ΚΕΦΑΛΑΙΟ 7

ΔΙΑΣΦΑΛΙΣΗ ΠΟΙΟΤΗΤΑΣ ΛΟΓΙΣΜΙΚΟΥ SOFTWARE QUALITY ASSURANCE (SQA)

Διασφάλιση Ποιότητας Λογισμικού

- Η Διασφάλιση Ποιότητας Λογισμικού (SQA) είναι μια δραστηριότητα που εφαρμόζεται καθόλη τη διάρκεια της διαδικασίας ανάπτυξης Λογισμικού.
- Η Διασφάλιση Ποιότητας Λογισμικού περιλαμβάνει :
 1)Τυπικές τεχνικές επιθεωρήσεις (Formal Technical Reviews) που εφαρμόζονται σε κάθε βήμα ανάπτυξης λογισμικού.
 - 2) Έλεγχο στην τεκμηρίωση του λογισμικού και στις αλλαγές που έχουν γίνει σε αυτό.
 - 3)Μια διαδικασία για την επιβεβαίωση ότι το λογισμικό υπόκειται στις προδιαγραφές που έχουν τεθεί.
 - 4) Μηχανισμούς μετρήσεως και αναφορών.

Ποιότητα Λογισμικού

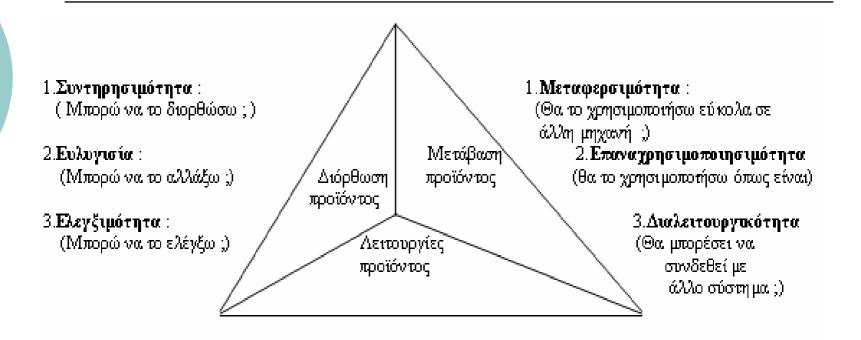
- Κατά τον Pressman η ποιότητα λογισμικού ορίζεται ως εξής:
 Η προσαρμογή του λογισμικού σε λειτουργικές απαιτήσεις και απαιτήσεις εκτέλεσης, σε επεξηγηματικά τεκμηριωμένες προδιαγραφές ανάπτυξης και σε χαρακτηριστικά που εννοούνται και αναμένονται από όλο το λογισμικό που αναπτύσσεται επαγγελματικά.
- Ο παραπάνω ορισμός τονίζει τρία σημεία :
 - **1)**Οι απαιτήσεις λογισμικού είναι η Βάση από την οποία μετριέται η ποιότητα.
 - 2)Κάποιες προκαθορισμένες προδιαγραφές ορίζουν κάποια κριτήρια ανάπτυξης που καθοδηγούν τον τρόπο με τον οποίο το λογισμικό αναπτύσσεται.
 - **3)**Υπάρχει ένα σύνολο **απαιτήσεων που εννοούνται** που πολλές φορές δεν αναφέρονται πουθενά αλλά αναμένονται.
- Η ποιότητα λογισμικού έχει να κάνει με παράγοντες ποιότητας.

Παράγοντες Ποιότητας Λογισμικού

Οι παράγοντες που επηρεάζουν την ποιότητα λογισμικού μπορούν να ταξινομηθούν σε δύο μεγάλες κατηγορίες:

- Παράγοντες που μπορούν να μετρηθούν άμεσα.
 π.χ. λάθη/ΚLΟC
 όπου KLOC=Kilo Lines Of Code
- Παράγοντες που μετριούνται ἐμμεσα.
 π.χ. χρησιμοποιησιμότητα, συντηρησιμότητα κ.λ.π.

