ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Μεθοδολογία έγκαιρης εκτίμησης της γνώμης
των χρηστών για την ποιότητα λογισμικού

Δημήτριος Π. Σταυρινούδης

Επιβλέπων Καθηγητής: Δ. Χριστοδουλάκης

ΠΑΤΡΑ, 2005

UNIVERSITY OF PATRAS
SCHOOL OF ENGINEERING
COMPUTER ENGINEERING AND INFORMATICS
DEPARTMENT

PhD THESIS

A methodology for early estimation of users'
opinions of software quality

Dimitrios P. Stavrinoudis

PATRAS, 2005

Περιεχόμενα

Περιεχόμενα	7
Περίληψη	11
Abstract	13
Ευχαριστίες	15
1. Εισαγωγή.....	17
1.1 Ορισμός της ποιότητας και Ιστορική Αναδρομή.....	17
1.2 Ποιότητα και Μετρήσεις	18
1.3 Κρίση του Λογισμικού	19
1.4 Μοντέλα Ποιότητας.....	21
1.4.1 Μοντέλο του McCall	21
1.4.2 Το πρότυπο ISO9126	22
1.5 Η Έννοια της Μετρικής.....	23
1.6 Στοιχεία ενός Προγράμματος Εξασφάλισης Ποιότητας.....	25
1.7 Ο Ρόλος του Πελάτη-Χρήστη	26
1.7.1 Διαφορετικές όψεις για την ποιότητα	27
1.7.2 Αντικείμενο της διδακτορικής διατριβής	28
1.8 Δομή της διδακτορικής διατριβής	29
2. Χρήση Εσωτερικών Μετρικών Λογισμικού	31

2.1	Πλεονεκτήματα και Μειονεκτήματα	31
2.2	Διαδεδομένες Μετρικές.....	32
2.2.1	Μετρικές της επιστήμης λογισμικού.....	32
2.2.2	Μετρική κυκλωματικής πολυπλοκότητας.....	33
2.2.3	Μετρική πολυπλοκότητας δομών δεδομένων.....	33
2.2.4	Άλλες γνωστές μετρικές.....	34
2.3	Εργαλεία Εσωτερικών Μετρήσεων	34
2.3.1	Το εργαλείο ATHENA	35
2.4	Μεταμετρικές.....	37
2.5	Έρευνα Αντικειμενοστραφών Μετρικών Λογισμικού	39
2.5.1	Παρουσίαση Μετρικών.....	39
2.5.2	Παρουσίαση Μεταμετρικών και Αποτελεσμάτων.....	43
3.	Εξωτερικές Μετρικές Λογισμικού.....	47
3.1	Πλεονεκτήματα και Μειονεκτήματα	47
3.2	Ποιος είναι ο Πελάτης.....	48
3.3	Τεχνικές Μέτρησης	50
3.3.1	Αναλυτικές Τεχνικές.....	50
3.3.2	Πειραματικές Τεχνικές.....	51
3.3.3	Διερευνητικές Τεχνικές.....	51
3.3.4	Τα στάδια μίας έρευνας με ερωτηματολόγιο	52
3.3.5	Πλεονεκτήματα και μειονεκτήματα.....	53
3.3.6	Τύποι και βάρη ερωτήσεων	55
3.4	Μέθοδος Ανάλυσης Αποτελεσμάτων	59
3.4.1	Μέση γνώμη.....	59
3.4.2	Γνώμη με βαρύτητα προσόντων	59
3.4.3	Γνώμη με βαρύτητα προσόντων και ερωτήσεις ασφαλείας.....	60
3.4.4	Γνώμη με βαρύτητα προσόντων και διπλές ερωτήσεις ασφαλείας.....	60
3.5	Το Εργαλείο CESM.....	61
3.5.1	Η αρχιτεκτονική του εργαλείου CESM	62
3.5.2	Σχεδιαστής Ερωτηματολογίων.....	63
3.5.3	Ιστοσελίδα ερωτηματολογίου	64
3.5.4	Στατιστικός Αναλυτής Ερωτηματολογίων.....	65
4.	Συσχέτιση Εσωτερικών και Εξωτερικών Μετρικών Λογισμικού....	67
4.1	Συσχέτιση εσωτερικών και εξωτερικών μετρήσεων	67
4.2	Περιπτωσιολογική μελέτη	69
4.2.1	Εσωτερικές μετρήσεις.....	70
4.2.2	Εξωτερικές μετρήσεις	71
4.2.3	Αποτελέσματα και συσχετίσεις.....	72
4.2.4	Συμπεράσματα	76
5.	Συσχέτιση Εσωτερικών Μετρικών και Συντηρησιμότητας.....	79
5.1	Η έννοια της συντηρησιμότητας	79
5.1.1	Οι τύποι της συντήρησης λογισμικού	80

5.1.2	Απόψεις χειρισμού της διαδικασίας συντήρησης	80
5.1.3	Προσεγγίσεις μέτρησης της συντηρησιμότητας	82
5.2	Χρήση μετρικών λογισμικού για τη βελτίωση της συντηρησιμότητας	82
5.3	Περιπτώσιολογικές μελέτες	85
5.3.1	Πρώτη περιπτώσιολογική μελέτη	85
5.3.2	Δεύτερη περιπτώσιολογική μελέτη	87
5.3.3	Συμπεράσματα	92
6	Μεταβολές στις Απόψεις για την Ποιότητα Λογισμικού	95
6.1	Η ανάγκη για μετρήσεις αξιολόγησης.....	95
6.2	Μετρήσεις με την πάροδο του χρόνου	97
6.2.1	Αποτελέσματα των μετρήσεων	97
6.2.2	Βελτίωση της διαδικασίας μετρήσεων	99
6.3	Η ανάγκη ενός μοντέλου.....	100
6.3.1	Προδιαγραφές του προτεινόμενου μοντέλου	101
6.3.2	Καθορισμός συνόλου από περιορισμούς	102
6.4	Έγκαιρη εκτίμηση της γνώμης των χρηστών	103
6.4.1	Σύστημα Σφαιρών του Grove	103
6.4.2	Σχεδίαση του μοντέλου.....	104
6.4.3	Μοντελοποίηση των διαφόρων κατηγοριών των χρηστών	105
6.4.4	Αναπαράσταση περιορισμών στο Σύστημα Σφαιρών	106
6.5	Συμπεράσματα	107
7	Χρήση Αρχών από τη θεωρία Αναθεώρησης Άποψης	109
7.1	Εισαγωγή στην AGM θεωρία Αναθεώρησης Άποψης	109
7.2	Αναθεωρήσεις της γνώμης των χρηστών για την ποιότητα λογισμικού	112
7.3	Σχέση της αναθεώρησης βαθμολογίας με την AGM αναθεώρηση άποψης.....	116
7.4	Μελέτη υπολογιστικής πολυπλοκότητας.....	117
8	Ανάλυση της προτεινόμενης μεθοδολογίας.....	121
8.1	Σύντομη περιγραφή της μεθοδολογίας	121
8.1.1	Σκοπός της μεθοδολογίας	121
8.1.2	Παρουσίαση των φάσεων της μεθοδολογίας	122
8.2	Ανάλυση των βημάτων και χρήση της μεθοδολογίας.....	125
8.2.1	Εσωτερικές μετρήσεις.....	126
8.2.2	Εξωτερικές μετρήσεις	127
8.2.3	Αναθεώρηση άποψης χρήστη	130
8.2.4	Συνεχής εξέλιξη	131
8.2.5	Πειραματική εφαρμογή.....	132
8.3	Συμπεράσματα και επίλογος.....	134
Βιβλιογραφία	137	

Περίληψη

Ανατρέχοντας στη βιβλιογραφία ο ερευνητής θα διαπιστώσει ότι κατά καιρούς έχουν δοθεί διάφοροι ορισμοί για την ποιότητα. Μάλιστα, εύκολα μπορεί να παρατηρήσει ότι οι δύο φράσεις-κλειδιά που πρέπει να χαρακτηρίζουν τον ορισμό της ποιότητας είναι α) τα ποιοτικά χαρακτηριστικά και β) οι απαιτήσεις του πελάτη-χρήστη. Η εξασφάλιση της ποιότητας επιπλέον ήταν ανέκαθεν συνυφασμένη με τις μετρήσεις. Ως μέτρηση ορίζεται η διαδικασία με την οποία αριθμοί ή σύμβολα αντιστοιχίζονται σε ιδιότητες οντοτήτων του πραγματικού κόσμου, έτσι ώστε να τις περιγράφουν σύμφωνα με καθορισμένους κανόνες. Ο DeMacro μάλιστα αναφέρει ότι είναι αδύνατο να ελέγξεις ό,τι δεν μπορείς να μετρήσεις. Πόσο μάλλον αδύνατον είναι να στοχεύεις να εξασφαλίζεις την ποιότητα για κάτι το οποίο δεν μπορείς να μετρήσεις.

Είναι σαφές λοιπόν ότι η εκπλήρωση των απαιτήσεων του πελάτη λαμβάνεται πάντα υπόψη στον ορισμό της ποιότητας. Όμως, παρόλο που νέες μεθοδολογίες και τεχνικές έχουν αναπτυχθεί και εφαρμοστεί κατά τη διάρκεια της διαδικασίας παραγωγής λογισμικού, αλλά και τη διαχείριση των έργων, οι τελικοί χρήστες εξακολουθούν συχνά να αντιμετωπίζουν προβλήματα με το επίπεδο της ποιότητας των προϊόντων λογισμικού που απορρέουν. Αυτό το αποτέλεσμα οφείλεται κυρίως στις διαφορές που υπάρχουν ανάμεσα στους χρήστες και τους παραγωγούς λογισμικού όσον αφορά την αντίληψή τους για την ποιότητα, αλλά και στη διαφορετική σπουδαιότητα και προτεραιότητα που δίνει ο καθένας από αυτούς στα διάφορα ποιοτικά χαρακτηριστικά. Η ποιότητα, όπως για παράδειγμα η εξωτερική ομορφιά, εξαρτάται από το «μάτι» του παρατηρητή και κρίνεται σύμφωνα με ένα σύνολο επιθυμητών χαρακτηριστικών τα οποία έχουν διαφορετική βαρύτητα από παρατηρητή σε παρατηρητή.

Ο πιο άμεσος τρόπος για να μετρηθεί η γνώμη των χρηστών για την ποιότητα ενός προϊόντος λογισμικού είναι να διεξαχθεί μία έρευνα (με ερωτηματολόγια ή συνεντεύξεις) κατά την οποία θα ρωτηθούν, αν όχι όλοι, σίγουρα ένας αξιόλογος αριθμός χρηστών (εξωτερική μέτρηση). Η έρευνα αυτή θα πρέπει να γίνει προφανώς μετά την τελική παράδοση του προϊόντος και κατόπιν ενός εύλογου χρόνου χρήσης του από όλους τους χρήστες που θα συμμετέχουν στην έρευνα. Αν και αυτός είναι ο εγκυρότερος τρόπος για να αποφανθούμε για την ποιότητα του λογισμικού όπως την αντιλαμβάνεται ο χρήστης, τα βασικά προβλήματα που προκύπτουν είναι πρώτον ότι

το κόστος μίας τέτοιας έρευνας μπορεί να γίνει πολύ μεγάλο, ειδικά σε περιπτώσεις που δεν είναι σωστά δομημένη, καθώς τότε θα απαιτούνται για αυτήν αρκετές ανθρωποώρες τόσο από πλευράς εταιρείας λογισμικού όσο και από πλευράς των ίδιων των χρηστών, και δεύτερον ότι η έρευνα αυτή ενδέχεται να διεξαχθεί πολύ αργά.

Το ερώτημα λοιπόν που προκύπτει είναι πώς μπορούμε να βρούμε κάποιες ενδείξεις για την ποιότητα του λογισμικού; Πώς μπορούμε δηλαδή να μετρήσουμε, να προβλέψουμε, να εκτιμήσουμε τη γνώμη του πελάτη πριν το τελικό στάδιο του κύκλου ζωής λογισμικού; Είναι εύκολα αντιληπτό ότι αν μία εταιρεία παραγωγής λογισμικού μπορεί εγκαίρως να παρατηρήσει και να κατανοήσει τις ελλείψεις και τα ελαττώματα που πιθανώς επιτελέστηκαν κατά τη διαδικασία παραγωγής, τότε μπορεί με διορθωτικές ενέργειες να βελτιώσει την ποιότητα των προϊόντων της. Κάτι τέτοιο θα αύξανε το κέρδος της, αφού οι παραπάνω διορθώσεις θα μπορούσαν να διεξαχθούν πριν από τη φάση της συντήρησης, μειώνοντας κατά πολύ το κόστος αλλαγής.

Στην παρούσα διδακτορική διατριβή προτείνονται διάφορες μέθοδοι και τεχνικές που συνεισφέρουν στην έγκαιρη εκτίμηση της γνώμης των χρηστών για την ποιότητα λογισμικού. Τέτοιες μέθοδοι είναι α) η επιλογή και χρήση μετρικών λογισμικού και η ανάλυση των αποτελεσμάτων τους, β) η δόμηση και η ανάλυση ερωτηματολογίων για τη μέτρηση της γνώμης των χρηστών για την ποιότητα λογισμικού, γ) η συσχέτιση εσωτερικών μετρικών λογισμικού με τα εξωτερικά ποιοτικά χαρακτηριστικά ενός προγράμματος λογισμικού, δ) η μεταβολή της γνώμης του χρήστη για την ποιότητα ενός προγράμματος λογισμικού με την πάροδο του χρόνου σε σχέση με το επίπεδο εμπειρίας του χρήστη και ε) η χρήση και η προσαρμογή κανόνων και μοντέλων από τη θεωρία Αναθεώρησης Άποψης. Όλες οι παραπάνω μέθοδοι συνδυάζονται ώστε να συνεισφέρουν στην προτεινόμενη μεθοδολογία της διδακτορικής διατριβής.

Οι βασικοί στόχοι της μεθοδολογίας συνοψίζονται στα παρακάτω: α) έγκαιρη εκτίμηση της ποιότητας λογισμικού, ώστε να διορθωθούν νωρίς τα πιθανά λάθη και να γίνουν βελτιώσεις σε πρώιμες φάσεις του κύκλου ζωής λογισμικού, β) ενεργή συμμετοχή του τελικού χρήστη στη διαδικασία παραγωγής λογισμικού, γ) μείωση της προσπάθειας που θα απαιτηθεί στη φάση της συντήρησης, δ) μείωση του κόστους για την εύρεση της γνώμης του χρήστη για την ποιότητα ενός προϊόντος λογισμικού, ε) καθορισμός μετρήσιμων στόχων κατά την παραγωγή του λογισμικού και παρακολούθηση των βελτιώσεων από έκδοση σε έκδοσή του και στ) δυνατότητα εστίασης σε συγκεκριμένα χαρακτηριστικά της ποιότητας και καθορισμός των απαιτούμενων ενεργειών.

Η μεθοδολογία που παρουσιάζεται στη διατριβή μπορεί να συμπεριληφθεί και να εφαρμοστεί στο πρόγραμμα ποιότητας μίας εταιρείας παραγωγής λογισμικού. Εναλλακτικά, μπορεί να γίνει εστίαση σε συγκεκριμένα βήματα που ακολουθεί και να επιλεγούν οι αντίστοιχες από τις προτεινόμενες μεθόδους που την αποτελούν. Τέλος, παρόλο που για την αντιμετώπιση του προβλήματος της έγκαιρης εκτίμησης της γνώμης των χρηστών για την ποιότητα λογισμικού η παρούσα μεθοδολογία μπορεί να θεωρηθεί πλήρης και να εφαρμοστεί αυτούσια σε ένα πρόγραμμα ποιότητας, παρέχει επίσης τη δυνατότητα εμπλουτισμού της και με άλλες ερευνητικές μεθόδους.

Abstract**Abstract**

In this dissertation, a number of methods and techniques that contribute to the early estimation of users' opinions of software quality are proposed. These are a) the selection and use of software metrics and the analysis of their results, b) the formation and analysis of questionnaires measuring users' opinions of software quality, c) the correlation between internal software metrics and the external quality characteristics of a software program, d) the differentiation of a user's opinion of the quality of a software program over time in relation to the experience level of the user and e) the use and adaptation of rules and models of Belief Revision theory. The combination of these methods results in the proposed methodology of this dissertation.

