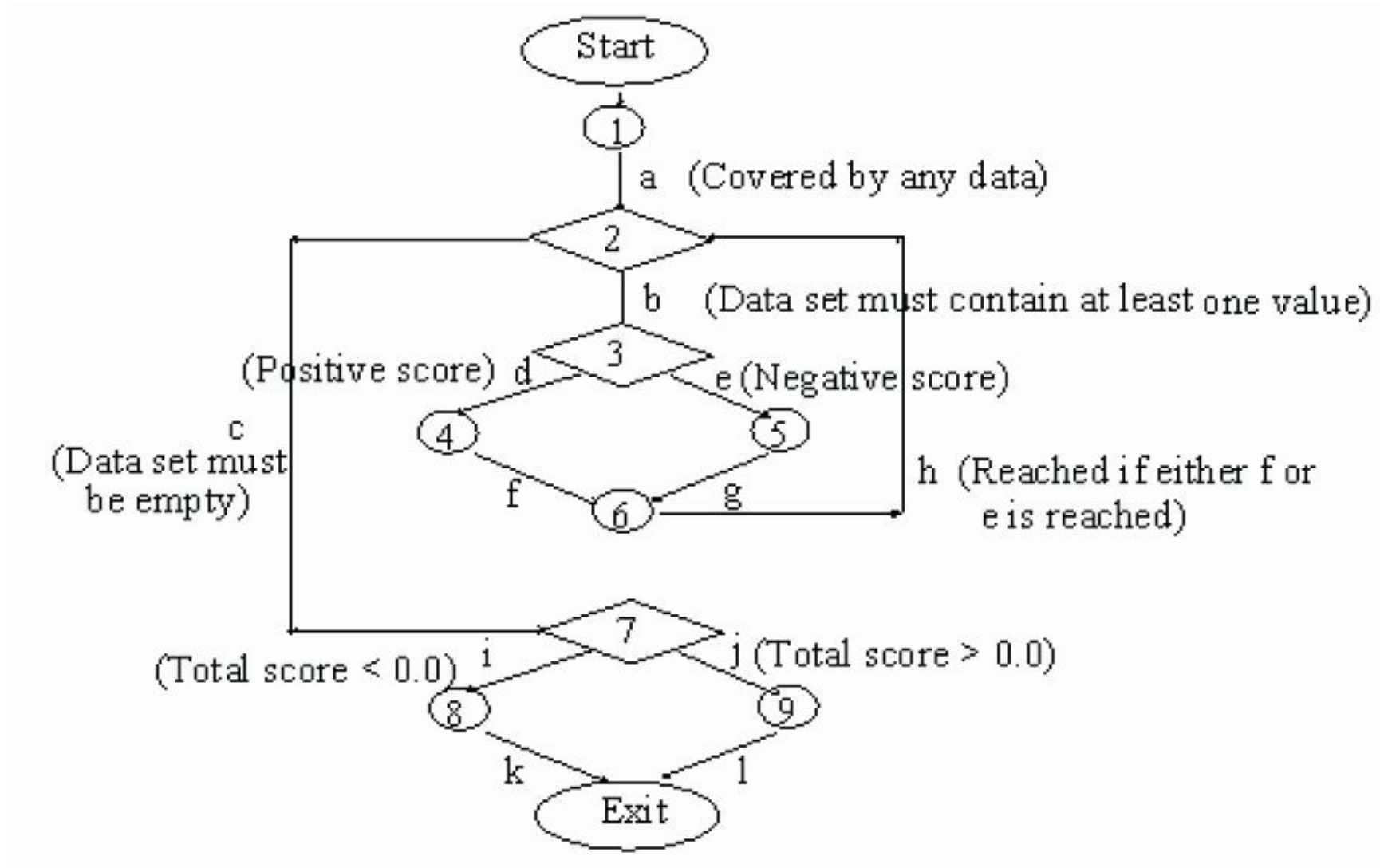
Έλεγχος με βάση τα μονοπάτια

```
FindMean (FILE ScoreFile)
{
  float SumOfScores = 0.0;
  int NumberOfScores = 0;
  float Mean=0.0; float Score;
  Read(ScoreFile, Score);
  ①
  while (! EOF(ScoreFile) {
    ②
    ③ if (Score > 0.0 ) {
      SumOfScores = SumOfScores + Score;
      NumberOfScores++;
    }
    ④
    ⑤
    Read(ScoreFile, Score);
    ⑥
  }
  /* Compute the mean and print the result */
  ⑦ if (NumberOfScores > 0) {
    Mean = SumOfScores / NumberOfScores;
    printf(" The mean score is %f\n", Mean);
    ⑧
  } else
    printf ("No scores found in file\n");
    ⑨
}
```

