

*ΕΜΠΕΙΡΙΚΗ ΜΕΛΕΤΗ ΑΝΑΣΚΟΠΗΣΗΣ ΤΗΣ
ΕΦΑΡΜΟΓΗΣΤΩΝ Α/Σ ΜΕΤΡΙΚΩΝ ΣΤΙΣ
ΒΙΒΛΙΟΘΗΚΕΣ ΛΟΓΙΣΜΙΚΟΥ (ΑΡΙΣ)*



*ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ
ΡΟΥΜΠΟΣ ΓΙΩΡΓΟΣ*

ΕΙΣΑΓΩΓΗ



Για τις ανάγκες της Πτυχιακής εργασίας θα μελετηθούν επιστημονικές εργασίες που εξετάζουν :

- ☞ Τις Α/Σ μετρικές
- ☞ Τα APIs
- ☞ Εμπειρικές μελέτες που αφορούν την εφαρμογή των μετρικών στην ανάπτυξη των APIs.

Περίγραμμα Πτυχιακής



- ❧ Στο κεφάλαιο 2 γίνεται μια σύντομη αναφορά στην έννοια της ποιότητας λογισμικού, καθώς και ανάλυση της <<κρίσης λογισμικού>>. Γίνεται παρουσίαση των βασικότερων μοντέλων ποιότητας και προτύπων με τα οποία γίνεται ανάπτυξη ποιότητας λογισμικού.
- ❧ Στο κεφάλαιο 3 γίνεται μια πρώτη γνωριμία με τις μετρήσει και τις μετρικές. Στο τέλος του κεφαλαίου παρουσιάζονται οι μέθοδοι διεξαγωγής μετρήσεων ποιότητας λογισμικού.
- ❧ Στο κεφάλαιο 4 ορίζεται η διεπαφή προγραμματισμού εφαρμογών (APIs). Γίνεται ανάλυση ώστε να <<αποκρυπτογραφηθεί>> η λειτουργία της δομής.
- ❧ Στο κεφάλαιο 5 θα αναφερθούν τα σημαντικότερα και δευτερεύοντα εργαλεία μετρικών.
- ❧ Στο κεφάλαιο 6 θα παρουσιαστούν εμπειρικές μελέτες που αφορούν την εφαρμογή των μετρικών στην ανάπτυξη των APIs.

Ποιότητα Λογισμικού



☞ Ορισμός: Η «ποιότητα» είναι ένας όρος πολυδιάστατος και δεν υπάρχει παγκόσμια κοινά αποδεκτός ορισμός για αυτήν, αφού ο κάθε υποψήφιος «πελάτης» έχει ένα διαφορετικό χαρακτηριστικό του προϊόντος / υπηρεσίας το οποία αποκαλεί «ποιότητα». Παρ' όλα αυτά, το κοινό σημείο όλων των ορισμών είναι ότι η ποιότητα συνεπάγεται την ικανοποίηση των αναγκών του πελάτη (Ζαβλανός, 2003).

ISO



Ο Διεθνής Οργανισμός Τυποποίησης ISO έχει υιοθετήσει μια δική του ερμηνεία για την ποιότητα η οποία αναφέρει ως «ποιότητα είναι η ολικότητα των στοιχείων και χαρακτηριστικών ενός προϊόντος ή υπηρεσίας για την εξυπηρέτηση μιας δοσμένης ανάγκης».

Συνοπτικά η ποιότητα έχει οριστεί ως:

1. Συμμόρφωση στις προσδοκίες των ανθρώπων
2. Συμμόρφωση με τις απαιτήσεις των ανθρώπων
3. Αποφυγή της ζημιάς
4. Ανταπόκριση στις προσδοκίες των πελατών
5. Υπεροχή και προστιθέμενη αξία.

Τομείς που θεωρούνται πρωτόποροι

☞ Στρατιωτικός

☞ Διαστημικός

Δραστηριοποιούμενοι στα συστήματα διασφάλισης ποιότητας. Το επόμενο στάδιο, το οποίο έχει υιοθετηθεί από πολλές επιχειρήσεις και οργανισμούς είναι «Η Ολική Ποιότητα» και η «Διοίκηση Ολικής Ποιότητας» είναι το σύνολο των δραστηριοτήτων που υπόσχεται ποιοτικά προϊόντα με ταυτόχρονη μείωση κόστους, πλήρη αξιοποίηση και ανάπτυξη του διαθέσιμου προσωπικού, εφαρμογή καινοτομιών, συνεχή βελτίωση και πλήρη συμμετοχή στην προσπάθεια όλων των επιπέδων των εργαζομένων.

ISO/IEC 9126-1



- ❧ Ποιότητα Λογισμικού: «Η ικανότητα ενός προϊόντος λογισμικού να επιτρέπει σε συγκεκριμένους επισκέπτες να επιτύχουν συγκεκριμένους στόχους με αποδοτικότητα, παραγωγικότητα, ασφάλεια και ικανοποίηση σε συγκεκριμένα περιβάλλοντα χρήσης του εν λόγω προϊόντος»
- ❧ Για τη διασφάλιση καθορίζει τα χαρακτηριστικά και τα υπό-χαρακτηριστικά της ποιότητας καθώς και τις συνδεδεμένες μετρικές.
 - i. Εξωτερικές μετρικές
 - ii. Εσωτερικές μετρικές
 - iii. Μετρικές σε χρήση