Ευχαριστίες

Καταρχάς θα ήθελα να ευχαριστήσω τον κ. Δημήτρη Χριστοδουλάκη, ο οποίος ήταν και ο επιβλέπων καθηγητής της διδακτορικής μου διατριβής. Δίπλα του μου δόθηκε η δυνατότητα να ξεκινήσω τα πρώτα μου βήματα ως Ερευνητής και Μηχανικός Η/Υ και Πληροφορικής στο Εργαστήριο Βάσεων Δεδομένων του Τμήματος Μηχανικών Η/Υ και Πληροφορικής και στην Ερευνητική Μονάδα ΙΙ του Ερευνητικού Ακαδημαϊκού Ινστιτούτου Τεχνολογίας Υπολογιστών.

Θα ήθελα επίσης να ευχαριστήσω ειλικρινά το Μιχάλη Ξένο, μέλος της εξεταστικής επιτροπής, για την αμέριστη βοήθεια, συμπαράσταση και άψογη συνεργασία που μου παρείχε καθόλη τη διάρκεια της διατριβής. Υπήρξε για μένα όχι μόνο σύμβουλος, καθοδηγητής και πραγματικός συνεργάτης σε όλη την πορεία των μεταπτυχιακών μου σπουδών, αλλά και ουσιαστικός μέντορας τόσο στον χώρο της ποιότητας λογισμικού όσο και γενικότερα στην ερευνητική μου πορεία. Η αρωγή του ήταν καταλυτική στη διαμόρφωση της σημερινής ερευνητικής μου προσωπικότητας.

Παράλληλα, ευχαριστώ όλα τα υπόλοιπα μέλη ΔΕΠ της επταμελούς εξεταστικής επιτροπής, τους κκ. Γιάννη Γαροφαλάκη, Παύλο Σπυράκη (μέλη τριμελούς συμβουλευτικής επιτροπής), Θανάση Τσακαλίδη, Νίκο Αβούρη και Παναγιώτη Πιντέλα που με τις υποδείξεις τους βοήθησαν στην αρτιότερη παρουσίαση της διατριβής.

Αν έπρεπε να προσδιορίσω ένα σημείο σταθμό για την πρόοδο της διδακτορικής μου διατριβής, θα ήταν η γνωριμία μου και στη συνέχεια η συνεργασία με τον καθηγητή Παύλο Πέππα που με τη βοήθειά του μπορέσαμε να ενσωματώσουμε αρχές από τη θεωρία Αναθεώρησης Άποψης στην έρευνά μας στην Ποιότητα Λογισμικού.

Θα ήταν παράλειψη να μην ευχαριστήσω την Κωνσταντινιά Ζηκούλη και την Αντωνία Στεφανή για τη συνεργασία μας σε δημοσιεύσεις, οι οποίες συνέβαλαν στην τεκμηρίωση της διδακτορικής διατριβής. Επίσης, ευχαριστώ ιδιαίτερα τους συναδέλφους Μανώλη Τζαγκαράκη και Κώστα Παπουτσή για την ανταλλαγή απόψεων και προβληματισμών σε όλη την πορεία των μεταπτυχιακών σπουδών μου. Θεωρώ ότι βοηθήσαμε ουσιαστικά και ψυχολογικά ο ένας τον άλλο για την ολοκλήρωση των διδακτορικών διατριβών μας.

Σημαντική ήταν η συμβολή του καλού μου φίλου Παναγιώτη Φαρμάκη, που με βοήθησε όχι μόνο σε θέματα έκφρασης και διατύπωσης στις δημοσιεύσεις μου, αλλά

γενικότερα στο να αντιλαμβάνομαι την πολύπλευρη διάσταση των πραγμάτων, τόσο σε ερευνητικά όσο και σε καθημερινά θέματα.

Οφείλω να ευχαριστήσω και τους συναδέλφους και φίλους του Εργαστηρίου Βάσεων Δεδομένων και της Ερευνητικής Μονάδας II του Ε.Α.Ι.Τ.Υ. που συνεργαστήκαμε τα χρόνια των μεταπτυχιακών σπουδών μου και ιδιαίτερα τους Μιχάλη Βαΐτη, Γιώργο Ορφανό, Δημήτρη Αβραμίδα, Μαρία Κυριακοπούλου, Νίκο Καρούσο, Δέσποινα Τζήμου, Κατερίνα Μπαρμπετάκη, Γιώργο Γκότση, Δημήτρη Χάνο, Χαρά Διαμαντοπούλου, Σπύρο Φατούρο και Χρήστο Χαιρετάκη. Παράλληλα, ευχαριστώ τους συνεργάτες μου στο Εργαστήριο Εκπαιδευτικού Υλικού και Εκπαιδευτικής Μεθοδολογίας του Ελληνικού Ανοικτού Πανεπιστημίου, στο χώρο του οποίου ολοκλήρωσα τα τελευταία βήματα της διατριβής μου.

Τέλος, θα πρέπει προφανώς να ευχαριστήσω την οικογένειά μου, που μου συμπαραστάθηκε σε όλη τη διάρκεια των σπουδών μου και μου παρείχε ψυχολογική υποστήριξη στις δυσκολίες που είχα να αντιμετωπίσω.

Κεφάλαιο**1**

Εισαγωγή

1.1 Ορισμός της ποιότητας και Ιστορική Αναδρομή

Ανατρέχοντας στη βιβλιογραφία ο ερευνητής θα διαπιστώσει ότι κατά καιρούς έχουν δοθεί διάφοροι ορισμοί για την ποιότητα. Η έννοια της ποιότητας ξεκινάει με τον Αριστοτέλη, που πρώτος διακρίνει το «ποιόν» και τα χαρακτηριστικά του, συνεχίζεται από τους σχολαστικούς και, αργότερα, από τον Μπέρκλυ, το Λάιμπνιτς, το Χέγκελ κ.ά. [ΕΓΚ-ΠΑΠ]. Σύμφωνα με το πρότυπο A3 του American Society for Quality Control, η ποιότητα ορίζεται ως «η συλλογή των χαρακτηριστικών και των ιδιοτήτων του προϊόντος που σχετίζονται με τη δυνατότητά του να εκπληρώνει τις ζητούμενες ανάγκες των πελατών» [AME78]. Άλλοι διαδεδομένοι ορισμοί που έχουν δοθεί για την ποιότητα είναι: «η συμμόρφωση με τις απαιτήσεις των χρηστών» [CRO79], «η καταλληλότητα για χρήση» [JUR80], «η συλλογή των χαρακτηριστικών σχεδιασμού, κατασκευής και συντήρησης, μέσω των οποίων το προϊόν κατά τη χρήση του θα εκπληρώσει τις προσδοκίες των πελατών» [FEI83], «η συμμόρφωση με τυποποιημένες προδιαγραφές που περιγράφουν τα βασικά χαρακτηριστικά του προϊόντος και έχουν βασιστεί στις ανάγκες και τις προσδοκίες των πελατών» [AUB85]. Από τα παραπάνω, εύκολα μπορεί να παρατηρήσει κάποιος ότι οι δύο φράσεις-κλειδιά που πρέπει να χαρακτηρίζουν τον ορισμό της ποιότητας είναι α) *τα ποιοτικά χαρακτηριστικά* και β) *οι απαιτήσεις του πελάτη-χρήστη*.

Σε όλες τις οργανωμένες κοινωνίες κάθε εποχής, από την αρχαιότητα μέχρι σήμερα, όταν υπάρχει μία υψηλή πολιτισμική στάθμη, αυτή συνοδεύεται πάντα από μία αναπτυγμένη τεχνολογία. Και, το πιο σημαντικό, η τελευταία λειτουργεί στη βάση ενός μηχανισμού που διασφαλίζει τον έλεγχο της ποιότητας και την προστασία του καταναλωτή. Χαρακτηριστικά παραδείγματα μπορεί κανείς να συναντήσει στην αρχαία Βαβυλώνα, όπου ανάμεσα στους νόμους του Hammurabi υπάρχει και ο αρχαιότερος κανονισμός σχετικά με την οικοδομή. Σύμφωνα με αυτόν, αν ένας οικοδόμος κτίσει μία κατοικία για κάποιον, αλλά δεν πραγματοποιήσει την εργασία του σύμφωνα με τους ισχύοντες κανονισμούς και πρότυπα σε τρόπο που ένας τοίχος παρουσιάσει κάποια κλίση, τότε ο οικοδόμος αυτός οφείλει να τον ενισχύσει με δικά του έξοδα [BAP96]. Στην αρχαία Αίγυπτο υπήρχε τυποποίηση των μετρήσεων βασισμένη στο μήκος του βραχίονα του εκάστοτε Φαραώ και με βάση αυτό το μήκος κατασκευαζόταν το «μοναδιαίο ραβδί». Κάθε κατασκευαστής έπρεπε να έχει ένα

αντίστοιχο ραβδί που όφειλε να το συγκρίνει με το μοναδιαίο κάθε γεμάτο φεγγάρι, ενώ η ποινή που του επιβαλλόταν σε περίπτωση διαφοράς ήταν ο θάνατος [XEN96a].

Στην κλασική Ελλάδα υπήρχε ένας μηχανισμός ελέγχου ποιότητας, τυποποίησης και πιστοποίησης των παραγόμενων και προσφερόμενων προϊόντων στον τόπο παραγωγής, αλλά και στην αγορά. Οι αρχαίοι Έλληνες εφάρμοζαν πρότυπα με πολύ αυστηρές προδιαγραφές που κάλυπταν όλο το φάσμα των τότε παραγόμενων προϊόντων, όπως τα μέταλλα και τα κράματά τους, τα γεωργικά προϊόντα, τα τρόφιμα και τα ποτά, αλλά και οι ίδιες οι κατασκευές, χαρακτηριστικό παράδειγμα των οποίων αποτελεί ο Παρθενώνας. Στην Αθηναϊκή Πολιτεία ορίζονταν με κλήρο δέκα μετρονόμοι που ευθύνονταν για όλα τα μέτρα και τα σταθμά της αγοράς και όφειλαν να μεριμνούν ώστε οι πωλητές να τα χρησιμοποιούν σωστά. Τέλος, η ποιότητα έπαιζε σημαντικό ρόλο και στην αρχαία Ρώμη. Αυτό μπορεί να διαπιστωθεί παρατηρώντας ότι αρκετές ρωμαϊκές γέφυρες όχι μόνο διατηρούνται, αλλά και χρησιμοποιούνται ακόμα και σήμερα.

1.2 Ποιότητα και Μετρήσεις

Η εξασφάλιση της ποιότητας λοιπόν ήταν ανέκαθεν συνυφασμένη με τις μετρήσεις. Και ως μέτρηση ορίζεται η διαδικασία με την οποία αριθμοί ή σύμβολα αντιστοιχίζονται σε ιδιότητες οντοτήτων του πραγματικού κόσμου, έτσι ώστε να τις περιγράφουν σύμφωνα με καθορισμένους κανόνες [FEN92]. Ο DeMacro μάλιστα αναφέρει ότι είναι αδύνατο να ελέγξεις ό,τι δεν μπορείς να μετρήσεις [DMA82]. Πόσο μάλλον αδύνατον είναι να στοχεύεις να εξασφαλίζεις την ποιότητα για κάτι το οποίο δεν μπορείς να μετρήσεις. Επιπλέον, δεν μπορούμε να συγκρίνουμε δύο αντικείμενα, παρά μόνο στα κοινά τους χαρακτηριστικά και η σύγκριση γίνεται βάσει του αριθμού (ή του συμβόλου) που δόθηκε κατά τη μέτρηση. Οι μετρήσεις χωρίζονται σε άμεσες και έμμεσες: Άμεση μέτρηση ενός χαρακτηριστικού είναι η μέτρηση που δεν βασίζεται σε μέτρηση κάποιου άλλου χαρακτηριστικού. Αντιθέτως, έμμεση μέτρηση ενός χαρακτηριστικού είναι η μέτρηση που περιλαμβάνει μετρήσεις ενός ή περισσότερων άλλων χαρακτηριστικών. Για να αποδείξουμε την αναγκαιότητα της μέτρησης, αρκεί να αναφέρουμε ότι οποιαδήποτε πρόοδος σε κάθε επιστήμη ακολούθησε σαν φυσική συνέπεια της προόδου που έγινε σε μονάδες και εργαλεία μετρήσεων.

Η εξασφάλιση της ποιότητας και η χρήση των μετρήσεων ήταν σημαντικοί αρωγοί στην ανάπτυξη της βιομηχανίας τις τελευταίες δεκαετίες. Είναι αδιανόητη η μαζική παραγωγή ενός υλικού χωρίς να τεθούν λεπτομερείς και μετρήσιμοι ποιοτικοί στόχοι και χωρίς έλεγχο των τελικών προϊόντων ως προς την εκπλήρωση των στόχων αυτών. Συγκεκριμένα, κατά την παραγωγή ενός προϊόντος στη βιομηχανία ακολουθείται πρόγραμμα εξασφάλισης ποιότητας, σύμφωνα με το οποίο α) καθορίζονται τα χαρακτηριστικά-ιδιότητες του προϊόντος που θα μετρηθούν, καθώς και των επιθυμητών ορίων, β) καθορίζονται οι διαδικασίες μέτρησης, γ) εντοπίζονται και απορρίπτονται τα προϊόντα που δεν πληρούν τις ποιοτικές προδιαγραφές και δ) βελτιώνεται η διαδικασία παραγωγής ώστε να ελαχιστοποιηθεί ο αριθμός των προϊόντων που απορρίπτονται.

Οι βασικές αρχές της εξασφάλισης ποιότητας λογισμικού είναι ίδιες με την εξασφάλιση ποιότητας κάθε άλλου προϊόντος. Παρόλα αυτά υπάρχει μία βασική αντίθεση που διαμορφώνει την εξής ιδιαιτερότητα κατά την παραγωγή προϊόντων λογισμικού: Στη μαζική παραγωγή καθορίζονται αρχικά οι προδιαγραφές και τα

χαρακτηριστικά ενός πρωτοτύπου και στη συνέχεια παράγονται ποσότητες προϊόντων ίδιων φαινομενικά με το αρχικό πρωτότυπο, με σκοπό να μην αποκλίνουν τα αντίγραφα αυτά από τις αρχικές προδιαγραφές. Αντίθετα, στην περίπτωση του λογισμικού, το τελικό προϊόν (αν εξαιρέσουμε τις διαδικασίες συσκευασίας και το συνοδευτικό υλικό, όπως π.χ. εγχειρίδια και μέσο αποθήκευσης) παράγεται μία μόνο φορά. Τα αντίτυπα που αναπαράγονται στη συνέχεια είναι πιστά αντίγραφα του πρωτοτύπου. Δηλαδή, το πρόβλημα που αντιμετωπίζεται κατά την παραγωγή λογισμικού δεν είναι η δημιουργία αντιτύπων που να μην αποκλίνουν από το αρχικό πρωτότυπο, αλλά η παραγωγή η ίδια αυτού του πρωτοτύπου σύμφωνα με τις αρχικές προδιαγραφές. Με άλλα λόγια, η εξασφάλιση της ποιότητας ενός πρωτοτύπου οδηγεί αυτόματα στην εξασφάλιση της ποιότητας για όλα τα υπόλοιπα αντίγραφα, ενώ κάθε ατέλεια του κληροδοτείται αυτόματα σε όλα τα αντίγραφα [XEN96a].

1.3 Κρίση του Λογισμικού

Ένα μεγάλο πρόβλημα της εξασφάλισης ποιότητας λογισμικού είναι η έλλειψη αυστηρά καθορισμένων μετρήσιμων στόχων και διαδικασιών μέτρησης. Ενώ για παράδειγμα είναι εφικτό και κατανοητό τόσο να ορίσουμε τις επιθυμητές τιμές όσο και να μετρήσουμε το ύψος, τη διάμετρο βάσης, το πάχος κ.τ.λ. ενός μπουκαλιού, κάτι αντίστοιχο δεν μπορεί άμεσα να εφαρμοστεί στην περίπτωση προϊόντων λογισμικού. Ως αποτέλεσμα, δεν μπορούμε να ζητάμε το λογισμικό να έχει υψηλή αξιοπιστία ή υψηλή λειτουργικότητα. Όπως χαρακτηριστικά αναφέρει ο Gustafsson [SQM93], «Όταν ένας εργάτης που δουλεύει σε μία γραμμή μαζικής παραγωγής κάνει ένα λάθος, τότε πετάει το εξάρτημα που έφτιαχνε. Αν το κάνει συχνά, σε λίγο ένας επιστάτης θα θέλει να μάθει γιατί η στοίβα με τα πεταμένα εξαρτήματα είναι τόσο μεγάλη. Αυτό συμβαίνει συνέχεια με το λογισμικό, αλλά κανένας δεν μπορεί να δει τη στοίβα με τα πεταμένα εξαρτήματα».