- Ο **McCall** και οι συνεργάτες του έχουν προτείνει μια ταξινόμηση παραγόντων που επηρεάζουν την ποιότητα λογισμικού. Όπως φαίνεται και στο παρακάτω σχήμα, οι παράγοντες έχουν να κάνουν με τρείς όψεις του προϊόντος λογισμικού:
 - 1) Τα λειτουργικά χαρακτηριστικά.
 - 2) Την ικανότητα να υποβάλλεται σε αλλαγές.
 - 3) Την προσαρμοστικότητα του σε νέα περιβάλλοντα.



1. Ορθότητα: (Κάνει αυτό που θέλω;)

2. **Αξωπωτία**: (Κάνει αυτό που θέλω με ακρίβεια όλη την

ώρα;)

 $3. A \pi \omega \delta \sigma \tau u \kappa \sigma \tau \eta \tau \alpha$: (Θα τρέξει στο υλικό μου με τον

καλύτερο δυνατό τρόπο ;)

4. Ακεραιότητα: (Είναι ασφαλές:)

5. **Χρησιμοποιησιμότητα**: (Είναι σχεδιασμένο για τον χρήστη:)

 Μερικές από τις περιγραφές που δίνει ο ΜcCall για τους παράγοντες ποιότητας που φαίνονται στο σχήμα είναι :

Ορθότητα: Ο βαθμός στον οποίο το πρόγραμμα ικανοποιεί τους προσδιορισμούς του (specifications) και ικανοποιεί τους στόχους του πελάτη.

Αξιοπιστία: Ο βαθμός στον οποίο μπορεί να περιμένει κανείς το πρόγραμμα να κάνει τις προτιθέμενες λειτουργίες με την απαιτούμενη ακρίβεια.

- Αποδοτικότητα: Η ποσότητα των πόρων του υπολογιστή και του κώδικα που χρειάζεται από ένα πρόγραμμα για να εκτελέσει τη λειτουργία του.
- Ακεραιότητα (Intergrity): Ο βαθμός στον οποίο ελέγχεται η πρόσβαση στο λογισμικό ή τα δεδομένα από ανθρώπους που δεν έχουν άδεια για αυτό.
- Χρησιμοποιησιμότητα: Η προσπάθεια που χρειάζεται για να μάθει ένας χρήστης ένα πρόγραμμα, να το λειτουργεί, να προετοιμάζει τα δεδομένα εισόδου και να ερμηνεύει τα αποτελέσματα εξόδου.
- ο Διαλειτουργικότητα: Η προσπάθεια που χρειάζεται για να λειτουργήσει ένα σύστημα μαζί με ένα άλλο.

Μετρικές(1)

- Είναι δύσκολο (και πολλές φορές ακατόρθωτο) να αναπτυχθούν άμεσα μέτρα των παραγόντων ποιότητας.
- Γι'αυτό χρησιμοποιούνται μια σειρά από μετρικές για να αναπτυχθούν εκφράσεις για κάθε έναν από τους παράγοντες της παρακάτω μορφής:

$$F_q = c_1 * m_1 + c_2 * m_2 + ... + c_n * m_n$$

- όπου F4 είναι ένας παράγων ποιότητας λογισμικού είναι κάποιος συντελεστής και είναι μια **μετρική** που επηρεάζει τον παράγοντα ποιότητας.
- Ο Δυστυχώς πολλές από τις μετρικές του McCall μπορούν να μετρηθούν **υποκειμενικά** μόνο. Οι μετρικές μπορεί να είναι στη μορφή μιας λίστας που χρησιμοποιείται για να «βαθμολογεί» μερικά χαρακτηριστικά του λογισμικού.

Μετρικές(2)

- Οι μετρικές που χρησιμοποιούνται είναι :
- 1) **Εξετασιμότητα** (Auditability):

Η ευκολία με την οποία μπορεί να ελεγχθεί η προσαρμογή στις προδιαγραφές.