Μοντέλο Εξωτερικής/Εσωτερικής Ποιότητας

του

προτύπου ISO/IEC 9126-1





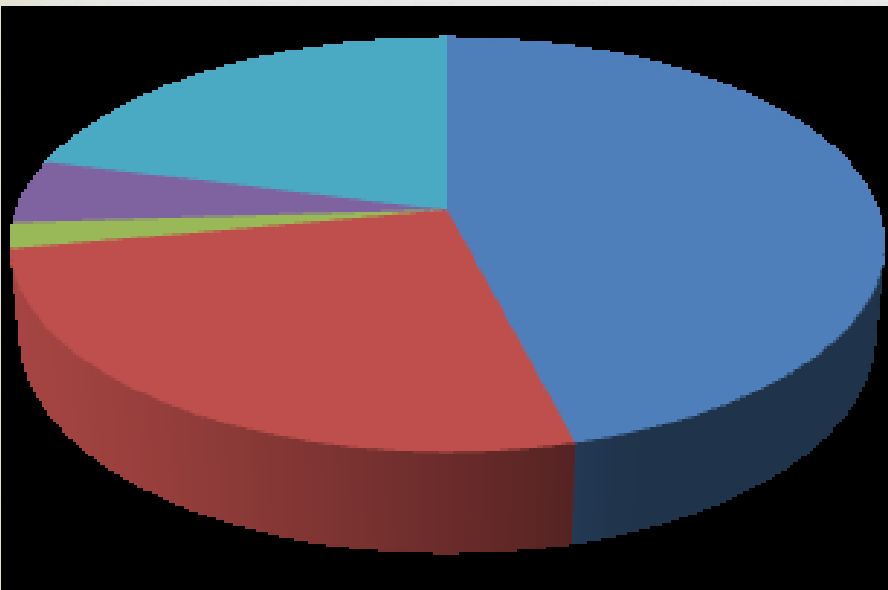
Τρία από τα σημαντικότερα μοντέλα ποιότητας λογισμικού που χρησιμοποιούνται μέχρι σήμερα είναι :

- a) Το μοντέλο του McCall (ή αλλιώς μοντέλο FCM),
- b) Το μοντέλο του Boehm
- c) Το πρότυπο ISO 9126-1.

Κρίση του λογισμικού



Αποτελεσματικότητα των προϊόντων λογισμικού



- Παραδόθηκαν αλλά δεν χρησιμοποιούνται (46%)
- Πληρώθηκαν αλλά δεν παραδόθηκαν πλήρως (26%)
- Χρησιμοποιήθηκαν όπως ακριβώς παραδόθηκαν (2%)
- Χρησιμοποιήθηκαν όπως ακριβώς παραδόθηκαν (5%)
- Εγκαταλείφθηκαν ή Επανασχεδιάστηκαν (21%)

Κρίση του λογισμικού

(De Marco)



Η κρίση του λογισμικού περιορίζεται με την πρόταση του De Marco ο οποίος, πρότεινε στις εταιρίες λογισμικού να

- i. να υλοποιούν λιγότερο κώδικα,
- ii. να επιλέγουν πολύ προσεκτικότερα τι ακριβώς υλοποιούν
- iii. να θέτουν ελαστικότερα όρια ολοκλήρωσης και παράδοσης των έργων τους (DeMarco, 1999).

Με την πάροδο των ετών παρέμενε και χαρακτηρίστηκε ως «χρόνια πάθηση»

Κρίση του λογισμικού



Σήμερα, η κρίση του λογισμικού έχει αντιμετωπιστεί, καθώς δημιουργήθηκαν πρότυπα, που καθορίζουν πως πρέπει να εφαρμοστεί ένα πρόγραμμα εξασφάλισης ποιότητας στη διαδικασία παραγωγής.

Πρότυπα

- ISO90004 ((ISO/IEC, 2000))
- Πρότυπα IEEE5 ((IEEE, 1989))
- Βραβεία Baldrige6 ((Brown, 1991))
- Capability Maturity Model (CMM)7
- Capability Maturity Model Integration (CMMI)8 ((Ahern, et al., 2004))

Μοντέλα ποιότητας λογισμικού και πρότυπα



Οι παράγοντες ποιότητας είναι οι παρακάτω:

- **Functionality** (λειτουργικότητα)
- **Usability** (ευχρηστία).
- **Correctness** (ορθότητα).
- **Integrity** (ακεραιότητα).
- **Maintainability** (συντηρησιμότητα).
- **Testability** (ελεγκσιμότητα).
- **Reusability** (επαναχρησιμοποίηση).
- **Portability** (μεταφερσιμότητα).
- **Efficiency** (αποδοτικότητα).
- **Reliability** (αξιοπιστία)