Η σπουδαιότητα του λογισμικού αναγνωρίστηκε δυστυχώς αρκετά αργά σαν η κινητήρια δύναμη πίσω από το υλικό. Αρχικά το λογισμικό αντιμετωπιζόταν σαν μία ιδέα εκ των υστέρων, αφού είχε ήδη ολοκληρωθεί το υλικό [XEN94]. Η παράληψη των πρώτων εκείνων εποχών της πληροφορικής καθώς και η νοοτροπία που επέβαλε οδήγησαν στη λεγόμενη «κρίση του λογισμικού» και συνίσταται στην αποτυχία των περισσότερων έργων λογισμικού [SOM89]. Η κρίση αυτή προέκυπτε κυρίως λόγω της μεγάλης απαίτησης για καινούρια προγράμματα και λόγω της κακής τους σχεδίασης με αποτέλεσμα να οδηγούνται οι εταιρείες παραγωγής λογισμικού σε αποτυχία είτε τήρησης του αρχικού προϋπολογισμού είτε ικανοποίησης των απαιτήσεων του πελάτη είτε συνήθως και στα δύο.

Αν και θεωρείται αυτονόητο πως ένα προϊόν λογισμικού πρέπει πάντοτε να συμμορφώνεται με αυτό που επιθυμούν οι τελικοί χρήστες του, έρευνες που έχουν διεξαχθεί κατά καιρούς παρουσιάζουν ότι κατά ένα πολύ μικρό ποσοστό ικανοποιείται το παραπάνω [BEL92]. Συγκεκριμένα, μόνο το 2% περίπου των προϊόντων λογισμικού χρησιμοποιήθηκε όπως ακριβώς παραδόθηκε στους πελάτες. Αξιοσημείωτο ακόμα είναι ότι και το ποσοστό των προϊόντων λογισμικού που τελικώς χρησιμοποιήθηκαν ύστερα από ορισμένες αλλαγές είναι επίσης πολύ μικρό (περίπου 5%). Αναλυτικότερα, τα αποτελέσματα αυτών των ερευνών παρουσιάζονται στο Σχήμα 1.1, στο οποίο τα διάφορα έργα λογισμικού έχουν κατηγοριοποιηθεί ανάλογα με την τελική τους κατάληξη. Διαπιστώνει λοιπόν κανείς ότι το ποσοστό των αποτυχημένων έργων είναι πάρα πολύ υψηλό.

Σχήμα 1.1: Αποτελεσματικότητα των προϊόντων λογισμικού

Ο Pressman [PRE97] υποστηρίζει ότι δεν μιλάμε πλέον για κρίση λογισμικού, αλλά για μία «χρόνια πάθηση» που ταλαιπωρεί το λογισμικό περισσότερο από 30 χρόνια. Τα λάθη και πολλές φορές η έλλειψη μίας συγκεκριμένης μεθοδολογίας κατά τη διάρκεια όλων των φάσεων του κύκλου ζωής λογισμικού έχουν ως αποτέλεσμα να απαιτείται μία τεράστια προσπάθεια (αναλογικά με τις άλλες φάσεις) για διορθωτική συντήρηση. Το κόστος συντήρησης απεικονίζεται καλύτερα στο Σχήμα 1.2, όπου παρουσιάζεται ποσοστιαία ο χρόνος που απαιτείται κατά τις διάφορες φάσεις παραγωγής λογισμικού [BEL92].

Σχήμα 1.2: Σχετικά κόστη των φάσεων της παραγωγής λογισμικού

Η λύση που προτείνεται ως θεραπεία για αυτή την «πάθηση» του λογισμικού είναι οι μεθοδολογίες εξασφάλισης ποιότητας λογισμικού. Σύμφωνα με τον DeMacro [DMA99] μία ορθή στρατηγική για τη βελτίωση της ποιότητας λογισμικού πρέπει να προτείνει στις εταιρείες παραγωγής λογισμικού α) να υλοποιούν πολύ λιγότερο

κώδικα, β) να επιλέγουν πολύ προσεκτικότερα τι ακριβώς υλοποιούν και γ) να θέτουν ελαστικότερα όρια ολοκλήρωσης και παράδοσης των έργων τους. Σήμερα υπάρχουν διεθνή πρότυπα, όπως το ISO9000 [ISO00], τα πρότυπα IEEE [IEEE89], τα βραβεία Baldrige [BRO91] [STE93], το Capability Maturity Model (CMM) [PAU93] και το Capability Maturity Model Integration (CMMI) [AHE04], τα οποία καθορίζουν πώς πρέπει να εφαρμοστεί ένα πρόγραμμα εξασφάλισης ποιότητας στη διαδικασία παραγωγής λογισμικού. Τα πρότυπα αυτά παρέχουν βασικούς κανόνες στις εταιρείες παραγωγής λογισμικού και ενθαρρύνουν την εφαρμογή μετρήσεων, χωρίς όμως να προτείνουν συγκεκριμένες λύσεις για τις μετρήσεις. Το βασικότερο πρόβλημα εξάλλου στον τομέα της εξασφάλισης ποιότητας λογισμικού είναι ότι για τα χαρακτηριστικά που επιθυμούμε να μετρήσουμε (και ενδιαφέρουν άμεσα το χρήστη) δεν υπάρχει μία δυνατότητα άμεσης μέτρησης.

1.4 Μοντέλα Ποιότητας

Είναι πλέον κοινώς αποδεκτό ότι η έννοια της ποιότητας λογισμικού είναι αρκετά αφηρημένη ως οντότητα για να κατανοηθεί και να μπορούν να τεθούν μετρήσιμοι στόχοι. Επομένως έγινε φανερό η ανάγκη επιμερισμού της ποιότητας σε συνιστώσες ή χαρακτηριστικά. Τα χαρακτηριστικά αυτά ονομάζονται ποιοτικά χαρακτηριστικά, συνθέτουν την ποιότητα ενός προϊόντος, έχουν την ελάχιστη δυνατή επικάλυψη μεταξύ τους και είναι επαρκή για τη σύνθεση της ποιότητας. Πολλοί ερευνητές, προκειμένου να κατανοηθεί η έννοια της ποιότητας και να μετρηθεί με ένα συστηματικό τρόπο, έχουν δημιουργήσει μοντέλα όπου περιγράφεται πώς σχετίζονται μεταξύ τους τα ποιοτικά χαρακτηριστικά τα οποία την αποτελούν.

Ένα από τα πρώτα μοντέλα που αναπτύχθηκαν ήταν το μοντέλο FCM (Factor-Criteria-Metrics) του McCall [MCC77] για την US Air Force Rome Air Development Center, όπου η ποιότητα αντιμετωπίζεται ως μία ιεραρχία από παράγοντες, καθένας από τους οποίους συσχετίζεται με κάποια μετρική. Ανάλογο, με κάποιες διαφορές, ήταν το μοντέλο του Boehm [BOE78]. Παρόμοια μοντέλα προτάθηκαν μετέπειτα, όπως το μοντέλο GRQM (Goal-Rule-Question-Metrics) [TER96] και το μοντέλο του Dromey [DRO95] [DRO96]. Σήμερα έχει καθοριστεί το πρότυπο ISO9126 [ISO01], όπου η ποιότητα χωρίζεται σε επιμέρους παράγοντες με την ελάχιστη μεταξύ τους επικάλυψη. Παρακάτω παρουσιάζονται το μοντέλο του McCall και το πρότυπο ISO9126.

1.4.1 Μοντέλο του McCall

Το μοντέλο του McCall [MCC77] εστιάζει στο τελικό προϊόν και αναγνωρίζει βασικά γνωρίσματα από την πλευρά του χρήστη. Τα βασικά αυτά γνωρίσματα ονομάζονται **παράγοντες ποιότητας** (quality factors) στους οποίους γίνεται μία πρώτη τμηματοποίηση της ποιότητας. Συγκεκριμένα, οι 11 παράγοντες που προτείνει ο McCall είναι οι:

- Χρησιμότητα (Usability)
- Ακεραιότητα (Integrity)
- Αποδοτικότητα (Efficiency)
- Ορθότητα (Correctness)
- Αξιοπιστία (Reliability)
- Συντηρησιμότητα (Maintainability)
- Ελεγχιμότητα (Testability)

- Ευελιξία (Flexibility)
- Επαναχρησιμοποιησιμότητα (Reusability)
- Μεταφερσιμότητα (Portability)
- Διαλειτουργικότητα (Interoperability)

Σχήμα 1.3: Το μοντέλο του McCall

Οι παράγοντες όμως αυτοί βρίσκονται σε υψηλό επίπεδο αφαίρεσης για να μπορούν να μετρηθούν άμεσα. Για το λόγο αυτό γίνεται μία τμηματοποίησή τους σε **25 κριτήρια ποιότητας** (quality criteria) και τα οποία σχετίζονται περισσότερο με την παραγωγή και το σχεδιασμό λογισμικού. Τέλος, προτείνεται και ένα επιπλέον στάδιο εκλέπτυνσης, κατά το οποίο σε κάθε κριτήριο ποιότητας αντιστοιχίζεται ένας αριθμός από χαμηλού επιπέδου μετρήσεις, που πραγματοποιούνται άμεσα, συλλαμβάνουν κάποια πλευρά του κριτηρίου ποιότητας και ονομάζονται **μετρικές** (metrics). Από τα συστατικά του αυτά στοιχεία το μοντέλο πήρε και την ονομασία FCM (Factors-Criteria-Metrics) και παρουσιάζεται αναλυτικότερα στο Σχήμα 1.3.

1.4.2 Το πρότυπο ISO9126

Το πρότυπο αυτό [ISO01] είναι το αποτέλεσμα της διεθνούς προσπάθειας να αναπτυχθεί ένα πρότυπο για τη μέτρηση της ποιότητας λογισμικού. Είχε ως βάση το μοντέλο του McCall και αρχικά ονομάστηκε “Software Product Evaluation: Quality Characteristics and Guidelines for their Use”, το οποίο αργότερα εξελίχθηκε στο πρότυπο ISO9126.

Σύμφωνα με αυτό, η ποιότητα ορίζεται ως το σύνολο των στοιχείων και των χαρακτηριστικών ενός προϊόντος λογισμικού που επιδρούν στην ικανότητά του να ανταπεξέρχεται και να ικανοποιεί είτε δεδηλωμένες είτε υπονοούμενες ανάγκες [FEN94]. Αρχικά η ποιότητα χωρίζεται σε έξι **ποιοτικά χαρακτηριστικά**. Επίσης, το

πρότυπο περιέχει ένα δείγμα διάσπασης των ποιοτικών αυτών χαρακτηριστικών σε ένα σύνολο από **υπο-χαρακτηριστικά**. Η εκλέπτυνση αυτή δεν είναι δεσμευτική και δηλώνει τη σημασία που αποδίδει το πρότυπο στην περαιτέρω διάσπαση των ποιοτικών χαρακτηριστικών προκειμένου να μετρηθεί η ποιότητα [BAC94].

Τέλος, τα ποιοτικά αυτά υπο-χαρακτηριστικά μετρώνται από **ενδείκτες** (indicators) οι οποίοι κατασκευάζονται από **στοιχεία δεδομένων** (data elements). Η διάσπαση της ποιότητας σε χαρακτηριστικά και υπο-χαρακτηριστικά σύμφωνα με το πρότυπο ISO9126 παρουσιάζεται στο Σχήμα 1.4.

Σχήμα 1.4: Το πρότυπο ISO9126

1.5 Η Έννοια της Μετρικής

Οι μετρήσεις των κατώτερων επιπέδων στο λογισμικό γίνονται, όπως φάνηκε και από την παρουσίαση των παραπάνω μοντέλων, με τη χρήση «μέτρων» και «μετρικών». Σύμφωνα με τον Fenton [FEN92], **μετρική** (ή μέτρο) είναι μία εμπειρική αντικειμενική αντιστοίχιση ενός αριθμού (ή συμβόλου) σε μία οντότητα με στόχο να χαρακτηρίσει ένα συγκεκριμένο χαρακτηριστικό της οντότητας αυτής. Η χρήση μετρικών στην τεχνολογία λογισμικού αντιμετωπίζει την αδυναμία του λογισμικού για σαφή καθορισμό μετρήσιμων ποσοτήτων κατά τη σχεδίαση και ανάπτυξη ενός έργου. Πολλές φορές διάφορες εταιρείες παραγωγής λογισμικού υπόσχονται ότι τα προγράμματά τους είναι π.χ. αξιόπιστα ή φιλικά προς το χρήστη, χωρίς όμως να έχουν προσδιορίσει εκ των προτέρων τι σημαίνει καθένα από αυτά με μετρήσιμες ποσότητες. Έτσι, η αποτυχία είναι φυσιολογική συνέπεια [XEN96a]. Χαρακτηριστικά, όπως αναφέρει ο Gilb [GIL87], «το μόνο ξεκάθαρο για έργα που δεν έχουν ξεκάθαρους στόχους είναι ότι δεν θα πετύχουν τους στόχους τους».

Παρόλο που πολλές εταιρείες αποτυγχάνουν να εκτιμήσουν βασικά ποσοτικά χαρακτηριστικά που σχετίζονται με ένα έργο (όπως π.χ. η αξιοπιστία του συστήματος κάτω από ορισμένες συνθήκες), πολλές φορές παρουσιάζονται να διαφημίζουν για παράδειγμα ότι «η νέα έκδοση ενός προϊόντος τους είναι κατά 50% περισσότερο φιλική προς το χρήστη» ή ότι «επιτυγχάνεται μείωση του χρόνου συντήρησης κατά 80%», χωρίς την παραμικρή ένδειξη πώς προκύπτουν αυτοί οι αριθμοί και με βάση ποιων μετρικών.

Μία πρώτη βασική κατηγοριοποίηση [FEN92] χωρίζει τις μετρήσεις και συνεπώς και τις μετρικές λογισμικού σε:

1. **Μετρικές Διαδικασιών**, που αφορούν κάθε δραστηριότητα που σχετίζεται με την παραγωγή λογισμικού. Συνήθως εμπεριέχουν τον παράγοντα του χρόνου. Παραδείγματα στα οποία εφαρμόζονται: χρόνος εκτέλεσης μίας διαδικασίας, προσπάθεια, αριθμός εμφάνισης γεγονότων ενός συγκεκριμένου τύπου, δυνατότητα ελέγχου κ.ά.
2. **Μετρικές Προϊόντων**, που αφορούν κάθε παραγόμενο, παραδοτέο ή κείμενο που προκύπτει από μία διαδικασία. Παραδείγματα στα οποία εφαρμόζονται: μήκος προγράμματος, τμηματοποίηση, λειτουργικότητα, συνοχή και σύνδεση τμημάτων λογισμικού, εγχειρίδια χρήσης, υπηρεσίες κ.ά.
3. **Μετρικές Πόρων**, που αφορούν οντότητες που αποτελούν δεδομένα εισόδου στις διαδικασίες. Παραδείγματα στα οποία εφαρμόζονται: μέγεθος ομάδων εργασίας, εμπειρία, ηλικία, κόστος, δομή, επικοινωνία, υλικό κ.ά.

Μία άλλη κατηγοριοποίηση των μετρικών λογισμικού γίνεται με βάση τα χαρακτηριστικά του λογισμικού στα οποία αναφέρονται: τα εσωτερικά και τα εξωτερικά. **Εσωτερικά χαρακτηριστικά του λογισμικού** είναι τα χαρακτηριστικά για τα οποία υπάρχει απτή φυσική αντίληψη και άμεσος τρόπος μέτρησής τους (π.χ. γραμμές κώδικα, ποσοστό σχολίων στον κώδικα, αριθμός εντολών διακλάδωσης κ.ά.). Αυτά τα χαρακτηριστικά μετρούνται εύκολα και αυτόματα, αλλά δεν προσφέρουν πληροφορίες υψηλού επιπέδου που σχετίζονται με την ποιότητα του προϊόντος. **Εξωτερικά χαρακτηριστικά του λογισμικού** είναι τα χαρακτηριστικά με υψηλό επίπεδο αφαίρεσης τα οποία συνθέτουν την ποιότητα του λογισμικού (π.χ. λειτουργικότητα, αξιοπιστία, αποδοτικότητα, συντηρησιμότητα κ.ά.). Παρόλο που αυτά τα χαρακτηριστικά δίνουν ακριβείς πληροφορίες για την ποιότητα του λογισμικού, δυστυχώς είναι δύσκολο έως αδύνατον να μετρηθούν άμεσα.