Γράφος Ροής



Περίπτωσης Δοκιμής



Κυκλωματική Πολυπλοκότητα

- $CC = E - N + 2$

E: ο αριθμός των ακμών

N: ο αριθμός των κόμβων

- Στο προηγούμενο παράδειγμα:

$$CC = 12 - 10 + 2 = 4$$

Προβλήματα

- Ο αριθμός των μονοπατιών είναι συνήθως πολύ μεγάλος
- Δεν καλύπτονται λάθη μη-συμμόρφωσης προς τις προδιαγραφές
- Δεν ανακαλύπτονται μονοπάτια που λείπουν
- Δεν ανακαλύπτονται λάθη που αφορούν ειδικές τιμές δεδομένων

Κριτήρια Κάλυψης (1/2)

- Στη στρατηγική άσπρου κουτιού χρησιμοποιούνται τα λεγόμενα κριτήρια κάλυψης.
- Τα κριτήρια αυτά μας δείχνουν τον βαθμό στον οποίο ο έλεγχος κάλυψε όλα τα σενάρια

Κριτήρια Κάλυψης (2/2)

- Κάλυψη εντολών
- Κάλυψη απόφασης ή κάλυψη διακλάδωσης
- Κάλυψη συνθήκης (συνδυασμοί ελέγχου που επιτρέπουν μια απόφαση)
- Κάλυψη απόφασης/συνθήκης
- Κάλυψη πολλαπλής συνθήκης (πολλαπλές διαδοχικές συνθήκες)

Συμπέρασμα...

Στην πράξη απαιτείται συνδυασμός των δύο στρατηγικών (διαφανούς / αδιαφανούς κουτιού)

Είναι δυνατό να χαραχθεί μια γενική στρατηγική που συνδυάζει τον λειτουργικό και δομικό έλεγχο (όπου αυτό είναι δυνατό)

Εφαρμογή Δυναμικού Ελέγχου (1/3)

Απαιτεί την κατασκευή ορισμένων προϊόντων:

- Ειδικός κώδικας (π.χ. οδηγοί-test drivers)
- Περιπτώσεις ελέγχου
- Ειδικά έγγραφα τεκμηρίωσης

Εφαρμογή Δυναμικού Ελέγχου (2/3)

Έλεγχος μονάδας (Unit Testing)

- Αποσκοπεί στο ναδειχθεί ότι κάθε μονάδα προγράμματος χωριστά δεν πληροί τις προδιαγραφές της
- Ετοιμάζονται περιπτώσεις ελέγχου ξεκινώντας με τη στρατηγική του άσπρου κουτιού που συμπληρώνονται με περιπτώσεις από τη στρατηγική του μαύρου κουτιού

Εφαρμογή Δυναμικού Ελέγχου (3/3)

Δυο είδη ελέγχου μονάδας

- Μη-Αυξητικός (κάθε μονάδα ξεχωριστά και στη συνέχεια συνένωση) –απαιτούνται οδηγοί και στελέχη
- Αυξητικός (κάθε μονάδα ελέγχεται σε συνδυασμό με άλλες μονάδες που έχουν ήδη ελεγχθεί) –απαιτούνται είτε οδηγοί είτε στελέχη ανάλογα με τον τρόπο εφαρμογής του ελέγχου

Αυξητικός Έλεγχος

Η επιλογή των μονάδων μπορεί να γίνει με διάφορα κριτήρια (π.χ. προηγείται ο έλεγχος των μονάδων που κάνουν είσοδο - έξοδο)

Δεν απαιτείται έλεγχος ενοποίησης

Έλεγχος Ενοποίησης (1/2)

- Έλεγχος Ολοκλήρωσης (*Integration Testing*)
- Αποσκοπεί στο ναδειχθεί ότι διαδοχικές ενώσεις μονάδων του προγράμματος σε όλο και μεγαλύτερα συστατικά δεν πληρούν τις προδιαγραφές
- Συχνά συνενώνονται μεγάλες μονάδες λογισμικού που έχουν ελεγχθεί εσωτερικά με αυξητικό ή μη-αυξητικό τρόπο

Έλεγχος Ενοποίησης (2/2)

- Απαιτείται ένα πλάνο ελέγχου συνένωσης που μπορεί να επιτρέψει την παράλληλη διεξαγωγή της κωδικοποίησης, ελέγχου μονάδας και ελέγχου ενοποίησης διαφόρων τμημάτων του συστήματος
- Ο λόγος είναι ότι η υλοποίηση των διαφόρων μονάδων συνήθως ολοκληρώνεται σε διαφορετικές χρονικές στιγμές, συμφωνά με το πλάνο του έργου