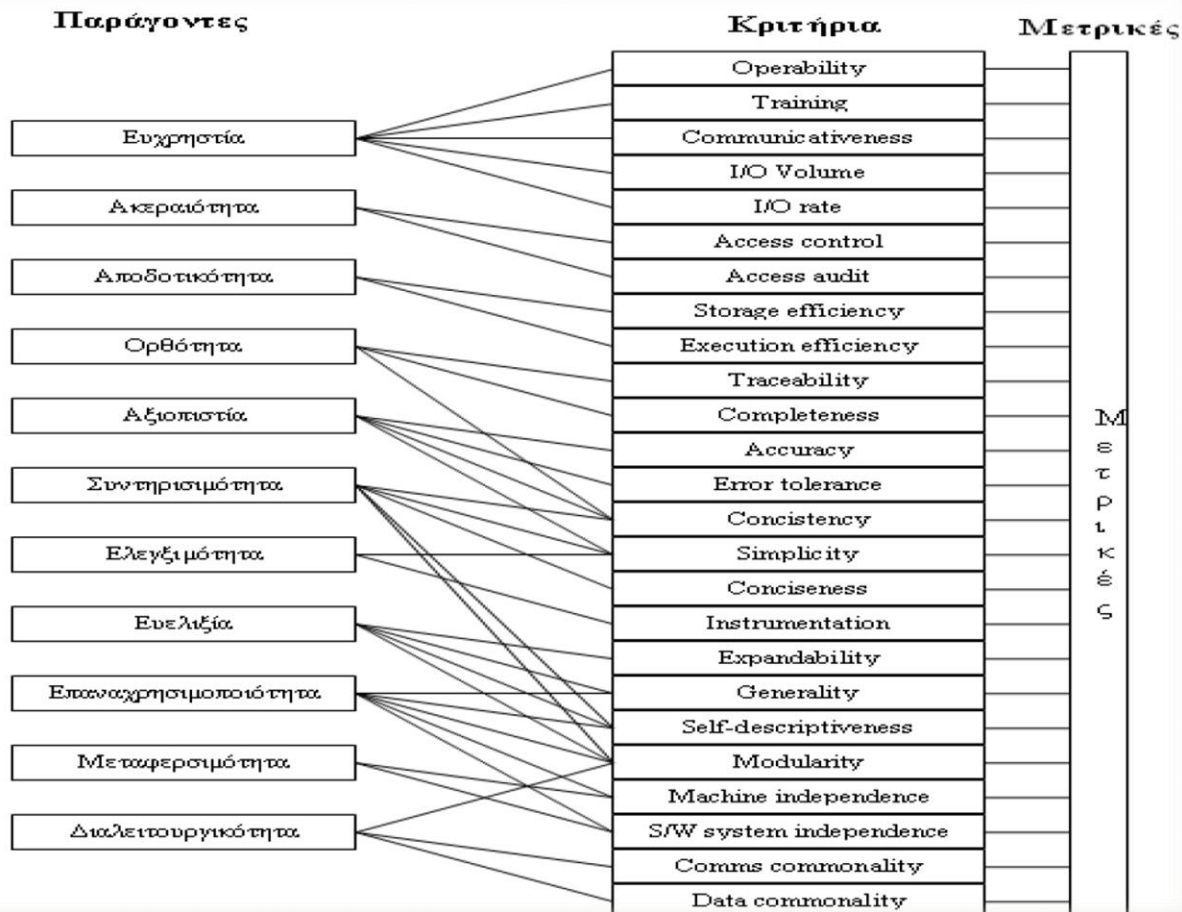
Το Μοντέλο του McCall

(ή μοντέλο FCM)



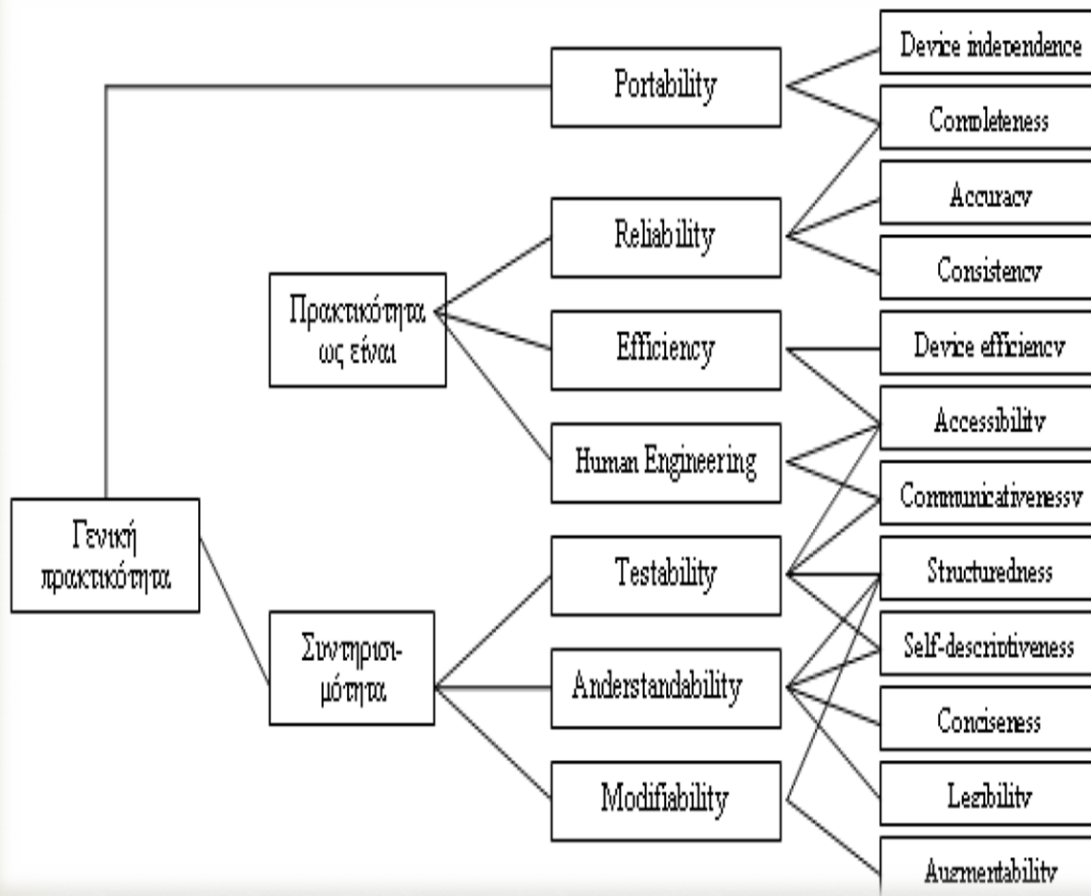
✧ Ο McCall προτείνει την τμηματοποίηση της ποιότητας σε παράγοντες ποιότητας. Επειδή τόσο η ίδια η ποιότητα, όσο και οι ποιοτικοί παράγοντες είναι εξαιρετικά αφηρημένες έννοιες, πρότεινε την τμηματοποίηση των παραγόντων σε κριτήρια (criteria) που βρίσκονται σε χαμηλότερο επίπεδο αφαίρεσης και τα οποία μπορούν να μετρηθούν άμεσα με μετρικές (metrics). Πρότεινε οι μετρήσεις για κάθε κριτήριο να προκύπτουν από απαντήσεις σε ερωτήσεις για το κριτήριο. Από τα ονόματα των τριών επιπέδων αφαίρεσης το μοντέλο αυτό ονομάστηκε FCM (Factors Criteria Metrics).