- 2) **Ακρίβεια** (Accurancy)
- 3) **Κοινή επικοινωνία** (Communication commonality):

Ο βαθμός στον οποίο χρησιμοποιούνται τα διαδεδομένα συστήματα διεπαφής και πρωτόκολα επικοινωνίας

4) Πληρότητα (Completeness):

Ο βαθμός στον οποίο έχει επιτευχθεί πλήρης υλοποίηση των απαιτούμενων λειτουργιών.

Μετρικές(3)

5) **Συνοπτικότητα** (Conciseness):

Η συμπύκνωση του προγράμματος σε σχέση με τις γραμμές κώδικα.

6) **Συνέπεια** (Consistency):

Η χρήση ομοιόμορφων τεχνικών σχεδιασμού και τεκμηρίωσης.

7) **Κοινά δεδομένα** (Data commonality):

Η χρήση διαδεδομένων δομών δεδομένων και τύπων μέσα στο πρόγραμμα.

8) Ανεκτικότητα λαθών (Error tolerance):

Η ζημιά που προκαλείται όταν το πρόγραμμα αντιμετωπίζει λάθος.

9) Αποδοτικότητα εκτέλεσης:

Η λειτουργία κατά τη διάρκεια εκτέλεσης.

10) **Επεκτασιμότητα** (Expandability):

Ο βαθμός στον οποίο ο σχεδιασμός είναι επεκτάσιμος και κατά συνέπεια το πρόγραμμα.

Μετρικές(4)

- 11) **Γενικότητα** (Generality):
 - Το πλάτος της δυνατής εφαρμογής των προγραμμάτων.
- 12) **Ανεξαρτησία υλικού** (Hardware independence) :
 - Ο βαθμός στον οποίο το λογισμικό δεν δεσμεύεται από το υλικό που χρησιμοποιείται.
- 13) **Ενοργάνωση** (Instrumentation) :
 - Ο βαθμός στον οποίο το πρόγραμμα παρακολουθεί την λειτουργία του και αναγνωρίζει τα λάθη που προκύπτουν.
- 14) **Τμηματικότητα** (Modularity):
 - Η λειτουργική ανεξαρτησία των εξαρτημάτων του προγράμματος.
- 15) **Λειτουργησιμότητα** (Operability) :
 - Η ευκολία λειτουργίας ενός προγράμματος.
- 16) **Ασφάλεια** (Security) :
 - Η ὑπαρξη μηχανισμών που ελέγχουν και προστατεύουν τα προγράμματα και τα δεδομένα.

Μετρικές(5)

- 17) **Αυτο-τεκμηρίωση** (Self-documentation) :
 - Ο βαθμός στον οποίο ο πηγαίος κώδικας παρέχει τεκμηρίωση με νόημα.
- 18) **Απλότητα** (Simplicity) :
 - Ο βαθμός στον οποίο το πρόγραμμα μπορεί να κατανοηθεί χωρίς δυσκολία.
- 19) Ανεξαρτησία συστήματος λογισμικού:
 - Ο βαθμός στον οποίο το πρόγραμμα είναι ανεξάρτητο από χαρακτηριστικά γλωσσών προγραμματισμού που δεν είναι πολύ διαδεδομένα και από περιβαλλοντικές δεσμεύσεις.
- 20) **Ανιχνευσιμότητα** (Traceability):
 - Η ικανότητα ανίχνευσης ενός σχεδιαστικού τμήματος ή ενός μέρους προγράμματος πίσω στις προδιαγραφές.
- 21) **Εκπαίδευση** (Training) :
 - Ο βαθμός στον οποίο το λογισμικό βοηθάει νέους χρήστες να λειτουργήσουν το σύστημα.