Αντίστοιχα με τα ποιοτικά χαρακτηριστικά λοιπόν ορίζονται και οι μετρικές λογισμικού. Συγκεκριμένα έχουμε τις:

1. **Εσωτερικές μετρικές**, που είναι οι μετρικές αυτές που χρησιμοποιούνται για τη μέτρηση χαρακτηριστικών του λογισμικού για τα οποία υπάρχει απτή αντίληψη για τη φυσική τους σημασία και δυνατότητα άμεσης μέτρησης.
2. **Εξωτερικές μετρικές**, που είναι οι μετρικές αυτές που χρησιμοποιούνται για τη μέτρηση χαρακτηριστικών του λογισμικού με υψηλό επίπεδο αφαίρεσης τα οποία σχετίζονται άμεσα με την ποιότητα του λογισμικού.

Οι εσωτερικές μετρικές μπορούν να υπολογιστούν εύκολα και αυτόματα, αλλά το ζητούμενο δεν είναι τόσο ο τρόπος μέτρησής τους, όσο η σύνδεσή τους με τα εξωτερικά χαρακτηριστικά του λογισμικού τα οποία τελικώς επιθυμούμε να μετρήσουμε. Σύμφωνα με τον Hambling [HAM96], επειδή δεν υπάρχει μία άμεση αντιστοιχία και συσχέτιση ανάμεσα σε εσωτερικά και εξωτερικά χαρακτηριστικά του

λογισμικού, το σκεπτικό αρχικά ήταν να μετρήσουμε οτιδήποτε, με την ελπίδα ότι τουλάχιστον κάποιες από τις μετρήσεις θα είναι χρήσιμες. Από την άλλη πλευρά, οι εξωτερικές μετρικές βασίζονται αποκλειστικά σε έρευνες γνώμης των πελατών-χρηστών με τη βοήθεια συνήθως ερωτηματολογίων και συνεντεύξεων. Αν και σύμφωνα με τον ορισμό της μετρικής που προηγήθηκε, οι εξωτερικές μετρικές δεν είναι κανονικές μετρικές, αρκετοί ερευνητές τις αναφέρουν και τις χρησιμοποιούν ως μετρικές ποιότητας [CON86] [JON91] [KAP95].

Οι μετρικές, εάν επιλεγούν και χρησιμοποιηθούν σωστά, μας επιτρέπουν να καθορίσουμε ποσοτικά την επιτυχία ή την αποτυχία κάποιου αντικειμένου, διαδικασίας ή εργαζομένου και να αναγνωρίσουμε τη βελτίωση ή τη χειροτέρευση που συντελέστηκε. Τέτοια στοιχεία που χρειάζεται να μετράμε ή να καθορίζουμε είναι τα εξής: κόστος διαδικασιών, παραγωγικότητα προσωπικού, παράγοντες που επηρεάζουν το κόστος και την παραγωγικότητα, ποιότητα προϊόντων, καθορισμός μετρήσιμων στόχων, αποδοτικότητα μεθόδων, επίβλεψη, ποιοτικές προδιαγραφές, επιβεβαίωση και πρόβλεψη. Τα παραπάνω στοιχεία είναι σημαντικά για τον υπεύθυνο διαχείρισης (manager) μίας επιχείρησης, γιατί τον βοηθούν να αντιληφθεί τι συμβαίνει στο εσωτερικό της επιχείρησής του και να λάβει εγκαίρως τις κατάλληλες αποφάσεις [FEN94].

1.6 Στοιχεία ενός Προγράμματος Εξασφάλισης Ποιότητας

Τα όσα αναφέρθηκαν μέχρι στιγμής για την ποιότητα, τα μοντέλα που την περιγράφουν και τις μετρικές που χρησιμοποιούνται δεν θα είχαν κανένα απολύτως νόημα αν δεν μπορούσαν να χρησιμοποιηθούν στην πράξη, δηλαδή κατά την ανάπτυξη ενός προϊόντος λογισμικού. Το ρόλο αυτό της επιλογής των εξωτερικών ποιοτικών χαρακτηριστικών του προϊόντος που η εξασφάλισή τους ενδιαφέρει την εφαρμογή και την αντιστοίχιση των κατάλληλων μετρικών για τον έλεγχό τους αναλαμβάνει το **πρόγραμμα εξασφάλισης ποιότητας** [INC93]. Τα βασικά στοιχεία του απεικονίζονται στο Σχήμα 1.5.

Το πρόγραμμα εξασφάλισης ποιότητας [INC94] καθορίζεται βάσει κάποιου **γενικού προτύπου (διεθνές standard)** που παρέχει οδηγίες για την εφαρμογή του. Παραδείγματα τέτοιων προτύπων αποτελούν η σειρά ISO9000, τα IEEE standards, τα βραβεία Baldrige και το CMM. Ένα πλήρες πρόγραμμα εξασφάλισης ποιότητας διακρίνεται σε τρεις κατηγορίες:

- i) στην εξασφάλιση ποιότητας διαδικασιών, που ασχολείται με τη συνεχή βελτιστοποίηση βάσει των προδιαγραφών ποιότητας, των διαδικασιών παραγωγής του προϊόντος και γενικότερα τη λειτουργία της επιχείρησης,
- ii) στην εξασφάλιση ποιότητας πόρων, που ασχολείται με τη βελτίωση των πόρων που χρησιμοποιούνται για την παραγωγή του προϊόντος και
- iii) στην εξασφάλιση ποιότητας προϊόντων, που ασχολείται άμεσα με τη βελτίωση της ποιότητας του τελικού προϊόντος.

Είναι προφανές ότι η ποιότητα του τελικού προϊόντος, πέρα από άμεσο στόχο της τρίτης κατηγορίας, αποτελεί και έμμεσο στόχο των άλλων δύο. Το πρόγραμμα εξασφάλισης ποιότητας αποτελείται από τις δομές, τις δραστηριότητες, τις αρμοδιότητες, τις διαδικασίες, τους πόρους, τις μετρικές και τα εργαλεία μέτρησης που χρησιμοποιούνται για να εξασφαλίσουν ότι τα έργα λογισμικού που αναπτύσσονται θα εκπληρώνουν τους ποιοτικούς παράγοντες οι οποίοι είναι

επιθυμητοί τόσο από τον πελάτη όσο και από την επιχείρηση. Οι παραπάνω λεπτομέρειες συντάσσονται στο **εγχειρίδιο ποιότητας** του προγράμματος εξασφάλισης ποιότητας.

Σχήμα 1.5: Πρόγραμμα Εξασφάλισης Ποιότητας

Από τη συγκέντρωση των παραπάνω ποιοτικών χαρακτηριστικών, στην εξασφάλιση των οποίων επικεντρώνει το ενδιαφέρον του κάθε έργο προκύπτει το **πλάνο εξασφάλισης ποιότητας έργου**. Αυτό είναι συγκεκριμένο για κάθε έργο που αναλαμβάνει να υλοποιήσει μία εταιρεία και πρέπει να αποφασίζεται από κοινού από το Διαχειριστή Ποιότητας (Quality Manager) και το Διαχειριστή κάθε έργου (Project Manager). Όμως, εκτός από τα ειδικά ποιοτικά χαρακτηριστικά, στο πλάνο εξασφάλισης ποιότητας θα πρέπει να ενσωματώνονται και οι διαδικασίες και οι μετρικές ελέγχου που στοχεύουν στην εξασφάλιση της «ελάχιστης αποδεκτής ποιότητας», όπως αυτή καθορίζεται από το εγχειρίδιο ποιότητας. Δηλαδή, το σύνολο των ποιοτικών χαρακτηριστικών που θα πρέπει να βρίσκονται οπωσδήποτε πάνω από κάποια όρια, ανεξάρτητα από τα υπόλοιπα χαρακτηριστικά στα οποία δίνει έμφαση το πλάνο εξασφάλισης ποιότητας του συγκεκριμένου έργου [XEN96a].

1.7 Ο Ρόλος του Πελάτη-Χρήστη

Είναι πλέον γνωστό ότι σε μία εταιρεία παραγωγής λογισμικού ο προγραμματισμός δεν πρέπει να θεωρείται ως μία μορφή τέχνης, αλλά ως τεχνολογία λογισμικού που διέπεται από βασικούς κανόνες, αρχές και μεθοδολογίες [PRE97]. Ακολουθώντας αυτούς τους κανόνες καθόλη τη διάρκεια του κύκλου ζωής λογισμικού, η επιτυχία ενός έργου δεν είναι πλέον ένα αποτέλεσμα που τυχαίνει, αλλά ένα αποτέλεσμα που πετυχαίνεται. Οι αρχές της εξασφάλισης ποιότητας

αποτελούν ένα αναπόσπαστο κομμάτι της τεχνολογίας λογισμικού και θα πρέπει να εφαρμόζονται σε όλες τις φάσεις παραγωγής λογισμικού και όχι περιστασιακά.

Ένας από τους βασικότερους, αν όχι ο βασικότερος στόχος, μίας εταιρείας λογισμικού είναι η αύξηση της πελατείας της και των πωλήσεών της γενικότερα. Από αυτά συνεπάγονται η αύξηση των κερδών της εταιρείας, η αύξηση του κύρους της, η βελτίωση και η ασφάλεια που προκύπτουν κατά τον καθορισμό της μακροπρόθεσμης στρατηγικής της κ.ά. Για να επιτευχθεί η αύξηση της πελατείας, μία επιχείρηση μπορεί να επιλέξει μία από τις παρακάτω δράσεις:

Πρώτον, αποφασίζει να δημιουργήσει ένα πρωτότυπο προϊόν που δεν υπάρχει στην αγορά και διαβλέπει ότι θα χαίρει θετικής εκτίμησης και αποδοχής από τους τελικούς χρήστες. Όμως, κάτι τέτοιο δεν μπορεί να επιτευχθεί εύκολα, γιατί χρειάζεται συνήθως ένα μεγάλο αποθεματικό για επένδυση και έρευνα. Παράλληλα, για να είναι εφικτό να περιοριστεί το ρίσκο της τελικής επιτυχίας, θα πρέπει η εταιρεία να ενστερνιστεί και να ακολουθήσει βασικούς κανόνες εξασφάλισης ποιότητας. Διαφορετικά είναι πολύ πιθανόν να υλοποιήσει ένα ομολογουμένως αξιόλογο έργο που όμως κανένας δεν θα το χρειάζεται.

Δεύτερον, αποφασίζει να δαπανήσει μεγαλύτερα ποσά για τη διαφήμιση και προώθηση των προϊόντων της (marketing). Όμως, όσο και αν ισχυρίζεται μία εταιρεία πως τα δικά της προϊόντα είναι καλύτερα και ποιοτικότερα από αντίστοιχα του εμπορίου, αν αυτό δεν έχει επιτευχθεί μέσα από τις διαδικασίες ενός προγράμματος εξασφάλισης ποιότητας, γρήγορα θα αποδειχτεί το αντίθετο.

Τρίτον, η εταιρεία αποφασίζει πραγματικά να διαθέσει υψηλότερα κονδύλια για την εξασφάλιση ποιότητας των προϊόντων της, αποβλέποντας στη δημιουργία προϊόντων λογισμικού με υψηλότερες προδιαγραφές στα ποιοτικά χαρακτηριστικά που ενδιαφέρουν τόσο την ίδια την εταιρεία όσο και κυρίως τους πελάτες της. Έχει μάλιστα διαφανεί σε πολλές εταιρείες που ακολουθούν πιστά ένα πρόγραμμα εξασφάλισης ποιότητας ότι παρόλο που αρχικά η επένδυση σε μεθόδους εξασφάλισης ποιότητας φαίνεται να κοστίζει περισσότερο στην εταιρεία, μακροπρόθεσμα επιτυγχάνεται ένα σημαντικό οικονομικό κέρδος κατά τη διάρκεια της παραγωγής του προϊόντος [ARN95].

1.7.1 Διαφορετικές όψεις για την ποιότητα

Όπως είδαμε στην παράγραφο 1.1, η εκπλήρωση των απαιτήσεων του πελάτη λαμβάνεται πάντα υπόψη στον ορισμό της ποιότητας. Όμως, παρόλο που νέες μεθοδολογίες και τεχνικές έχουν αναπτυχθεί και εφαρμοστεί κατά τη διάρκεια της διαδικασίας παραγωγής λογισμικού, αλλά και τη διαχείριση των έργων, οι τελικοί χρήστες εξακολουθούν συχνά να αντιμετωπίζουν προβλήματα με το επίπεδο της ποιότητας των προϊόντων λογισμικού που απορρέουν [NOR93]. Αυτό το αποτέλεσμα οφείλεται κυρίως στις διαφορές που υπάρχουν ανάμεσα στους χρήστες και τους παραγωγούς λογισμικού όσον αφορά την αντίληψή τους για την ποιότητα, αλλά και στη διαφορετική σπουδαιότητα και προτεραιότητα που δίνει ο καθένας από αυτούς στα διάφορα ποιοτικά χαρακτηριστικά [AND96].

Η ποιότητα, όπως για παράδειγμα η εξωτερική ομορφιά, εξαρτάται από το «μάτι» του παρατηρητή και κρίνεται σύμφωνα με ένα σύνολο επιθυμητών χαρακτηριστικών τα οποία έχουν διαφορετική βαρύτητα από παρατηρητή σε παρατηρητή. Έτσι λοιπόν, τουλάχιστον δύο όψεις της ποιότητας λογισμικού μπορούν να παρατηρηθούν: Αυτή του κατασκευαστή και αυτή του χρήστη. Σήμερα, η συνεχής ροπή προς ποιοτικές βελτιώσεις αλλά και τα συνεχώς νέα προϊόντα λογισμικού καλλιεργούν ένα κλίμα

εστίασης προς τη διαδικασία παραγωγής. Για παράδειγμα, στην τεχνολογία λογισμικού η ποιότητα συχνά θεωρείται ότι αντιπροσωπεύει την δυνατότητα παραγωγής ενός προϊόντος που ικανοποιεί τις προδιαγραφές που έχουν τεθεί από την επιχείρηση όσον αφορά θέματα χρονοδιαγράμματος και προϋπολογισμού [LYY88]. Η έμφαση δηλαδή δινόταν στη βελτίωση του **πώς** ένα προϊόν παράγεται και όχι στην ερώτηση του **τι** θα παραχθεί, για **ποιον** και, μερικές φορές, **γιατί** θα παραχθεί. Οι παραπάνω ερωτήσεις είναι οι πλέον κατάλληλες και αυτές που αφορούν άμεσα τους χρήστες και τις απαιτήσεις τους.

Σύμφωνα με τον Deming [DEM82], ο απόλυτος και θεμελιώδης στόχος κάθε κερδοσκοπικής εταιρείας πρέπει να είναι η παραγωγή προϊόντων που ικανοποιούν τον τελικό χρήστη και μάλιστα με ένα κόστος που ο χρήστης είναι προδιαθετιμένος να πληρώσει. Επομένως, θα πρέπει τόσο οι ανάγκες των χρηστών, όσο και ο τρόπος που αυτοί αντιλαμβάνονται την ποιότητα ενός προϊόντος να διαγνωστούν εγκαίρως και να αντιμετωπιστούν ως προδιαγραφές. Όπως λέει και το γνωστό απόφθεγμα: «Ο πελάτης έχει πάντα δίκιο».

1.7.2 Αντικείμενο της διδακτορικής διατριβής

Ο πιο άμεσος τρόπος για να μετρηθεί η γνώμη των χρηστών για την ποιότητα ενός προϊόντος λογισμικού είναι να διεξαχθεί μία έρευνα (π.χ. με ερωτηματολόγια ή συνεντεύξεις) κατά την οποία θα ρωτηθούν, αν όχι όλοι, σίγουρα ένας αξιόλογος αριθμός χρηστών (εξωτερική μέτρηση). Η έρευνα αυτή θα πρέπει να γίνει προφανώς μετά την τελική παράδοση του προϊόντος και κατόπιν ενός εύλογου χρόνου χρήσης του από όλους τους χρήστες που θα συμμετέχουν στην έρευνα. Αν και αυτός είναι ο εγκυρότερος τρόπος για να αποφανθούμε για την ποιότητα του λογισμικού όπως την αντιλαμβάνεται ο χρήστης, τα βασικά προβλήματα που προκύπτουν είναι πρώτον ότι το κόστος μίας τέτοιας έρευνας μπορεί να γίνει πολύ μεγάλο, ειδικά σε περιπτώσεις που δεν είναι σωστά δομημένη, καθώς τότε θα απαιτούνται για αυτήν αρκετές ανθρωποώρες τόσο από πλευράς εταιρείας λογισμικού όσο και από πλευράς των ίδιων των χρηστών, και δεύτερον ότι η έρευνα αυτή ενδέχεται να διεξαχθεί πολύ αργά.