Μοντέλο FCM



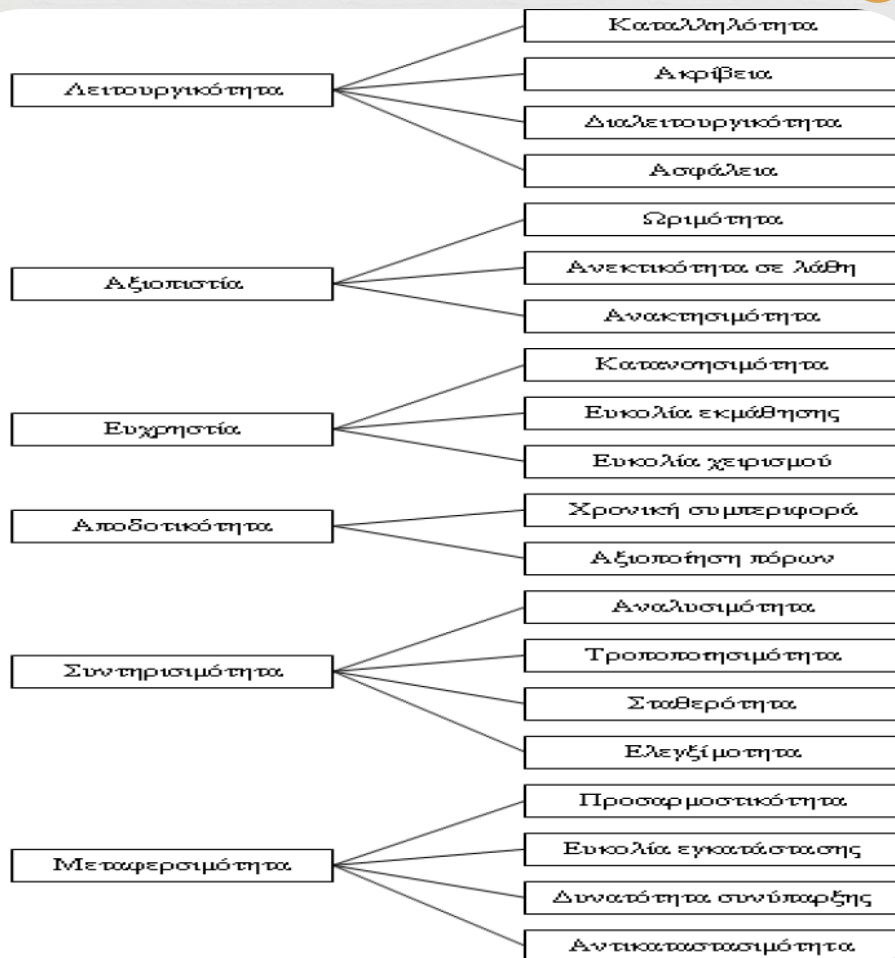
Το μοντέλο αυτό αποτέλεσε ένα από τα πιο ολοκληρωμένα μοντέλα της εποχής του κι έγινε βάση για το διεθνές πρότυπο ISO 9126. Παρά τα προβλήματα της υποκειμενικότητας των ερωτήσεων, της ύπαρξης περιορισμένης κλίμακας (το μοντέλο δεχόταν μόνο απαντήσεις «ΝΑΙ» και «ΟΧΙ») και της αδυναμίας συνδυασμού μετρικών, το μοντέλο αυτό γνώρισε ευρεία αποδοχή και ακόμη και σήμερα που το ISO 9126 κυριαρχεί, αρκετές επιχειρήσεις βασίζουν το σύστημα ποιότητάς τους στην τμηματοποίηση του FCM μοντέλου.

Μοντέλο Boehm



Το μοντέλο του Boehm έχει παρόμοια ιεραρχική δομή με το FCM μοντέλου, σύμφωνα με την οποία διασπά την ποιότητα του λογισμικού σε πρωταρχικές χρήσεις (primary uses) και αυτές σε ενδιάμεσες κατασκευές (intermediate constructs), ανάλογες με τα κριτήρια ποιότητας του μοντέλου FCM. Οι ενδιάμεσες κατασκευές με τη σειρά τους διασπώνται σε πρωτογενείς κατασκευές (primitive constructs) οι οποίες μετρώνται άμεσα με μετρικές (metrics).

Πρότυπο ISO 9126



Το πρότυπο ISO 9126 αποτελεί ένα μοντέλο ποιότητας λογισμικού που εξελίχθηκε σε διεθνές πρότυπο από το διεθνή οργανισμό τυποποίησης ISO. Ως μοντέλο ποιότητας λογισμικού διαφέρει από τα προγενέστερα μοντέλα τόσο στην ορολογία όσο και στη δομή καθώς είναι απόλυτα ιεραρχικό.