Πίνακας Σχέσης Παραγόντων Ποιότητας με Μετρικές

Μετρικές Παράγ ποιότη-	₿ó	1076 (ပောင လည်	έ ρα 1 Ιότ (φηρι σιμό	. 1	1.	ηξιο μότο	φερ μ φερ μ	ησυσχ ησιμο τραισι φτητοι 1	επου η ργικό τ	domot
Εξεπισιμότητας				k			-	*				
Ακρίβεια		x										
Κοινή επικοινωνία											×	
Πληρότητοι	x											
Συνοπτικότητα			×		×	24						
Συνέπεια	×	×		x	x							
Κοινά δεδομένα											×	
Ανεκτικότητο. λοθών		x										
Αποδοτικότητα εκτέθεσης			×									
Επεκικοτιμότητα						×						
Γενικότητα						×		2	ŧ .	*	×	
Ανεζαρτησία υλικού										k	k	
Ευοργάνωση				*	×			×				
Τμηματικότητα		×			×	- 23		x :	k	x	x	
Δειτουργησιμότητα	_		×	ــــــ	_		\perp					×
Ασφάθεια				x								
Αυτο-τεκμηρίωση					×	2	:	x :	k	×		
Απλότητα		×			k	×	3	*				
Ανεζαρτησία Άργισμαρού									ĸ	×		
Ανιχιευσιμότητα	x											
Εκπαίδευση												x

Παράγοντες FURPS(1)

- Η Hewlett Packard έχει αναπτύξει ένα άλλο σύνολο παραγόντων ποιότητας που ονομάζεται FURPS (Functionality, Usability, Reliability, Performance, Supportability).
- Τα χαρακτηριστικά των παραγόντων τα περιγράφουν ως εξής :
 - 1) **Λειτουργικότητα** (Functionality): υπολογίζεται με την αξιολόγηση το σύνολο χαρακτηριστικών και δυνατοτήτων του προγράμματος, την γενικότητα των λειτουργιών που παραδόθηκαν και την ασφάλεια όλου του συστήματος.
 - 2) **Χρησιμοποιησιμότητα** (Usability) : υπολογίζεται λαμβάνοντας υπόψη ανθρώπινους παράγοντες, γενική αισθητική, συνέπεια και τεκμηρίωση.

Παράγοντες FURPS(2)

- 3) **Αξιοπιστία** (Reliability):
 - υπολογίζεται με τη μέτρηση της συχνότητας και σοβαρότητας των αποτυχιών, την ακρίβεια των αποτελεσμάτων, το χρόνο μεταξύ αποτυχιών (Mean Time Between Failure: **MTBF**), η ικανότητα ανάνηψης από αποτυχία και η προβλεψιμότητα του προγράμματος.
- 4) **Εκτέλεση** (Performance) : μετριέται με την αξιολόγηση της ταχύτητας εκτέλεσης, του χρόνου ανταπόκρισης, της κατανάλωσης πόρων και της αποδοτικότητας.
- 5) Υποστηριξιμότητα (Supportability): συνδυάζει την ικανότητα επέκτασης του προγράμματος, την προσαρμοστικότητα.

Τυπικές Τεχνικές Επιθεωρήσεις (Formal Technical Reviews)

- Οι τεχνικές των Τυπικών Επιθεωρήσεων (Formal Reviews) δείχνουν ότι είναι αποτελεσματικές κατά 75% στην ανίχνευση λαθών σχεδιασμού.
- Ο Η έυρεση και απομάκρυνση τέτοιων λαθών κατά τη διάρκεια της επιθεώρησης μειώνει σημαντικά το κόστος των λαθών αυτών στις επόμενες φάσεις.
- Ο Η τυπική τεχνική επιθεώρηση είναι μια δραστηριότητα της Βεβαίωσης Ποιότητας Λογισμικού που εκτελείται από μηχανικούς Λογισμικού.