Το ερώτημα λοιπόν που προκύπτει είναι πώς μπορούμε να βρούμε κάποιες ενδείξεις για την ποιότητα του λογισμικού μας; Πώς μπορούμε, δηλαδή, να μετρήσουμε, να προβλέψουμε, να εκτιμήσουμε τη γνώμη του πελάτη πριν το τελικό στάδιο του κύκλου ζωής λογισμικού; Είναι εύκολα αντιληπτό ότι αν μία εταιρεία παραγωγής λογισμικού μπορεί εγκαίρως να παρατηρήσει και να κατανοήσει τις ελλείψεις και τα ελαττώματα που πιθανώς επιτελέστηκαν κατά τη διαδικασία παραγωγής, τότε μπορεί με διορθωτικές ενέργειες να βελτιώσει την ποιότητα των προϊόντων της. Κάτι τέτοιο θα αύξανε το κέρδος της, αφού οι παραπάνω διορθώσεις θα μπορούσαν να διεξαχθούν πριν από τη φάση της συντήρησης, μειώνοντας κατά πολύ το κόστος αλλαγής [PRE97].

Στην παρούσα διδακτορική διατριβή παρουσιάζονται διάφορες μέθοδοι και τεχνικές που συνεισφέρουν στην έγκαιρη εκτίμηση της γνώμης των χρηστών για την ποιότητα λογισμικού. Τέτοιες μέθοδοι είναι α) η επιλογή και χρήση μετρικών λογισμικού και η ανάλυση των αποτελεσμάτων τους, β) η δόμηση και η ανάλυση ερωτηματολογίων για τη μέτρηση της γνώμης των χρηστών για την ποιότητα λογισμικού, γ) η συσχέτιση εσωτερικών μετρικών λογισμικού με τα εξωτερικά ποιοτικά χαρακτηριστικά ενός προγράμματος λογισμικού, δ) η μεταβολή της γνώμης του χρήστη για την ποιότητα ενός προγράμματος λογισμικού με την πάροδο του

χρόνου σε σχέση με το επίπεδο εμπειρίας του χρήστη και ε) η χρήση και η προσαρμογή κανόνων και μοντέλων από τη θεωρία Αναθεώρησης Άποψης. Όλες οι παραπάνω μέθοδοι παρουσιάζονται αναλυτικά και, τέλος, συνδυάζονται και συνεισφέρουν στην προτεινόμενη μεθοδολογία της διδακτορικής διατριβής.

Η μεθοδολογία αυτή επικεντρώνεται τόσο στον τομέα των εσωτερικών όσο και στον τομέα των εξωτερικών μετρήσεων. Έχει στηριχθεί σε πρωτότυπες δημοσιεύσεις σε διεθνή συνέδρια και περιοδικά. Συγκεκριμένα η δημοσίευση [XEN00] αποτελεί μία έρευνα κατά την οποία έγινε συλλογή, ομαδοποίηση και χαρακτηρισμός αντικειμενοστραφών μετρικών λογισμικού. Οι δημοσιεύσεις [XEN96b], [STA99b] παρουσιάζουν περιπτώσιολογικές μελέτες στις οποίες αναλύονται η συσχέτιση εσωτερικών μετρικών και εξωτερικών ποιοτικών χαρακτηριστικών που ενδιαφέρουν άμεσα τον τελικό χρήστη η πρώτη και η συσχέτιση εσωτερικών μετρικών και συντηρησιμότητας η δεύτερη. Στη δημοσίευση [XEN99] παρουσιάζεται μέσω ενός εργαλείου η μέθοδος δόμησης και ανάλυσης ερωτηματολογίων για τη μέτρηση της γνώμης των χρηστών για την ποιότητα λογισμικού. Οι δημοσιεύσεις [STA99a] και [STA00] αφορούν τη μεταβολή της γνώμης του χρήστη για την ποιότητα ενός προγράμματος λογισμικού με την πάροδο του χρόνου σε σχέση με το επίπεδο εμπειρίας του χρήστη, ενώ παράλληλα γίνεται χρήση και προσαρμογή κανόνων και μοντέλων από τη θεωρία Αναθεώρησης Άποψης. Τέλος, στη δημοσίευση [STA05] γίνεται η παρουσίαση της προτεινόμενης μεθοδολογίας. Παράλληλα, διεξήχθηκε έρευνα και σε παρεμφερείς τομείς, όπως η μοντελοποίηση της ποιότητας συστημάτων ηλεκτρονικού εμπορίου, που παρουσιάζεται στις δημοσιεύσεις [STF03], [STF04] και [STF05], αλλά και σε θέματα εξ αποστάσεως εκπαίδευσης [XEN04a], [XEN04b] και [XEN04c].

1.8 Δομή της διδακτορικής διατριβής

Στα επόμενα κεφάλαια παρουσιάζεται η έρευνα που έγινε στα πλαίσια της διατριβής, περιγράφονται τα αποτελέσματα που βρέθηκαν από τις διάφορες μελέτες, αναλύονται τα συμπεράσματα που προέκυψαν και τέλος περιγράφεται η προτεινόμενη μεθοδολογία. Σε ορισμένα σημεία της διατριβής αναφέρονται σύντομα βασικά θέματα θεωρίας και προϋπάρχουσας βιβλιογραφίας, τα οποία θα βοηθήσουν τον αναγνώστη στην καλύτερη κατανόηση της διατριβής.

Στο δεύτερο κεφάλαιο περιγράφεται η χρήση των εσωτερικών μετρικών λογισμικού. Αναφέρονται τα πλεονεκτήματα και τα μειονεκτήματά τους, καθώς και ορισμένες διαδεδομένες και ευρέως χρησιμοποιούμενες μετρικές. Παρουσιάζεται το εργαλείο ATHENA που χρησιμοποιήθηκε στα πλαίσια της διατριβής και αναφέρονται μερικά γνωστά εναλλακτικά περιβάλλοντα μετρήσεων. Περιγράφονται οι μεταμετρικές και ο τρόπος χρήσης τους. Τέλος, παρουσιάζεται μία έρευνα σε αντικειμενοστραφείς μετρικές λογισμικού.

Στο τρίτο κεφάλαιο περιγράφονται οι εξωτερικές μετρικές λογισμικού, όπου αναφέρονται τα πλεονεκτήματα και τα μειονεκτήματά τους σε σύγκριση με τις εσωτερικές μετρικές. Περιγράφεται ο ρόλος του πελάτη στις διάφορες φάσεις του κύκλου ζωής λογισμικού. Παραθέτονται και κατηγοριοποιούνται οι γνωστότερες τεχνικές μέτρησης και μέθοδοι ανάλυσης των αποτελεσμάτων. Τέλος, παρουσιάζεται το εργαλείο CESM που αναπτύχθηκε και χρησιμοποιήθηκε στα πλαίσια της διατριβής.

Στο τέταρτο κεφάλαιο γίνεται μελέτη της συσχέτισης ανάμεσα στις εσωτερικές και τις εξωτερικές μετρικές λογισμικού και παρουσιάζονται τα αποτελέσματα από μία περιπτώσιολογική μελέτη και τα συμπεράσματα που προέκυψαν. Αντίστοιχα, στο πέμπτο κεφάλαιο γίνεται παρόμοια μελέτη της συσχέτισης ανάμεσα στις εσωτερικές μετρικές και τη συντηρησιμότητα, όπου πέρα από τις διάφορες έννοιες της συντηρησιμότητας και τις προσεγγίσεις μέτρησής της, παρουσιάζονται τα αποτελέσματα από δύο περιπτώσιολογικές μελέτες και τα συμπεράσματα που προέκυψαν.

Στο έκτο κεφάλαιο παρουσιάζονται τα αποτελέσματα από διαδοχικές εξωτερικές μετρήσεις και αναλύεται ο τρόπος μεταβολής της γνώμης των χρηστών για την ποιότητα του λογισμικού με την πάροδο του χρόνου. Αναφέρονται οι ανάγκες της μοντελοποίησης των απόψεων των διαφορετικών κατηγοριών από χρήστες και περιγράφονται τα βήματα για την κατασκευή ενός μοντέλου έγκαιρης εκτίμησης της γνώμης των χρηστών, ώστε να μην είναι απαραίτητη η διεξαγωγή διαδοχικών ερευνών με την πάροδο του χρόνου.

Στο έβδομο κεφάλαιο παρουσιάζονται οι βασικές αρχές από τη θεωρία Αναθεώρησης Άποψης που χρησιμοποιήθηκαν και προσαρμόστηκαν στην παρούσα διατριβή. Αποδεικνύεται η δυνατότητα της μοντελοποίησης της γνώμης των χρηστών για την ποιότητα λογισμικού και των πιθανών αναθεωρήσεων που γίνονται στη γνώμη αυτή κατά τη διάρκεια χρήσης του λογισμικού. Τέλος, περιγράφονται και αποδεικνύονται θεωρήματα σχετικά με θέματα υπολογιστικής πολυπλοκότητας, τα οποία προέκυψαν από την προσαρμογή και τη χρήση των αρχών αυτών στα πλαίσια της διατριβής.

Έχοντας λοιπόν περιγράψει αναλυτικά τις διάφορες μεθόδους και τεχνικές που αναπτύχθηκαν ή ακολουθήθηκαν, στο όγδοο κεφάλαιο γίνεται μία σύνθεσή τους σε ακολουθιακά στάδια από την οποία προκύπτει η προτεινόμενη μεθοδολογία. Γίνεται μία σαφής καταγραφή των προδιαγραφών της μεθοδολογίας και του τρόπου χρήσης της από μία εταιρεία παραγωγής λογισμικού. Τέλος, αναφέρονται τα πλεονεκτήματα που προκύπτουν από την ενσωμάτωση και εφαρμογή της στα πλαίσια ενός γενικότερου προγράμματος εξασφάλισης ποιότητας λογισμικού.

Κεφάλαιο**2**

Χρήση Εσωτερικών Μετρικών Λογισμικού

Σε αυτό το κεφάλαιο γίνεται αναφορά στην επιλογή και χρήση εσωτερικών μετρικών λογισμικού. Παρουσιάζονται τα πλεονεκτήματα και τα μειονεκτήματά τους, καθώς και μερικά ευρέως διαδεδομένα παραδείγματα μετρικών που μπορούν να εφαρμοστούν σε όλες τις γνωστές γλώσσες προγραμματισμού. Περιγράφεται το εργαλείο Athena, ένα περιβάλλον για διεξαγωγή μετρήσεων που χρησιμοποιήθηκε στην παρούσα διδακτορική διατριβή. Γίνεται αναφορά στον τρόπο χαρακτηρισμού εσωτερικών μετρικών λογισμικού μέσω μεταμετρικών. Τέλος, παρουσιάζεται μία περιπτωσιολογική μελέτη συλλογής, ομαδοποίησης και χαρακτηρισμού μέσω μεταμετρικών των αντικειμενοστραφών μετρικών λογισμικού.

2.1 Πλεονεκτήματα και Μειονεκτήματα

Όπως ήδη έχει αναφερθεί, οι εσωτερικές μετρικές λογισμικού χρησιμοποιούνται για τη μέτρηση χαρακτηριστικών του λογισμικού για τα οποία υπάρχει απτή αντίληψη για τη φυσική τους σημασία και δυνατότητα άμεσης μέτρησης. Η χρήση εσωτερικών μετρικών λογισμικού ενθαρρύνεται από όλα τα διεθνή πρότυπα. Για παράδειγμα, στο κεφάλαιο 4.11a του ISO9001 [ISO00] αναφέρεται ότι πρέπει «να καθορίζονται οι μετρήσεις που πρέπει να γίνουν, η ακρίβεια που απαιτείται και να επιλέγονται τα κατάλληλα εργαλεία για επιθεώρηση, μέτρηση και έλεγχο». Τα πρότυπα αυτά παρέχουν βασικούς κανόνες στις εταιρείες παραγωγής, χωρίς όμως να προτείνουν συγκεκριμένες λύσεις για τις μετρήσεις. Έτσι, στο ISO9001 δεν υπάρχουν περιγραφές συγκεκριμένων μετρικών, αφού το ISO9001 αναφέρεται περισσότερο στις διαδικασίες παραγωγής κάθε προϊόντος [HAN97].

Η μέτρηση εσωτερικών χαρακτηριστικών του λογισμικού μπορεί να γίνει πολύ εύκολα, γρήγορα και αυτόματα. Προφανώς, δεν είναι δύσκολη η κατασκευή ενός προγράμματος-εργαλείου που θα μετράει για παράδειγμα τον αριθμό των ρουτινών ενός τμήματος (module) κάποιου κώδικα ή τον αριθμό των διακριτών μεταβλητών που χρησιμοποιούνται σε μία ρουτίνα. Η διαδικασία μέτρησης μπορεί να πραγματοποιηθεί με ελάχιστη ανθρώπινη συμμετοχή, βασισμένη αποκλειστικά στον κώδικα του προγράμματος. Επομένως, η συχνότητα λαθών κατά τη διάρκεια αυτής της διαδικασίας είναι μηδαμινή και αμελητέα και τα αποτελέσματα των εσωτερικών μετρήσεων θεωρούνται έγκυρα και αντικειμενικά.

Το κόστος διεξαγωγής εσωτερικών μετρήσεων είναι ασφαλώς πολύ μικρό, αφού τον αντικείμενο της μέτρησης απαρτίζεται από απτά χαρακτηριστικά του κώδικα. Τέλος, ένα ακόμα βασικό πλεονέκτημα αυτών των μετρήσεων είναι ότι μπορούν να διεξαχθούν πριν από την ολοκλήρωση και έκδοση του προϊόντος, γεγονός που δίνει τη δυνατότητα έγκαιρων διορθωτικών αλλαγών, πριν παραδοθεί δηλαδή το λογισμικό στον πελάτη.

Παρόλα αυτά, οι εσωτερικές μετρικές δεν προσφέρουν συνήθως πληροφορίες υψηλού επιπέδου που σχετίζονται με την ποιότητα του προϊόντος. Τα δεδομένα που προκύπτουν δεν είναι άμεσα αξιοποιήσιμα και ερμηνεύσιμα, αφού παρουσιάζουν μία αδυναμία σύνδεσης με τα εξωτερικά ποιοτικά χαρακτηριστικά του λογισμικού. Ο τρόπος μέτρησης δηλαδή σε αυτήν την περίπτωση είναι έμμεσος, γιατί οι εσωτερικές μετρικές χρησιμοποιούνται για να εξαχθούν ή να προβλεφθούν συμπεράσματα για τα εξωτερικά χαρακτηριστικά, μιας που αυτά περιγράφουν και απαρτίζουν την αντίληψη του χρήστη για την ποιότητα του λογισμικού. Επιπλέον, για πολλές μετρικές έχουν διατυπωθεί αρκετές απόψεις σχετικές με τη δυνατότητα συσχέτισής τους με τα εξωτερικά ποιοτικά χαρακτηριστικά που επιθυμούμε να διαμορφώσουμε άποψη ή και κατά πόσο τα εξωτερικά αυτά χαρακτηριστικά αντιστοιχούν μόνο στις προτεινόμενες μετρικές.

2.2 Διαδεδομένες Μετρικές

Στη βιβλιογραφία σήμερα υπάρχουν εκατοντάδες μετρικές. Μία βασική τους κατηγοριοποίηση, όπως έχει ήδη αναφερθεί, είναι σε μετρικές **πόρων**, **διαδικασιών** και **προϊόντων**. Η παρούσα διδακτορική διατριβή εστιάζει στις μετρικές προϊόντων, οι οποίες ως γνωστόν αφορούν κάθε παραγόμενο που προκύπτει από μία διαδικασία, αφού αυτές μπορούν να συσχετισθούν ευκολότερα με την αντίληψη του χρήστη για την ποιότητα. Μία επιπλέον κατηγοριοποίηση των μετρικών προϊόντων γίνεται σε μετρικές **μεγέθους** (όπως οι μετρικές της επιστήμης του λογισμικού του Halstead [HAL75]), **δομής** (όπως η μετρική πολυπλοκότητας του McCabe [MCB76]) και **δεδομένων** (όπως η μετρική πολυπλοκότητας δομών δεδομένων του Tsai [TSA86]).