Μετρικές Ποιότητας Λογισμικού



Μετρήσεις και Μετρικές

Στην τεχνολογία λογισμικού, οι όροι μέτρηση (Measure) και μετρική (Metric) είναι αλληλένδετοι και άμεσα εξαρτώμενοι. Μπορούμε να ορίσουμε ως Μέτρηση την αντιστοίχιση ενός μεγέθους με μια τιμή και την μετρική ως τη ποσοτική μέτρηση του βαθμού, στον οποίο ένα σύστημα ή τμήμα αυτού έχει ένα χαρακτηριστικό. Βέβαια οι ορισμοί είναι γενικοί αλλά στην τεχνολογία λογισμικού και ποιότητα λογισμικού ορίζεται πιο εμπειριστατωμένα, όπως θα δούμε.

Μετρική



- ☞ Σύμφωνα με το Fenton, η μετρική ορίζεται ως μία εμπειρική και αντικειμενική αντιστοίχιση ενός αριθμού ή συμβόλου σε μία οντότητα με σκοπό να χαρακτηρίσει ένα συγκεκριμένο χαρακτηριστικό της.
- ☞ Σύμφωνα με το Edward Berard η μετρική ορίζεται ως μία μονάδα μέτρησης και αναφέρει ότι ο όρος χρησιμοποιείται για να δηλώσει ένα σύνολο μετρήσεων που πραγματοποιούνται πάνω σε συγκεκριμένο αντικείμενο ή διαδικασία.
- ☞ Τέλος, ο Tom De Marco (1982) είχε δηλώσει πολύ εύστοχα ότι “You cannot control what you cannot measure”.

Συμπερασματικά, μπορούμε να πούμε ότι μετρική είναι μία ‘μεθοδολογία’ μέτρησης που αντιστοιχεί μία τιμή σε κάποια προϋπάρχουσα ιδιότητα του αντικειμένου.

Οι μετρικές χωρίζονται σε :



- A. Μετρικές Διαδικασίας (Άνθρωποι, Τεχνολογία, Προϊόν)
- B. Μετρικές Έργου (Π.χ. εκτίμηση διάρκειας και κόστους του έργου, COCOMO)
- C. Μετρικές Προϊόντος (Pressman, 1997)
 - 1. **Εσωτερικές:** αριθμός γραμμών (LOC), χρόνος εκτέλεσης, λάθη του κώδικα
 - 2. **Εξωτερικές** – λειτουργικότητα (functionality), ποιότητα (quality), πολυπλοκότητα (complexity), αποτελεσματικότητα (efficiency), αξιοπιστία (reliability), συντηρησιμότητα (maintainability).

Εσωτερικές Μετρικές



- ∞ Διαδομένες Εσωτερικές Μετρικές
- I. Μετρικές της επιστήμης λογισμικού
- II. Μετρική κυκλωματικής πολυπλοκότητας
- III. Μετρική πολυπλοκότητας δομών δεδομένων

Εξωτερικές Μετρικές



☞ Εξυπηρετούν με άμεσο τρόπο τη δυνατότητα μέτρησης της ικανοποίησης των απαιτήσεων του πελάτη, η οποία αποτελεί βασικό στοιχείο κάθε ορισμού της ποιότητας.

Τα αρνητικά στοιχεία των εξωτερικών μετρικών είναι:

1. Βασίζονται σε υποκειμενικά δεδομένα και επομένως τα αποτελέσματά τους είναι πιθανόν να υπόκεινται σε αμφισβητήσεις
2. Η αδυναμία αυτοματοποίησης της διαδικασίας των εξωτερικών μετρήσεων.