Στόχοι της δραστηριότητας Βεβαίωσης Ποιότητας Λογισμικού

- 1) Να αποκαλύψει τα λάθη στη λειτουργία, λογική ή υλοποίηση κάποιου τμήματος του λογισμικού.
- 2) Να επαληθεύσει ότι το λογισμικό υπό επιθεώρηση ανταποκρίνεται στις απαιτήσεις.
- 3) Να επιβεβαιώσει ότι το λογισμικό έχει αναπαρασταθεί με βάση τις προαπαιτούμενες προδιαγραφές.
- 4) Να επιβεβαιώσει ότι το λογισμικό αναπτύσεται με ομοιόμορφο τρόπο.
- 5) Να κάνει το έργο πιο εύκολο στη διαχείριση.
- Κάθε Τυπική Τεχνική Επιθεώρηση πραγματοποιείται σε μια σύσκεψη και θα είναι επιτυχημένη μόνο αν έχει προσχεδιασθεί καλά.

Η Σύσκεψη Επιθεώρησης

- Κάθε σύσκεψη επιθεώρησης θα πρέπει να ανταποκρίνεται στους εξής περιορισμούς :
- 1) Στη σύσκεψη θα υπάρχουν μεταξύ 3 και 5 άτομα.
- 2) Η προετοιμασία δεν πρέπει να υπερβαίνει τις 2 με 3 ώρες δουλειάς για κάθε άτομο.
- 3) Η διάρκεια της σύσκεψης δεν πρέπει να είναι μεγαλύτερη από 2 ώρες

Διαδικασία της Σύσκεψης(1)

- Το κύριο μέλημα της σύσκεψης είναι το προϊόν.
- Αυτός που έχει αναπτύξει το προϊόν, δηλαδή ο παραγωγός ενημερώνει τον αρχηγό του έργου (project leader) ότι το προϊόν είναι έτοιμο για επιθεώρηση.
- Ο αρχηγός του έργου έρχεται σε επαφή με έναν αρχηγό επιθεώρησης που αξιολογεί το προϊόν κατά πόσο είναι έτοιμο και μοιράζει το υλικό του προϊόντος σε άλλους 2 ή 3 επιθεωρητές για να προετοιμασθούν.Την ίδια στιγμή ο αρχηγός επιθεώρησης φτιάχνει ένα πλάνο για τη σύσκεψη.

Διαδικασία της Σύσκεψης(2)

- Στη σύσκεψη λαβαίνουν μέρος : ο παραγωγός, ο αρχηγός επιθεώρησης και οι άλλοι επιθεωρητές. Ένας από τους επιθεωρητές κρατάει τα πρακτικά της σύσκεψης.
- ο Στο τέλος της σύσκεψης όλοι πρέπει να αποφασίσουν αν:
 - 1) Πρέπει να αποδεχθούν το προϊόν χωρίς άλλη αλλαγή.
 - 2) Πρέπει να απορρίψουν το προϊόν εξαιτίας σοβαρών λαθών. Σ'αυτή την περίπτωση όταν τα λάθη διορθωθούν θα πρέπει να γίνει καινούρια σύσκεψη.
 - 3) Να αποδεχθούν το προϊόν με την προϋπόθεση ότι θα διορθωθούν κάποια λάθη. Αλλά σ'αυτή την περίπτωση δεν γίνεται άλλη σύσκεψη επιθεώρησης.
- ο Στο τέλος όλοι οι παρευρισκόμενοι πρέπει να υπογράψουν.

Τα πρακτικά της Επιθεώρησης

- Θα πρέπει να υπάρχουν απαντήσεις στα ακόλουθα ερωτήματα :
- 1) Τι επιθεωρήθηκε ;
- 2) Ποιός το επιθεώρησε;
- 3) Τι βρέθηκε και ποιά ήταν τα συμπεράσματα ;

Οδηγίες επιθεώρησης

- 1) Να επιθεωρείται το προϊόν και όχι ο παραγωγός.
- 2) Να τίθεται ένα πλάνο εργασιών και να τηρείται.
- 3) Να περιορίζονται οι διαμάχες.
- 4) Να μην επιχειρείται να λυθεί το **κάθε** πρόβλημα που προκύπτει.
- 5) Να κρατούνται γραπτές σημειώσεις.
- 6) Να υπάρχει προετοιμασία των μελών της σύσκεψης.
- 7) Να γίνεται επιθεώρηση κάποιων προηγούμενων επιθεωρήσεων.