2.2.1 Μετρικές της επιστήμης λογισμικού

Οι μετρικές της επιστήμης λογισμικού του Halstead [HAL75] μετρούν αριθμήσιμα στοιχεία του λογισμικού σχετιζόμενα με το μέγεθος του πηγαίου κώδικα. Οι τέσσερις βασικές μετρήσιμες ποσότητες που προτείνει είναι οι ακόλουθες:

- n_1 ο αριθμός των διακριτών τελεστών που εμφανίζονται στο πρόγραμμα.
- n_2 ο αριθμός των διακριτών εντέλων που εμφανίζονται στο πρόγραμμα.
- N_1 ο αριθμός των συνολικών εμφανίσεων τελεστών στο πρόγραμμα.
- N_2 ο αριθμός των συνολικών εμφανίσεων εντέλων στο πρόγραμμα.

Από τα τέσσερα παραπάνω προκύπτουν και οι υπόλοιπες μετρικές της επιστήμης λογισμικού που είναι οι εξής:

• Λεξιλόγιο προγράμματος (n)	$n = n_1 + n_2$
• Μήκος προγράμματος (N)	$N = N_1 + N_2$
• Εκτιμητής μήκους προγράμματος (N_{est})	$N_{est} = n_1 \cdot \log_2 n_1 + n_2 \cdot \log_2 n_2$
• Όγκος προγράμματος (V)	$V = N \cdot \log_2 n$
• Δυναμικός όγκος προγράμματος (V^*) που αναφέρεται στην πιο συνοπτική θεωρητική υλοποίηση του προγράμματος	

- Επίπεδο προγράμματος (L) $L = V^*/V$
- Εκτιμητής επιπέδου προγράμματος (L_{est}) $L_{est} = 2 \cdot n_2 / n_1 \cdot N_2$
- Επίπεδο γλώσσας στο πρόγραμμα (λ) $\lambda = L^2 \cdot V$
- Δυσκολία προγράμματος (D) $D = 1/L$
- Προσπάθεια υλοποίησης προγράμματος (E) $E = DV$
- Χρόνος υλοποίησης προγράμματος (T) $T = E/S$, (S ο αριθμός Stroud)
- Εκτίμηση αριθμού λαθών στο πρόγραμμα (B) $B = E^{2/3}/E_0$, ($E_0 \in [3000, 3200]$)

Με τη χρήση των μετρικών του Halstead μπορούν εύκολα να διαγνωσθούν τυχόν προγραμματιστικές ατέλειες σε ένα πρόγραμμα λογισμικού. Συγκεκριμένα, ρουτίνες στις οποίες θα εμφανίζονταν συμπληρωματικοί τελεστές, διφορούμενα ή συνώνυμα έντελα, αγρείαστες καταχωρήσεις, όμοιες υποεκφράσεις και μη παραγοντοποιημένες εκφράσεις θα έδιναν μικρά αποτελέσματα στις παραπάνω μετρικές [XEN91] [STA95].

2.2.2 Μετρική κυκλωματικής πολυπλοκότητας

Η μετρική κυκλωματικής πολυπλοκότητας του McCabe είναι μία μετρική δομής και μετράει πόσο πολύπλοκος είναι ο γράφος ροής του προγράμματος [MCB76]. Επικεντρώνει το ενδιαφέρον της στα σημεία του προγράμματος που μπορεί να ληφθεί απόφαση και σχετίζεται με τα πιθανά μονοπάτια που μπορεί να ακολουθήσει η ροή του προγράμματος. Η μετρική αυτή ονομάζεται και κυκλωματικός αριθμός $V(G)$, όπου G ο γράφος του προγράμματος, και δίνεται από την παρακάτω σχέση:

$$V(G) = e - n + 2 \cdot p, \quad \begin{array}{l} \text{όπου } e \text{ ο αριθμός των ακμών του γράφου,} \\ n \text{ ο αριθμός των κόμβων του γράφου και} \\ p \text{ ο αριθμός των συνεκτικών συνιστωσών του γράφου.} \end{array}$$

Στον παραπάνω γράφο, που είναι ένας απλοποιημένος γράφος ελέγχου, οι κόμβοι είναι τα σημεία απόφασης και οι ακμές είναι οι εναλλακτικές δράσεις. Προφανώς, όσο μεγαλύτερος είναι ο κυκλωματικός αριθμός μίας ρουτίνας, τόσο πιο πολύπλοκη είναι αυτή. Σε περιπτώσεις ρουτινών με πολύ μεγάλο κυκλωματικό αριθμό συνιστάται να γίνεται διάσπασή τους σε περισσότερες.

2.2.3 Μετρική πολυπλοκότητας δομών δεδομένων

Η μετρική της πολυπλοκότητας δομών δεδομένων του Tsai [TSA86] εφαρμόζεται στα δεδομένα του προγράμματος, επομένως το μόνο που χρειάζεται είναι να υπάρχουν οι λεπτομερείς κατάλογοι των δεδομένων που θα επεξεργαστεί το πρόγραμμα και όχι η τελική έκδοση του κώδικα του προγράμματος.

Σύμφωνα με τον αλγόριθμο εφαρμογής αυτής της μετρικής, κάθε δομή δεδομένων D είναι της μορφής: $D::F(D_1, \dots, D_n)$, με $n \geq 1$, όπου F ο τελεστής σχηματισμού της δομής δεδομένων, D_i οι αναφορές σε δομές δεδομένων που χρησιμοποιήθηκαν για το σχηματισμό της D και n ο αριθμός των D_i . Τα επόμενα βήματα είναι α) η κατασκευή ενός κατευθυνόμενου γράφου, στον οποίο η κάθε δομή δεδομένων αναπαρίσταται με έναν κόμβο, β) η μείωση του γράφου, κατά την οποία αφαιρούνται κόμβοι με μοναδική είσοδο και μοναδική έξοδο και γ) ο εντοπισμός των ισχυρά συνεκτικών συνιστωσών.

Στη συνέχεια υπολογίζονται τα μονώνυμα $S(K)$ των συνεκτικών συνιστωσών βάσει του τύπου $S(K) = (V + E)x^L$, όπου K η συνιστώσα, V ο αριθμός των κόμβων της, E ο αριθμός των ακμών της και L ο αριθμός των απλών κυκλικών μονοπατιών μέσα σε αυτήν. Τέλος, υπολογίζονται τα πολώνυμα πολυπλοκότητας $C(K)$ βάσει του

τύπου $C(K) = \sum_{\forall K_i \leftarrow K} [C(K_i)] + S(K)$ όπου ορίζοντας τη σχέση διάταξης $\leftarrow$, λέμε ότι

$K_m \leftarrow K_n$ με $K_m \neq K_n$ αν και μόνο αν υπάρχει μονοπάτι από κάποιον κόμβο της K_n σε κάποιον κόμβο της K_m .

Προφανώς, όσο μεγαλύτερου βαθμού είναι τα πολυώνυμα που προκύπτουν, τόσο πιο πολύπλοκες είναι οι δομές δεδομένων του προγράμματος. Με τη χρήση αυτής της μετρικής είναι δυνατή η μείωση της πολυπλοκότητας των προγραμμάτων λογισμικού αρκετά πριν την ολοκλήρωσή τους. Με άλλα λόγια, μπορούν να προληφθούν τα προβλήματα πριν δημιουργηθούν, αντί να εντοπίζονται ήδη υπάρχοντα προβλήματα και να αναζητείται η μέθοδος διόρθωσής τους.

2.2.4 Άλλες γνωστές μετρικές

Στο μεγάλο πλήθος των μετρικών λογισμικού που έχουν προταθεί και έχουν παρουσιαστεί σε διάφορα βιβλία, όπως του Fenton [FEN97], του Shepperd [SHE93], της Zahedi [ZAH95] κ.ά., μπορεί να διαπιστώσει κάποιος ότι αρκετές από αυτές είναι γενικές και μπορούν να εφαρμοστούν σε οποιαδήποτε γλώσσα προγραμματισμού, ενώ άλλες είναι ειδικές, εστιάζουν και έχουν νόημα εφαρμογής σε συγκεκριμένο μόνο στυλ προγραμματισμού. Για παράδειγμα, όσον αφορά τον αντικειμενοστραφή προγραμματισμό, ο μηχανικός ποιότητας μπορεί να χρησιμοποιήσει όχι μόνο γενικές, αλλά και αντικειμενοστραφείς μετρικές λογισμικού.

Οι μετρικές που παρουσιάστηκαν παραπάνω ανήκουν στην πρώτη κατηγορία. Άλλα γνωστά παραδείγματα εσωτερικών μετρικών λογισμικού αυτής της κατηγορίας είναι τα εξής: ‘μετρική γραμμών κώδικα’, ‘μετρική γραμμών σχολίων προς γραμμές κώδικα’ [CON86], ‘μετρικές για λειτουργικά σημεία’ [ALB79], ‘μετρικές τμημάτων’ [BAS79], ‘μετρικές ζωντανών μεταβλητών’ [DUN79], ‘μετρική ελαχίστου αριθμού μονοπατιών’ [SCH79], ‘μετρικές διασταυρώσεων’ [WOO79], ‘μετρικές εισόδων – εξόδων’ [HEN81], ‘μετρικές πολυπλοκότητας’ [HAS84], ‘μετρική ουσιώδους πολυπλοκότητας’ [FEN97]. Οι μετρικές αυτές είναι αξιοποιήσιμες για κάθε γλώσσα προγραμματισμού που βασίζεται σε πηγαίο κώδικα και δεν σχετίζονται με ειδικά χαρακτηριστικά κάποιας συγκεκριμένης γλώσσας προγραμματισμού.

2.3 Εργαλεία Εσωτερικών Μετρήσεων

Όπως προαναφέρθηκε, ένα από τα πλεονεκτήματα των εσωτερικών μετρικών είναι ότι η μέτρησή τους αποτελεί μία διαδικασία που μπορεί εύκολα να αυτοματοποιηθεί. Τις τελευταίες δύο δεκαετίες έχει γίνει εμφανής η επιθυμία για την ανάπτυξη ανάλογων εργαλείων λογισμικού που θα υποστηρίζουν έναν αριθμό από εσωτερικές μετρικές λογισμικού και θα διεξάγουν μετρήσεις εύκολα, γρήγορα, αυτόματα και χωρίς ιδιαίτερο ανθρώπινο κόστος. Η επιθυμία αυτή έχει προκύψει από την ανάγκη διεξαγωγής μετρήσεων σε μεγάλο βαθμό δεδομένων στη βιομηχανία παραγωγής λογισμικού, κυρίως κατά τη φάση της υλοποίησης και του ελέγχου [FEN92]. Αρκεί να σημειωθεί ότι ο υπολογισμός της μετρικής συντηρησιμότητας των Yau και Collofello [YAU80] απαιτεί περίπου δέκα λεπτά για κάθε τμήμα κώδικα (module) αν γίνει «με το χέρι». Επομένως, η παραπάνω μέτρηση σε όλο το σύστημα λογισμικού, που μπορεί και να αποτελείται από εκατοντάδες τέτοια τμήματα είναι ανέφικτη και απαγορευτική. Τα εργαλεία μετρήσεων όχι μόνο μειώνουν σημαντικά το κόστος της μέτρησης, αλλά και τα αποτελέσματα που προκύπτουν δεν χρειάζονται περαιτέρω έλεγχο για την ορθότητά τους.

Μερικά χαρακτηριστικά παραδείγματα από εργαλεία εσωτερικών μετρήσεων είναι το QUALMS [FEN92] και το METRICS [ZUS90] που εστιάζουν στις ιεραρχικές μετρήσεις, το PRO-QUEST (John Bell Systems) και το TESTBED (Program Analysers Ltd) που αφορούν γραφικές μετρήσεις, ενώ επίσης έχουν αναπτυχθεί και διάφορα εργαλεία ελέγχου, όπως τα OLIVER και SIMON (Advanced Programming Techniques Ltd), το PATHVU (XA Systems Ltd) και το VAX Test Manager (DEC). Τέλος, δύο πολύ γνωστά εργαλεία μετρήσεων που χρησιμοποιούνται και ευρέως στο εμπόριο είναι το LOGISCOPE της γαλλικής εταιρείας Verilog και το QUALIGRAPH της γερμανικής εταιρείας Computer Kontor.

2.3.1 Το εργαλείο ATHENA

Το λογισμικό εργαλείο ATHENA [TSL90, TSL91] χρησιμοποιήθηκε για τη διεξαγωγή εσωτερικών μετρήσεων στην παρούσα διδακτορική διατριβή. Στην πραγματικότητα, πρόκειται για ένα παραμετρικό περιβάλλον που η φιλοσοφία του είναι προσανατολισμένη στη μέτρηση των ποιοτικών χαρακτηριστικών του λογισμικού. Η μέτρηση αυτή βασίζεται στη συντακτική και σημασιολογική ανάλυση των προγραμμάτων. Το ATHENA είναι ένα εργαλείο ανεξαρτήτου γλώσσας και εύκολα τροποποιήσιμο έτσι, ώστε οι προδιαγραφές της κάθε γλώσσας καθώς και οι αλγόριθμοι μέτρησης να δίνονται σαν είσοδος στο εργαλείο, το οποίο στη συνέχεια παράγει αυτόματα το περιβάλλον μέτρησης για τη δοθείσα γλώσσα.

Προς το παρόν, το ATHENA υποστηρίζει τις γλώσσες Pascal, ADA και C και παράλληλα παρέχει τη δυνατότητα μέτρησης ενός συνόλου από μετρικές πυρήνα (δηλαδή τις μετρικές της Επιστήμης Λογισμικού του Halstead, τη μετρική κυκλωματικής πολυπλοκότητας του McCabe και τη μετρική πολυπλοκότητας δομών δεδομένων του Tsai). Επίσης, μία ειδικά κατασκευασμένη γλώσσα προδιαγραφών επιτρέπει την άνετη κατασκευή και κωδικοποίηση των μετρικών. Αυτός είναι ένας από τους βασικούς λόγους που το εργαλείο ATHENA είναι ιδιαίτερα χρήσιμο για τον έλεγχο της συντήρησης του λογισμικού. Τέλος, ένα ακόμα χρήσιμο χαρακτηριστικό του είναι η παραγωγή αναφοράς, όπου γράφονται επακριβώς τα αποτελέσματα των μετρήσεων για κάθε υπορουτίνα του προγράμματος εισόδου.

Ένα βασικό μέρος του ATHENA είναι ο επεξεργαστής μετρικών (metric processor – mp) που είναι το εκτελέσιμο πρόγραμμα το οποίο δέχεται ως είσοδο το αρχείο πηγαίου κώδικα (π.χ. σε γλώσσα C) και παράγει τα αποτελέσματα των μετρικών αυτού του κώδικα. Στην πραγματικότητα, στην αρχή γίνεται μία προεπεξεργασία (precompiling) των προγραμμάτων εισόδου και το αποτέλεσμα αυτής της προεπεξεργασίας δίνεται ως είσοδος στον επεξεργαστή μετρικών από την προκαθορισμένη είσοδο (standard input), ενώ τα αποτελέσματα των μετρικών παράγονται στην προκαθορισμένη έξοδο (standard output) [STA95]. Στο Σχήμα 2.1 παρατίθεται μία γραφική παράσταση του περιβάλλοντος ATHENA.

Χρησιμοποιείται ένας επεξεργαστής γραμματικών (gp), ο οποίος δέχεται ως είσοδο το αρχείο καθορισμού της γλώσσας (LSF) και παράγει τον επεξεργαστή μετρικών (mp) για τη δοθείσα γλώσσα. Στη συνέχεια, ο mp δέχεται ως είσοδο το προς εξέταση πρόγραμμα (program) και παράγει ένα αρχείο αποτελεσμάτων (RESULTS). Αυτό το αρχείο μπορεί να γίνει είσοδος του επεξεργαστή αναφορών (rp), που θα παράγει μία γραπτή αναφορά για τις μετρήσεις, του επεξεργαστή γραφικών (gxp), που θα δώσει μία γραφική παράσταση των μετρήσεων και τέλος, του επεξεργαστή βάσης δεδομένων (dbp), που θα αποθηκεύσει τα αποτελέσματα των μετρήσεων σε βάση δεδομένων για μετέπειτα στατιστική ανάλυση.