Στόχοι των Μετρικών

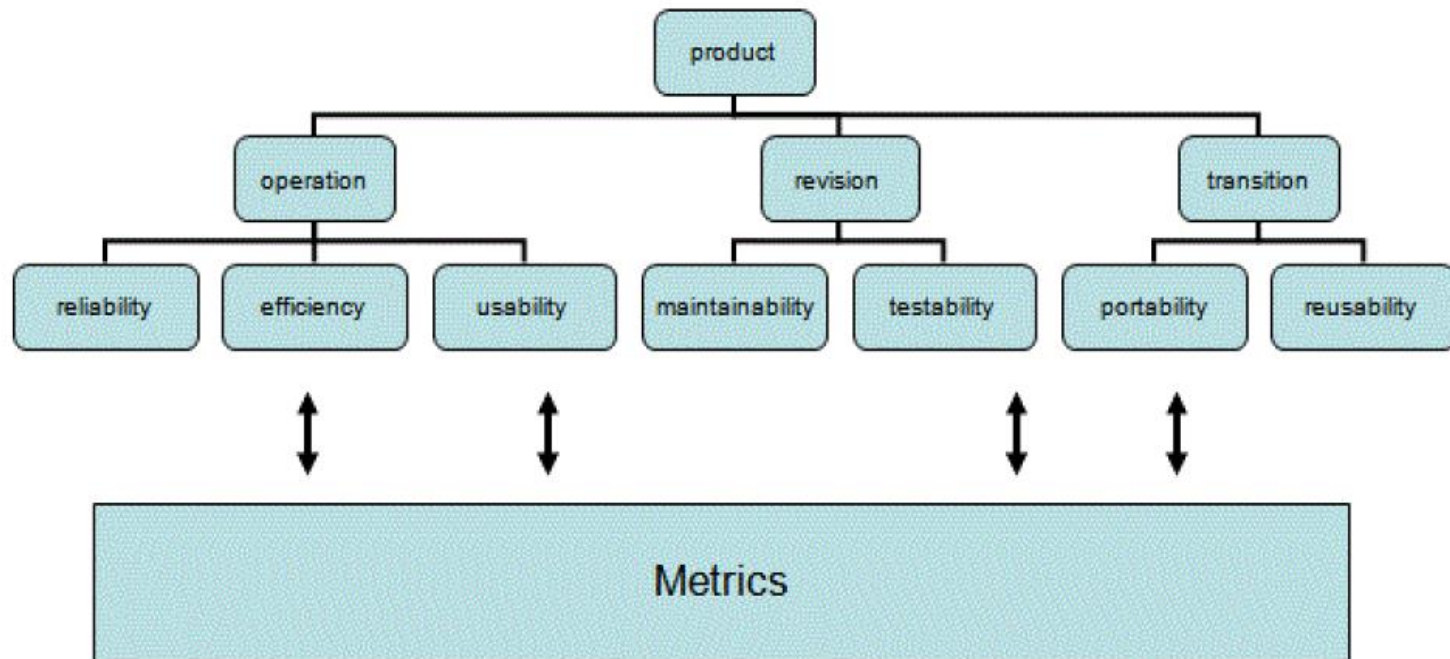


Ο κυριότερος στόχος των μετρικών που εφαρμόζονται και διεξάγονται σε ένα λογισμικό είναι η διασφάλιση της ποιότητας.

☞ Παράλληλα με τις μετρικές που χρησιμοποιούνται, η διασφάλιση ποιότητας διακρίνεται σε:

1. Διασφάλιση ποιότητας προϊόντος (Product Quality Assurance): στόχος είναι η ποιότητα του προϊόντος και κατ' επέκταση η ικανοποίηση του χρήστη.
2. Διασφάλιση ποιότητας έργου (Project Quality Assurance): που στοχεύει
 - α) στην αποτίμηση της κατάστασης του έργου,
 - β) την επίλυση προβλημάτων πριν αυτά γίνουν 'κρίσιμα' και επηρεάσουν την πορεία του έργου
 - γ) τη ρύθμιση και το διαχωρισμό των εργασιών
 - δ) την εκτίμηση της ικανότητας των ομάδων να παράγουν 'ποιοτικό' λογισμικό.
3. Διασφάλιση ποιότητας διαδικασιών (Process Quality Assurance): που βοηθά την εταιρεία να έχει γνώση για την αποτελεσματικότητα των διαδικασιών.

Μοντέλο Ποιότητας Μετρικών



Στόχοι των Μετρικών



Με βάση τους στόχους για τους οποίους έχουν αναπτυχθεί οι μετρικές μπορούν να διαχωριστούν στις ακόλουθες κατηγορίες:

- **Λειτουργικές μετρικές (Functional Metrics)** που αποτελούνται από μετρικές καταλληλότητας (Suitability), ακρίβειας (Accuracy), διαλειτουργικότητας (interoperability), ελαστικότητας (Compliance) και ασφάλειας (Security) του λογισμικού.
- **Μετρικές αξιοπιστίας (Reliability Metrics)** που αποτελούνται από μετρικές ωριμότητας (Maturity), ανοχής λαθών (Fault Tolerance), ικανότητας ανάκτησης (Recoverability).
- **Μετρικές χρησιμότητας (Usability metrics)** που αποτελούνται από μετρικές “κατανοησιμότητας” (Understandability), ικανότητας μάθησης (Learn Ability) και λειτουργικότητας (Operability) του λογισμικού.
- **Μετρικές αποδοτικότητας (Efficiency Metrics)** που αποτελούνται από μετρικές συμπεριφοράς του συστήματος στο χρόνο (System Behavior over time) και χρήσης των πόρων (Usage of Resources).
- **Μετρικές συντηρησιμότητας (Maintainability Metrics)** που αποτελούνται από μετρικές ικανότητας ανάλυσης (Analysis Ability), τροποποιησιμότητας (Changeability), σταθερότητας (Stability) και ικανότητας ελέγχου (Testability).
- **Μετρικές μεταφερσιμότητας (Portability Metrics)** που αποτελούνται από μετρικές προσαρμοστικότητας (Adaptability), ικανότητας εγκατάστασης (Install Ability), συμμόρφωσης (Conformance) και ικανότητας αντικατάστασης (Replace Ability).

Μετρικές Λογισμικού



Οι μετρικές λογισμικού εφαρμόζονται γενικά στα λογισμικά παραγωγής, αρκετές από αυτές βρίσκουν εφαρμογή και σε αντικειμενοστραφή λογισμικό, το οποίο είναι αποτέλεσμα μεταγλώττιση κώδικα κάποιας αντικειμενοστραφής γλώσσας (Java, JavaScript, PHP, C++, Python). Κατατάσσονται στις εσωτερικές μετρικές του τελικού προϊόντος.

Σύμφωνα με τον Roger Pressman, οι μετρικές λογισμικού χωρίζονται σε κατηγορίες, ανάλογα με τη φάση ανάπτυξης του λογισμικού που εφαρμόζονται:

- Μετρικές για το μοντέλο της ανάλυσης απαιτήσεων
- Μετρικές για το μοντέλο του σχεδιασμού
- Μετρικές για τον πηγαίο κώδικα
- Μετρικές για τον έλεγχο του λογισμικού
- Μετρικές για τη συντήρηση

Μετρικές Αντικειμενοστραφούς Λογισμικού



Ονομαστικά οι πιο γνωστές μετρικές προϊόντος που εφαρμόζονται στον τελικό κώδικα του λογισμικού.