(Για να γίνονται οι διαδικασίες των επιθεωρήσεων πιο αποτελεσματικές).

Η λίστα ελέγχου μιας επιθεώρησης (1)

- ο Οι επιθεωρήσεις γίνονται σε κάθε βήμα:
- Καθορισμός συστήματος :
 - 1) Είναι οι κεντρικές λειτουργίες ορισμένες καλά;
 - 2) Έχουν καθοριστεί οι περιορισμοί σχεδιασμού;
 - 3) Είναι η λύση τεχνολογικά εφικτή;
 - 4) Υπάρχει καθορισμός του μηχανισμού επαλήθευσης και επικύρωσης ;
 - 5) Υπάρχει συνέπεια μεταξύ όλων των στοιχείων του συστήματος ;

Σχεδιασμός Έργου Λογισμικού :

- 1) Είναι η ορολογία σαφής ;
- 2) Είναι οι πόροι αρκετοί για το έργο ;
- 3) Είναι οι πόροι διαθέσιμοι ;
- 4) Έχει γίνει ο προϋπολογισμός χρησιμοποιώντας 2 διαφορετικές μεθόδους ;
- 5) Έχουν συμφιλιωθεί οι διαφορές στους προϋπολογισμούς;
- 6) Είναι οι προθεσμίες ρεαλιστικές ;

Η λίστα ελέγχου μιας επιθεώρησης(2)

Ανάλυση Απαιτήσεων :

- 1) Είναι η ανάλυση του Πεδίου πλήρης, συνεπής και ακριβής ;
- 2) Έχουν καθοριστεί οι εξωτερικές και εσωτερικές διεπαφές;
- 3) Είναι οι απαιτήσεις συνεπείς με το πλάνο, τους πόρους και τον προϋπολογισμό ;
- 4) Υπάρχουν κριτήρια επικύρωσης;

Σχεδιασμός Λογισμικού

- 1) Υπάρχουν όλες οι απαιτήσεις στην αρχιτεκτονική του λογισμικού;
- 2) Υπάρχει καλή τμηματικότητα;
- 3) Έχει ληφθεί υπόψη η συντηρησιμότητα ;
- 4) Έχουν εκτιμηθεί όλοι οι παράγοντες ποιότητας ;
- 5) Οι αλγόριθμοι πετυχαίνουν τις αναμενόμενες λειτουργίες ;
- 6) Είναι οι αλγόριθμοι λογικά σωστοί;
- 7) Έχει καθοριστεί η μέθοδος αποσφαλμάτωσης;
- 8) Είναι οι λεπτομέρειες του σχεδιασμού υλοποιήσιμες;

Η λίστα ελέγχου μιας επιθεώρησης(3)

ο Κωδικοποίηση

- 1) Έχει γίνει καλή μετάφραση του σχεδιασμού σε κώδικα;
- 2) Έχουν χρησιμοποιηθεί καλά οι συμβάσεις της γλώσσας;
- 3) Υπάρχουν λανθασμένα ή ασαφή σχόλια ;
- 4) Είναι οι τύποι δεδομένων σωστοί;

Έλεγχος λογισμικού

- 1) Υπάρχει έγκαιρη επίδειξη των κυριοτέρων λειτουργιών ;
- 2) Υπάρχουν εργαλεία και πόροι ελέγχου ;
- 3) Υπάρχει πλάνο ελέγχου;

ο Συντήρηση

- 1) Έχουν ληφθεί υπόψη τα δευτερεύοντα αποτελέσματα μιας αλλαγής ;
- 2) Έχει η αίτηση για αλλαγή τεκμηριωθεί, αξιολογηθεί και εγκριθεί ;
- 3) Έχει η αλλαγή τεκμηριωθεί και αναφερθεί σε όλα τα ενδιαφερόμενα σημεία ;
- 4) Έχει γίνει μια τελική Επιθεώρηση Αποδοχής των αλλαγών ;