Σχήμα 2.1: Το περιβάλλον του ATHENA

Ο *mp* παράγει ακόμα άλλα δύο αρχεία: Το πρώτο (*ppp*) περιέχει μία κομψά τυπωμένη έκδοση του εισαγόμενου προγράμματος. Το δεύτερο (*DEBUG*) περιέχει μία λίστα των ενεργειών που έγιναν κατά τη διάρκεια της ανάλυσης του προγράμματος από το συντακτικό αναλυτή. Δηλαδή, αυτό το αρχείο μπορεί να χρησιμοποιηθεί για την εύρεση συντακτικών λαθών κατά τη διάρκεια της ανάλυσης.

Σχήμα 2.2: Ο μηχανισμός μέτρησης

Ο μηχανισμός της μέτρησης με το εργαλείο ATHENA, το επαναλαμβανόμενο δηλαδή τμήμα της μεθόδου που είναι υπεύθυνο για τις εσωτερικές μετρήσεις, απεικονίζεται στο Σχήμα 2.2. Ο μηχανισμός δέχεται ως είσοδο ένα τμήμα κώδικα (module) το οποίο αποτελείται από μία σειρά από διαδικασίες (υπορουτίνες ή συναρτήσεις). Σε κάθε μία από τις διαδικασίες εφαρμόζονται οι μετρικές του Halstead

και η μετρική του McCabe και εξάγονται πρωτογενή αποτελέσματα, δηλαδή οι τιμές των παραπάνω μετρικών για κάθε μία διαδικασία. Αντίστοιχα, εφαρμόζεται η μετρική του Tsai στα δεδομένα του τμήματος και εξάγονται τα πολυώνυμα που προκύπτουν για κάθε δομή δεδομένων που περιλαμβάνεται στο τμήμα. Στη συνέχεια, γίνεται μία ανάλυση των αποτελεσμάτων που προέκυψαν, η οποία θα οδηγήσει σε εξαγωγή συμπερασμάτων για την ποιότητα του κώδικα. Ειδικότερα, μπορούν να εντοπιστούν τμήματα του κώδικα που ενδέχεται να περιέχουν έναν υψηλό αριθμό λαθών και να δημιουργήσουν προβλήματα στις επόμενες φάσεις του κύκλου ζωής του λογισμικού. Τα χαμηλά αποτελέσματα στις εσωτερικές μετρικές σε ένα τμήμα του κώδικα μπορούν, ορισμένες φορές, να οδηγήσουν ακόμα και στην επανασχεδίαση και επανακατασκευή του.

2.4 Μεταμετρικές

Τα τελευταία χρόνια ένα μεγάλο πλήθος από μετρικές λογισμικού έχουν προταθεί, με αποτέλεσμα οι κατασκευαστές λογισμικού να οδηγούνται εύλογα κάποιες φορές σε σύγχυση, έχοντας να αντιμετωπίσουν έναν τόσο μεγάλο αριθμό από μετρήσεις. Επιπλέον, μερικές προσπάθειες έχουν γίνει από ερευνητές για τη συγχώνευση αυτού του μεγάλου όγκου των μετρικών έτσι, ώστε να προκύψει τελικά ένα μικρό σύνολο από αντιπροσωπευτικές μετρικές για κάθε κατηγορία. Ένα πρώτο βήμα όμως που πρέπει αρχικά να γίνεται είναι η αξιολόγηση των ίδιων των μετρικών λογισμικού.

Στο εγχειρίδιο των μετρικών που δημοσιεύτηκε από την U.S. Air Force [AIR91] [ZAH95], αναφέρονται τα βασικά χαρακτηριστικά που πρέπει να έχει μία μετρική. Σύμφωνα λοιπόν με αυτό το εγχειρίδιο, μία καλή μετρική πρέπει να:

- έχει κάποια σημασία για τον πελάτη
- εμπεριέχει τους στόχους της επιχείρησης
- είναι απλή, κατανοητή, λογική και επαναληπτική
- είναι σαφώς και ξεκάθαρα ορισμένη
- είναι ικανή να παρουσιάσει την τάση της επιχείρησης
- είναι έγκαιρη και οικονομική στη συλλογή αποτελεσμάτων
- καθοδηγεί σε κατάλληλες δράσεις

Έτσι λοιπόν διαφαίνεται ότι η αξιολόγηση τόσο του λογισμικού όσο και των μετρικών του έχουν ένα κοινό σημείο: την ύπαρξη αναλόγων κριτηρίων αξιολόγησης. Τα κριτήρια που τίθενται για να χαρακτηρίσουν την καταλληλότητα των μετρικών λογισμικού ονομάζονται μεταμετρικές. Σε μία πρώτη προσπάθεια καθορισμού ενός συνόλου τέτοιων κριτηρίων προτάθηκαν οι παρακάτω μεταμετρικές [CON86]:

- **Απλότητα (Simplicity):** Μετράει τη δυνατότητα και την ευκολία να οδηγήσουν τα αποτελέσματα μίας μετρικής σε ερμηνεύσιμα συμπεράσματα.
- **Εγκυρότητα (Validity):** Εξετάζει την ικανότητα μίας μετρικής να μετράει τελικά αντικειμενικά αυτό που στοχεύει να μετρήσει.
- **Ανθεκτικότητα (Robustness):** Εξετάζει αν μία μετρική είναι ευαίσθητη σε αλλαγές παραγόντων που δεν επηρεάζουν την απόδοση του λογισμικού.
- **Καθοδηγισιμότητα (Prescriptiveness):** Εξετάζει την ικανότητα μίας μετρικής να χρησιμεύσει στη δημιουργία ενός πλάνου βασισμένου στις μετρήσεις και τα αποτελέσματά της έτσι, ώστε να μπορεί να καθοριστεί μία μεθοδολογία δράσεων έναντι στο λογισμικό.

- **Αναλυσιμότητα (Analysability):** Μετράει τη δυνατότητα ανάλυσης των αποτελεσμάτων μίας μετρικής ώστε να μπορούν να εξαχθούν συμπεράσματα στατιστικής μορφής με στόχο την επαγωγή γενικότερων συμπερασμάτων.

Μία άλλη αντιμετώπιση των μετρικών έγινε το 1988 από την Weyuker [WEY88], η οποία περιγράφει κάποιες πιο αφηρημένες ιδιότητες των καλών μετρικών. Αν λοιπόν με $m(P)$ συμβολίζεται το αποτέλεσμα μίας μετρικής m για το πρόγραμμα P , τότε οι ιδιότητες που προτείνονται είναι οι εξής:

- **Διακριτικότητα (Bob-coarseness):** Δοθέντος ενός προγράμματος P , υπάρχει και ένα άλλο πρόγραμμα Q τέτοιο, ώστε $m(P) \neq m(Q)$.
- **Μη μοναδικότητα (Non-uniqueness):** υπάρχουν προγράμματα P και Q τέτοια, ώστε $m(P) = m(Q)$.
- **Η αναδιάταξη είναι σημαντική (Permutation is significant):** Υπάρχουν προγράμματα P και Q τέτοια, ώστε το Q να είναι μία αναδιάταξη του P και $m(P) \neq m(Q)$.
- **Η υλοποίηση είναι σημαντική (Implementation is important):** Υπάρχουν προγράμματα P και Q τέτοια, ώστε να παράγουν την ίδια έξοδο για την ίδια είσοδο και $m(P) \neq m(Q)$.
- **Μονοτονία (Monotonicity):** Για όλα τα προγράμματα P και Q πρέπει να ισχύει $m(P) \leq m(P+Q)$ και $m(Q) \leq m(P+Q)$, όπου $P+Q$ είναι η συνένωση των P και Q .
- **Μη ισοδυναμία στην αλληλεπίδραση (Non-equivalence of interaction):** Υπάρχουν προγράμματα P, Q και R τέτοια, ώστε να ισχύει $m(P) = m(Q)$ και $m(P+R) \neq m(Q+R)$.
- **Η αλληλεπίδραση αυξάνει την πολυπλοκότητα (Interaction increases complexity):** Υπάρχουν προγράμματα P και Q για τα οποία να ισχύει $m(P) + m(Q) \leq m(P+Q)$.
- **Πεπερασμένος αριθμός προγραμμάτων (Finiteness):** Δοθέντος ενός μη αρνητικού αριθμού c , υπάρχει πεπερασμένος αριθμός προγραμμάτων που να ισχύει $m(P) = c$.
- **Η αρχή του Rose (Rose principle):** Αν το πρόγραμμα P είναι μία μετονομασία του προγράμματος Q , τότε πρέπει $m(P) = m(Q)$.

Τέλος, μία άλλη μεταμετρική που προστέθηκε αργότερα ως ένας καθοριστικός παράγοντας για την αξία μίας μετρικής και την αναγκαιότητα στατιστικών σχέσεων ανάμεσα στη μετρική και την προσπάθεια ανάπτυξης λογισμικού είναι η **αυτοματοποιησιμότητα**, η ικανότητα δηλαδή αυτόματης συλλογής των δεδομένων της μετρικής [KOL93].

Το ερώτημα όμως που παραμένει είναι ποιες είναι οι καταλληλότερες μετρικές που πρέπει να επιλέξει ένας κατασκευαστής λογισμικού, ανεξάρτητα από το προϊόν λογισμικού που υλοποιεί. Στο μεγάλο πλήθος των εσωτερικών μετρικών που έχουν προταθεί μπορεί κάποιος να παρατηρήσει αρκετές μετρικές που τα αποτελέσματά τους παρουσιάζουν συνήθως μεγάλο βαθμό συσχέτισης μεταξύ τους. Για παράδειγμα, η μετρική “γραμμές κώδικα” έχει μεγάλη συσχέτιση με τη μετρική “μήκους” του Halstead, αν και δεν είναι απαραίτητα απόλυτο ότι η αύξηση ή η μείωση της μίας ακολουθείται πάντοτε από αύξηση ή μείωση αντίστοιχα της άλλης. Επίσης, παρατηρούνται συχνά και αντιμαχόμενες μετρικές, μιας που για να πετύχει ο κατασκευαστής υψηλότερο αποτέλεσμα σε μία, οδηγείται σε χαμηλότερο αποτέλεσμα σε άλλη.

Από διάφορες έρευνες που έχουν γίνει κατά καιρούς διαφαίνεται ότι είναι τελικά εφικτός ο καθορισμός ενός ελάχιστου συνόλου από αντιπροσωπευτικές εσωτερικές μετρικές οι οποίες μπορούν να εφαρμοστούν σε κάθε προϊόν λογισμικού, αφού τα αποτελέσματά τους είναι πάντοτε ερμηνεύσιμα και καθοδηγήσιμα. Για παράδειγμα, οι Khoshgoftaar και Munson [KHO90] πρότειναν τον παράγοντα control, ο οποίος προκύπτει από το εξής άθροισμα:

$$\text{control} = a_1\text{HNK} + a_2\text{PRC} + a_3\text{E} + a_4\text{VG} + a_5\text{MMC} + a_6\text{Error} + a_7\text{HNP} + a_8\text{LOC}$$

όπου HNK η μετρική πολυπλοκότητας ροής πληροφορίας των Henry και Kafura, PRC ο αριθμός των ρουτινών, E η μετρική προσπάθειας υλοποίησης του Halstead, VG η μετρική κυκλωματικής πολυπλοκότητας του McCabe, MMC η μετρική πολυπλοκότητας του Harrison, Error ο αριθμός των λαθών (defects) και LOC ο αριθμός γραμμών του κώδικα, ενώ τα a_i είναι τα αντίστοιχα βάρη των παραπάνω μετρικών.

Παρόλα αυτά, ένα τέτοιο ελάχιστο σύνολο από αντιπροσωπευτικές μετρικές δεν είναι από μόνο του επαρκές για τη διεξαγωγή μετρήσεων σε διάφορα εξειδικευμένα στυλ προγραμματισμού. Σε αυτές τις περιπτώσεις το παραπάνω σύνολο θα πρέπει να εμπλουτιστεί από έναν επιπρόσθετο αριθμό μετρικών που έχουν προταθεί για το συγκεκριμένο στυλ ή ακόμα και για τη συγκεκριμένη γλώσσα προγραμματισμού. Οι επιπρόσθετες αυτές μετρικές δεν έχουν συνήθως νόημα εφαρμογής σε διαφορετικό είδος προγραμματισμού. Έτσι για παράδειγμα, σε περιπτώσεις προϊόντων λογισμικού που ακολουθούν τη φιλοσοφία του αντικειμενοστραφή προγραμματισμού θα πρέπει, πέρα από τη διεξαγωγή μετρήσεων σύμφωνα με ένα σύνολο κλασικών μετρικών (όπως η μετρική του McCabe), να χρησιμοποιηθεί και ένας αριθμός από αντικειμενοστραφείς μετρικές (όπως το βάθος του δένδρου κληρονομικότητας). Η επιλογή των εξειδικευμένων μετρικών θα πρέπει να γίνεται ακολουθώντας τις προδιαγραφές που προτείνουν οι μεταμετρικές.

2.5 Έρευνα Αντικειμενοστραφών Μετρικών Λογισμικού

Η παρούσα έρευνα που παρουσιάζεται σε αυτήν την ενότητα [XEN00] αφορά μετρικές λογισμικού που προτάθηκαν ειδικά για αντικειμενοστραφείς μετρήσεις και δεν μπορούν να εφαρμοστούν σε διαφορετικό στυλ προγραμματισμού. Οι μετρικές που συλλέχτηκαν [ZIK00] χαρακτηρίστηκαν με βάση γνωστές μεταμετρικές της βιβλιογραφίας, εστιάζοντας περισσότερο στην εφαρμογή τους σε γνωστές αντικειμενοστραφείς γλώσσες προγραμματισμού, όπως Object Pascal, C++ και Java. Η έρευνα αυτή μπορεί μεν να μην καλύπτει πλήρως όλες τις αντικειμενοστραφείς μετρικές που έχουν προταθεί μέχρι σήμερα, αλλά περιλαμβάνει τουλάχιστον όλες τις πιο γνωστές και πιο χαρακτηριστικές. Εξάλλου, ο βασικός σκοπός της δεν ήταν η αναφορά όλων των μετρικών, αλλά η παροχή βοήθειας στη συλλογή ενός κατάλληλου συνόλου μετρικών, ανάλογα με το προϊόν λογισμικού.

2.5.1 Παρουσίαση Μετρικών

Για μεγαλύτερη ευκολία στην παρουσίαση, οι μετρικές χωρίστηκαν σε πέντε διαφορετικές κατηγορίες, χωρίς αυτό να σημαίνει απαραίτητα ότι μία μετρική δεν μπορεί να ανήκει σε περισσότερες από μία κατηγορία. Οι κατηγορίες αυτές είναι οι εξής: μετρικές κλάσης, μεθόδου, σύνδεσης, κληρονομικότητας και συστήματος. Οι μετρικές κάθε κατηγορίας παρουσιάζονται παρακάτω αλφαβητικά, σύμφωνα με τη συντόμευσή τους.