Έλεγχος Συστήματος

System Testing

- Αποσκοπεί στο ναδειχθεί ότι το σύστημα δεν πληροί τις απαιτήσεις από το σύστημα και απαιτήσεις από το λογισμικό
- Χρησιμοποιείται η στρατηγική του μαύρου κουτιού και ελέγχονται οι λειτουργικές απαιτήσεις, οι απαιτήσεις διασύνδεσης και οι μη λειτουργικές απαιτήσεις (επίδοσης, περιορισμών στη σχεδίαση, ασφάλειας, αξιοπιστίας, ξεκινήματος ύστερα από αιφνίδιο σταμάτημα, διαθεσιμότητας κτλ)
- Επίσης διενεργούνται έλεγχοι πίεσης (stress tests) και έλεγχοι όγκου δεδομένων (volume tests)

Έλεγχος Αποδοχής

Acceptance Testing

Αποσκοπεί στο να πεισθεί ο πελάτης ότι το σύστημα που κατασκευάστηκε για λογαριασμό του, πληροί τις προδιαγραφές του

Τεκμηρίωση Δυναμικού Ελέγχου

- Τυπικά έγγραφα τεκμηρίωσης ελέγχου

Προαπαιτούμενα έγγραφα:

Έγγραφα έργου

Τεκμηρίωση Μονάδων / Συστατικών προς έλεγχο

Προετοιμασία εκτέλεσης του ελέγχου

Πλάνο Ελέγχου

Προδιαγραφές σχεδίου ελέγχου

Προδιαγραφές περιπτώσεων ελέγχου

Προδιαγραφές διαδικασίας διεκπεραίωσης ελέγχου

Μετά την εκτέλεση του ελέγχου

Ημερολόγιο ελέγχου

Έκθεση απροόπτων

Περίληπτική έκθεση ελέγχου

- Πρότυπο ελέγχου ΙΕΕΕ: 8 έγγραφα τεκμηρίωσης

Δυναμικός Έλεγχος & Παραλαβή (1/2)

- Επαλήθευση

Επαληθευμένο λογισμικό: έχει γίνει έλεγχος συστήματος από την ομάδα που κατασκεύασε το λογισμικό (α -test)

- Επικύρωση

Επικυρωμένο λογισμικό: έχει γίνει έλεγχος συστήματος από ομάδα ανεξάρτητη από εκείνη που το κατασκεύασε (β -test)

Δυναμικός Έλεγχος & Παραλαβή (2/2)

- Πιστοποίηση λογισμικού

Πιστοποιημένο λογισμικό: το λογισμικό έχει περάσει τους προηγούμενους ελέγχους και έχει παραμείνει για ένα διάστημα σε δοκιμαστική λειτουργία στο πραγματικό περιβάλλον λειτουργίας κάτω από την εποπτεία του πελάτη

Αποσφαλμάτωση

Εντοπισμός και εκκαθάριση λαθών

Διάφορες στρατηγικές (Επαγωγή, Αναδρομή, Οπισθοδρόμηση)

Συμβατικές τεχνικές (διαγνωστικά μηνύματα, στιγμιαία αντίγραφα της μνήμης, επιλεκτική λήψη ιχνών δεδομένων και ροής ελέγχου, σημεία διακοπής)

Κριτήρια Τερματισμού (1/2)

Πότε πρέπει να τερματίζει ο έλεγχος;

Όλες οι περιπτώσεις ελέγχου να είναι ανεπιτυχείς και οι περιπτώσεις αυτές να έχουν παραχθεί με βάση το κριτήριο κάλυψης πολλαπλής συνθήκης και να έχει γίνει ανάλυση συνοριακών τιμών στη διασύνδεση της μονάδας

Όλες οι περιπτώσεις ελέγχου να είναι ανεπιτυχείς και οι περιπτώσεις αυτές να έχουν παραχθεί με βάση την προσέγγιση αιτίου-αποτελέσματος και να έχει γίνει ανάλυση συνοριακών τιμών με ταυτόχρονη χρήση της διαίσθησης των ελεγκτών

Κριτήρια Τερματισμού (2/2)

Άλλα κριτήρια μπορεί να είναι:

Αποκάλυψη ενός ορισμένου αριθμού λαθών ή εκπνοή μιας ορισμένης χρονικής περιόδου του ελέγχου, όποιο συμβεί τελευταίο

Ο ρυθμός ανακάλυψης λαθών να μην είναι ανοδικός

Συνδυασμός των παραπάνω