- Average Module Length
- Chen Metric
- Decision Count
- Executable Statements
- Extent Of Reuse
- Function Count
- Hausen Metric On Modularity
- Live Variables
- Lines Of Code
- McCabe' s Essential Complexity Measure
- Reach Ability
- Tree Impurity

Μετρικές Αντικειμενοστραφούς Λογισμικού

Αρχές αντικειμενοστραφούς σχεδίασης

Στον αντικειμενοστραφή προγραμματισμό τον κύριο ρόλο έχουν οι κλάσεις και τα αντικείμενα. Κάθε αντικείμενο έχει μια συμπεριφορά, η οποία καθορίζεται από τις μεθόδους (συναρτήσεις) που αυτό διαθέτει. Μεταξύ των αντικειμένων και των κλάσεων δεν υπάρχουν απλώς σχέσεις καλούντος-καλούμενου, αλλά πλήθος άλλων σχέσεων, όπως ιεραρχίας, συνδέσμου κλπ.

Οι βασικές αρχές του Αντικειμενοστραφούς Προγραμματισμού είναι:

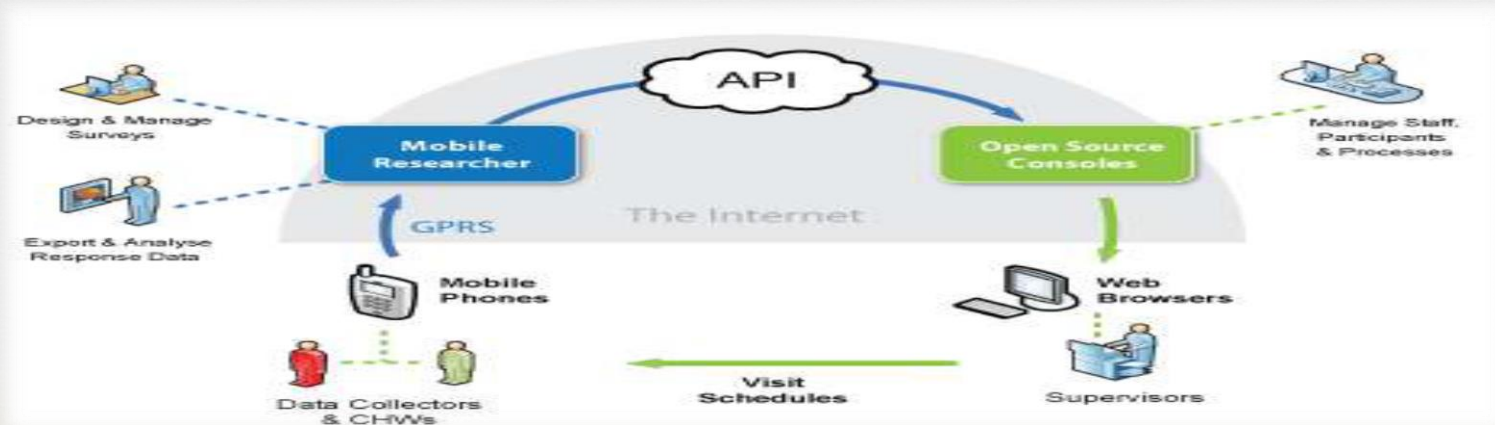
- ✓ Η Αφαίρεση (Abstraction)
- ✓ Η Ενθυλάκωση (Encapsulation) – ή Απόκρυψη Δεδομένων (Data Hiding)
- ✓ Η Κληρονομικότητα (Inheritance)
- ✓ Ο Πολυμορφισμός (Polymorphism)

Διεπαφή Προγραμματισμού Εφαρμογών (Application Program Interfaces - APIs)



Ορισμός

- ✧ Η διεπαφή των προγραμματιστικών διαδικασιών που ένα λειτουργικό σύστημα, βιβλιοθήκη ή εφαρμογή παρέχει προκειμένου να επιτρέψει να γίνονται προς αυτό αιτήσεις από άλλα προγράμματα ή / και ανταλλαγή δεδομένων. Το API επιτρέπει στους προγραμματιστές να έχουν πρόσβαση στις λειτουργικότητες ενός προγράμματος μέσα από καλά καθορισμένες δομές δεδομένων.



Λειτουργίες των APIs

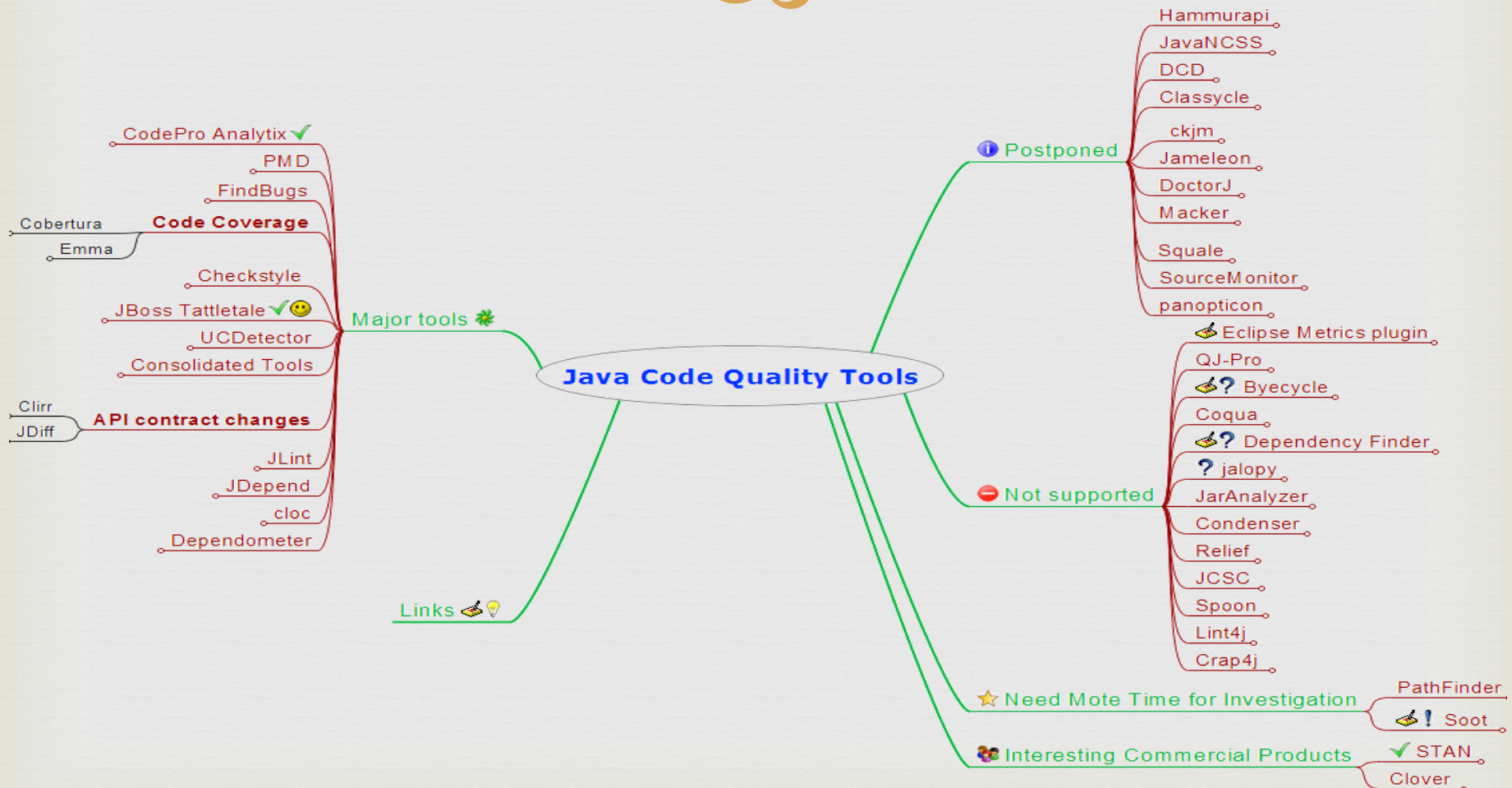


Ένα API είναι μία λογισμικό-προς-λογισμικό διεπαφή και όχι μία διεπαφή χρήστη. Με τα API, οι εφαρμογές μιλούν μεταξύ τους χωρίς τη γνώση ή την παρέμβαση κάποιου χρήστη.

Ένα API μπορεί να είναι:

- Language - Dependent
- Language - Independent

Εργαλεία Μετρικών



Εργαλεία Μετρικών



<u>Σημαντικότερα εργαλεία</u>	<u>Δευτερεύοντα Εργαλεία</u>
Codepro Analytix	Eclipse Metrics plugin
PMD	QJ-Pro
Findbugs	Condenser
Checkstyle	JCSC
JBoss Tattletale	Spoon
QALab	JavaNCSS

Εμπειρικές μελέτες



Στα πλαίσια της πτυχιακής εργασίας πραγματοποιήθηκε εμπειρική μελέτη ανασκόπησης, όπου κατεγράφησαν και αναλύθηκαν 30 έρευνες, χρονολογούμενες από το 2010 έως το 2015, ενώ στο ερευνητικό μέρος οι 17 εξ αυτών αναλύθηκαν σχολαστικά και παρουσιάστηκαν στο έκτο κεφάλαιο.

Ακτινογραφώντας και αναλύοντας τις υπό εξέταση έρευνες, εξήχθησαν και τα παρακάτω συμπεράσματα:

☞ Η βιβλιοθήκες λογισμικού αποτελούν πλέον είναι αναπόσπαστο τμήμα του δομημένου προγραμματισμού και αναπτύχθηκαν παράλληλα με αυτόν καθώς περιέχουν υποβοηθητικό κώδικα και δεδομένα, παρέχοντας, με αυτόν τον τρόπο, υπηρεσίες σε προγράμματα.

Εμπειρικές μελέτες



☞ Η λειτουργία τους είναι ιδιαίτερα σημαντική στην λειτουργία και χρησιμότητα των εφαρμογών και για τον λόγο αυτόν η χρήση των μετρικών στα APIs είναι κομβικής σημασίας.

☞ Μείζον θέμα στις μετρικές είναι η εμπειρία, καθώς όσο μεγαλύτερη εμπειρία έχουν οι μετρικές, τόσο καλύτερα αποτελέσματα εξάγονται.

☞ Οι μετρικές των βιβλιοθηκών αντικειμενοστρεφούς προγραμματισμού, κερδίζουν ολοένα και περισσότερο «έδαφος» και αυτό έχει ως συνέπεια να υλοποιούνται νέες μετρικές και να αναβαθμίζονται οι υπάρχουσες. Μελλοντική προοπτική της εργασίας είναι αφενός, η διερεύνηση και ανάλυση περισσότερων μετρικών Α/Σ λογισμικού και αφετέρου η εμπειρική μελέτη ανασκόπησης από το 2015 και μετά.

Ευχαριστώ για το χρόνο

σας

Ω

Ρούμπος Γιώργος