1. Μετρικές Κλάσης

- AHF** (Attribute Hiding Factor) είναι ο λόγος του αθροίσματος των κληρονομημένων κατηγορημάτων σε όλες τις κλάσεις του συστήματος προς το συνολικό αριθμό των κατηγορημάτων των διαθέσιμων κλάσεων [MOR88].
- CCO** (Class Cohesion) μετράει τις σχέσεις ανάμεσα στις κλάσεις [CHI91].
- CEC** (Class Entropy Complexity) μετράει την πολυπλοκότητα των κλάσεων βασισμένη στην πληροφορία που περιλαμβάνουν [BAN97].
- CLM** (Comment Lines per Method) μετράει το ποσοστό των σχολίων στις μεθόδους [LOR94].
- DAM** (Data Access Metric) είναι ο λόγος του αριθμού των ιδιωτικών κατηγορημάτων προς το συνολικό αριθμό κατηγορημάτων που έχουν δηλωθεί σε μία κλάση [BAN97].
- FOC** (Function Oriented Code) μετράει το ποσοστό του μη αντικειμενοστραφή κώδικα στο πρόγραμμα [LOR94].
- INP** (Internal Privacy) αναφέρεται στη χρήση συνεργών λειτουργιών μέσα σε μία κλάση [CHI94].
- LCM** (Lack of Cohesion between Methods) ορίζει το επίπεδο συνοχής ανάμεσα στις μεθόδους [CHI94].
- MAA** (Measure of Attribute Abstraction) είναι ο λόγος του αριθμού των κατηγορημάτων που κληρονομούνται από μία κλάση προς το συνολικό αριθμό κατηγορημάτων της κλάσης [BAN97].
- MFA** (Measure of Functional Abstraction) είναι ο λόγος του αριθμού των μεθόδων που κληρονομούνται από μία κλάση προς το συνολικό αριθμό των προσβάσιμων μεθόδων της κλάσης [BAN97].
- MHF** (Method Hiding Factor) υπολογίζει το λόγο του αθροίσματος των προστατευόμενων και ιδιωτικών μεθόδων που έχουν οριστεί σε όλες τις κλάσεις προς το συνολικό αριθμό μεθόδων που έχουν οριστεί σε όλες τις κλάσεις [HAR98].
- NAD** (Number of Abstract Data types) είναι ο αριθμός των αντικειμένων που έχουν οριστεί και χρησιμοποιούνται ως κατηγορήματα σε μία κλάση τα οποία είναι απαραίτητα για να καθορίσουν ένα στιγμιότυπο της κλάσης [BAN97].
- NCM** (Number of Class Methods in a class) μετράει τις μεθόδους που είναι διαθέσιμες σε μία κλάση, αλλά όχι στα στιγμιότυπά της [LOR94].
- NIV** (Number of Instance Variables in a class) μετράει τις σχέσεις μίας κλάσης με άλλα αντικείμενα του προγράμματος [LOR94].
- NOA** (Number Of Ancestors) είναι ο συνολικός αριθμός των προγόνων μίας κλάσης [KOL93].
- NPA** (Number of Public Attributes) είναι ο αριθμός των κοινών κατηγορημάτων μίας κλάσης [BAN97].
- NPM** (Number of Parameters per Method) είναι ο μέσος αριθμός παραμέτρων των μεθόδων μίας κλάσης [BAN97].

- NRA** (Number of Reference Attributes) είναι ο αριθμός των δεικτών και των αναφορών που χρησιμοποιούνται ως κατηγορήματα σε μία κλάση [BAN97].
- PCM** (Percentage of Commented Methods) είναι το ποσοστό των σχολιασμένων μεθόδων [LOR94].
- PDA** (Public Data) μετράει την πρόσβαση στα κοινά και προστατευμένα δεδομένα μίας κλάσης [MCB94b].
- PMR** (Percent of Potential Method uses actually Reused) είναι το ποσοστό των πραγματικών χρήσεων των μεθόδων [MOR88].
- PPD** (Percentage of Public Data) είναι το ποσοστό των κοινών δεδομένων μίας κλάσης [MCB94b].
- RFC** (Response For a Class) είναι ο αριθμός των μεθόδων που τίθενται σε λειτουργία όταν ένα μήνυμα στέλνεται από μία κλάση [SHA93].
- WCS** (Weighted Class Size) αναφέρεται στον αριθμό των προγόνων και το συνολικό μέγεθος των μεθόδων της κλάσης [KOL93].
- WMC** (Weighted Methods per Class) είναι το άθροισμα των βαρών όλων των μεθόδων μίας κλάσης [MCB94a].

2. Μετρικές Μεθόδου

- AMC** (Average Method Complexity) είναι ο μέσος όρος της κυκλωματικής πολυπλοκότητας για όλες τις μεθόδους [MOR88].
- AMS** (Average Method Size) είναι ο μέσος όρος του μεγέθους των μεθόδων του προγράμματος [LOR94].
- MAG** (MAX V(G)) είναι η μέγιστη κυκλωματική πολυπλοκότητα των μεθόδων μίας κλάσης [MCB94b].
- MCX** (Method Complexity) σχετίζει την πολυπλοκότητα με τον αριθμό των μηνυμάτων [DRE89].

3. Μετρικές Σύνδεσης

- CBO** (Coupling Between Objects) μετράει τον αριθμό των κλάσεων με τις οποίες συνδέεται μία κλάση [FEN97].
- CCP** (Class Coupling) μετράει τις συνδέσεις ανάμεσα στις κλάσεις με βάση τα μηνύματα που ανταλλάσσουν [BOY93].
- CFA** (Coupling Factor) είναι ο λόγος του μέγιστου αριθμού συνδέσεων στο σύστημα προς τον πραγματικό αριθμό των μη κληρονομικών συνδέσεων [HAR98].

4. Μετρικές Κληρονομικότητας

- AIF** (Attribute Inheritance Factor) είναι ο λόγος του αθροίσματος όλων των κληρονομημένων κατηγορημάτων σε όλες τις κλάσεις προς το συνολικό αριθμό των διαθέσιμων κατηγορημάτων όλων των κλάσεων [BAS96].
- DIT** (Depth of Inheritance Tree) μετράει τον αριθμό των προγόνων μίας κλάσης [SHI97].
- FEF** (Factoring Effectiveness) είναι ο λόγος του αριθμού των μοναδικών μεθόδων προς το συνολικό αριθμό μεθόδων [MOR88].

FIN	(FAN-IN) είναι ο αριθμός των κλάσεων από όπου προέρχεται μία κλάση [MCB94b].
HNL	(Class Hierarchy Nesting Level) μετράει το βάθος της ιεραρχίας όπου εντοπίζεται κάθε κλάση [LOR91].
MIF	(Method Inheritance Factor) είναι ο λόγος του αθροίσματος όλων των κληρονομημένων μεθόδων σε όλες τις κλάσεις προς το συνολικό αριθμό των διαθέσιμων μεθόδων για όλες τις κλάσεις [HAR98].
MRE	(Method Reuse metrics) ορίζει το επίπεδο επαναχρησιμοποίησης μίας μεθόδου [ABR94a].
NMI	(Number of Methods Inherited) μετράει τον αριθμό μεθόδων που κληρονομεί μία κλάση [LOR94].
NMO	(Number of Methods Overridden) είναι ο αριθμός των μεθόδων που πρέπει να επαναοριστούν από την κλάση που τις κληρονομεί [LOR94].
NOC	(Number Of Children) είναι ο συνολικός αριθμός των παιδιών μίας κλάσης [ROS97].
PFA	(Polymorphism Factor) είναι ο λόγος του πραγματικού αριθμού των διαφορετικών πιθανών πολυμορφικών καταστάσεων μίας κλάσης προς το μέγιστο αριθμό των διαφορετικών πιθανών πολυμορφικών καταστάσεων της κλάσης [ABR94b].
PMO	(Percent of Potential Method uses Overridden) είναι το ποσοστό των μεθόδων που υπερκαλύπτονται [MOR88].
RDB	(Ratio between Depth and Breadth) είναι ο λόγος του βάθους προς το πλάτος στην ιεραρχία των κλάσεων [BEN99].
RER	(Reuse Ratio) είναι ο λόγος του αριθμού των υπερκλάσεων προς το συνολικό αριθμό των κλάσεων [ROS97].
SIX	(Specialisation Index) μετράει τον τύπο της ειδίκευσης [LOR94].
SPR	(Specialisation Ratio) είναι ο λόγος του αριθμού των υποκλάσεων προς τον αριθμό των υπερκλάσεων [ROS97].

5. Μετρικές Συστήματος

ADI	(Average Depth of Inheritance) είναι ο λόγος του αθροίσματος των εμφωλιασμένων επιπέδων όλων των κλάσεων προς τον αριθμό των κλάσεων [BAN97].
ANA	(Average Number of Ancestors) είναι ο μέσος αριθμός των προγόνων όλων των κλάσεων [BAN97].
APG	(Application Granularity) είναι ο λόγος του συνολικού αριθμού των αντικειμένων προς το συνολικό αριθμό των λειτουργικών σημείων [MOR88].
ASC	(Association Complexity) μετράει την πολυπλοκότητα της δομής των συσχετίσεων στο σύστημα [KOL93].
CAN	(Category Naming) κατηγοριοποιεί τις κλάσεις σε σημασιολογικές ομάδες που έχουν σημασία [CHI94].

- CRE** (Number of time a Class is Reused) μετράει τις αναφορές σε μία κλάση και τον αριθμό των εφαρμογών που επαναχρησιμοποιούν αυτή την κλάση [ABR94a].
- FDE** (Functional Density) είναι ο λόγος του αριθμού των γραμμών κώδικα προς τον αριθμό των λειτουργικών σημείων [FEN97].
- NCT** (Number of Classes Thrown away) μετράει πόσες φορές μία κλάση απορρίπτεται μέχρι τελικώς να γίνει αποδεκτή [WES92].
- NOH** (Number Of Hierarchies) είναι ο αριθμός των διακριτών ιεραρχιών του συστήματος [KOL93].
- OLE** (Object Library Effectiveness) είναι ο λόγος του συνολικού αριθμού επαναχρησιμοποιήσεων αντικειμένων προς το συνολικό αριθμό των αντικειμένων της βιβλιοθήκης [MOR88].
- PRC** (Problem Reports per Class) υπολογίζει τα λάθη που διαπιστώνονται σε κάθε κλάση [LOR94].
- PRO** (Percent of Reused Objects Modified) δηλώνει το ποσοστό των επαναχρησιμοποιημένων αντικειμένων που έχουν τροποποιηθεί [BEN99].
- SRE** (System Reuse) δηλώνει το ποσοστό επαναχρησιμοποίησης των κλάσεων [ABR94a].

2.5.2 Παρουσίαση Μεταμετρικών και Αποτελεσμάτων

Οι αντικειμενοστραφείς μετρικές που αναφέρθηκαν παραπάνω αξιολογήθηκαν και χαρακτηρίστηκαν με τη βοήθεια ενός συνόλου μεταμετρικών. Ο σκοπός αυτής της αξιολόγησης δεν ήταν η κρίση της ορθότητας των μετρικών ή η σύγκριση μεταξύ τους, αλλά η παροχή βοήθειας στην επιλογή ενός κατάλληλου υποσυνόλου από τις παραπάνω μετρικές για κάθε εξειδικευμένη περίπτωση διεξαγωγής μετρήσεων. Ο υπεύθυνος της διεξαγωγής αυτών των μετρήσεων θα πρέπει να κρίνει ποιες μετρικές έχουν νόημα εφαρμογής και θα οδηγήσουν σε κατανοητά και εφαρμόσιμα συμπεράσματα.

Για την παρούσα έρευνα χρησιμοποιήθηκαν 7 διαφορετικές μεταμετρικές, οι οποίες καλύπτουν όλο το φάσμα της διαδικασίας μέτρησης. Περιγράφονται σύντομα παρακάτω, ενώ οι συντομεύσεις τους στις παρενθέσεις θα διευκολύνουν την παρουσίαση των αποτελεσμάτων που γίνεται στη συνέχεια.

Κλίμακα μέτρησης (Measurement scale – MS). Οι τιμές που αναθέτονται στις μετρικές μπορεί να αντιστοιχούν σε διάφορες κλίμακες. Αυτές οι κλίμακες, σύμφωνα με τον Stevens [STV47] είναι οι: ονομαστική (nominal), τακτική (ordinal), διαστήματος (interval), αναλογίας (ratio) και απόλυτη (absolute). Είναι προφανές ότι οι μετρικές που αντιστοιχούν στις δύο πρώτες κλίμακες δεν μπορούν να χρησιμοποιηθούν τόσο εύκολα στην επιτέλεση διαφόρων πράξεων όσο οι μετρικές που αντιστοιχούν στις τρεις τελευταίες κλίμακες. Στην παρουσίαση των αποτελεσμάτων χρησιμοποιείται το σύμβολο “+” για τις τρεις τελευταίες και το σύμβολο “-” για τις δύο πρώτες.

Ασφάλεια στην ανάθεση των τιμών της μετρικής (Measurements independence – MI). Με τον όρο αυτό υπονοείται η ικανότητα της μετρικής να δίνει για τον ίδιο κώδικα το ίδιο αποτέλεσμα, χωρίς να χρειάζεται να επέμβει ο χρήστης για να ορίσει τη μετρική. Μετρικές στις οποίες μπορεί να δοθεί διαφορετική ερμηνεία

από διαφορετικούς χρήστες δεν είναι εύχρηστες σε μελέτες ιστορικών δεδομένων. Για παράδειγμα η μετρική των γραμμών κώδικα έχει κακή συμπεριφορά στο σημείο αυτό, αφού προτού εφαρμοστεί, θα πρέπει να οριστεί επακριβώς η έννοια της γραμμής κώδικα, κάτι το οποίο μπορεί να γίνει διαφορετικά από διαφορετικούς χρήστες. Αντίθετα, η μετρική του αριθμού των μεμονωμένων κλάσεων έχει καλύτερη συμπεριφορά, αφού υπολογίζεται πάντα με τον ίδιο τρόπο, χωρίς να χρειάζεται ορισμούς. Στην παρουσίαση των αποτελεσμάτων χρησιμοποιείται το σύμβολο “+” για τις μετρικές που υπολογίζονται πάντα με τον ίδιο τρόπο, διαφορετικά χρησιμοποιείται το σύμβολο “-”.

Αυτοματοποιησιμότητα (Automation – AU). Εξετάζει την ευκολία υλοποίησης της μετρικής προκειμένου να εφαρμοστεί στον κώδικα ενός προγράμματος, χωρίς όμως το πρόγραμμα να τρέχει. Η προσπάθεια υλοποίησης ποικίλει για τις διάφορες μετρικές, ενώ υπάρχουν και μετρικές που, λόγω της φύσης τους, είναι μη αυτοματοποιήσιμες. Στην παρουσίαση των αποτελεσμάτων χρησιμοποιείται το σύμβολο “+” για τις μετρικές που αυτοματοποιούνται εύκολα, το σύμβολο “=” για αυτές που χρειάζεται αρκετή προσπάθεια για να αυτοματοποιηθούν και το σύμβολο “-” για αυτές που δεν αυτοματοποιούνται.

Σημασία της υλοποίησης (Value of implementation – VI). Εξετάζει αν το αποτέλεσμα μίας μετρικής εξαρτάται από την υλοποίηση του προγράμματος. Αν είναι ανεξάρτητη, τότε μπορεί να εφαρμοστεί στα αρχικά στάδια της ανάπτυξης λογισμικού, ανεξαρτήτως της γλώσσας και του στυλ προγραμματισμού που θα επιλεγεί. Στην αντίθετη περίπτωση όμως, αν εξαρτάται από την υλοποίηση, τότε η μετρική μπορεί να παρέχει μία καλή ένδειξη για το πόσο καλή είναι η υλοποίηση αυτή και αν χρειάζεται διορθωτικές ενέργειες και βελτιώσεις. Στην παρουσίαση των αποτελεσμάτων χρησιμοποιείται το σύμβολο “+” για τις μετρικές που εξαρτώνται από την υλοποίηση και το σύμβολο “-” σε αντίθετη περίπτωση.

Μονοτονία (Monotonicity – MO). Ελέγχει αν δοθέντων δύο προγραμμάτων και ενός τρίτου προγράμματος που τα περιέχει, η τιμή της μετρικής για το τρίτο πρόγραμμα είναι μεγαλύτερη ή ίση από το άθροισμα των τιμών των μετρικών των άλλων δύο. Στην παρουσίαση των αποτελεσμάτων χρησιμοποιείται το σύμβολο “+” στις περιπτώσεις ύπαρξης μονοτονίας και το σύμβολο “-” σε αντίθετη περίπτωση.

Απλότητα (Simplicity – SI). Εξετάζει πόσο απλά και κατανοητά είναι ορισμένη μία μετρική και έχει να κάνει με την αποδοχή της από τους χρήστες. Ένας μέσος χρήστης, που μπορεί και να μην είναι ειδικός της επιστήμης των υπολογιστών, αισθάνεται πιο ασφαλής και σίγουρος όταν καταλαβαίνει τι κάνει αυτό που χρησιμοποιεί. Στην παρουσίαση των αποτελεσμάτων χρησιμοποιείται το σύμβολο “+” για τις πολύ καλά ορισμένες μετρικές, το σύμβολο “=” για τις μέτρια ορισμένες και το σύμβολο “-” για τις δυσνόητες μετρικές.

Ακρίβεια (Accuracy – AC). Εξετάζει αν η μετρική μετράει τελικά αυτό στο οποίο αναφέρεται και πώς σχετίζεται με τα χαρακτηριστικά ή τους παράγοντες του λογισμικού που επιθυμούμε να μετρήσουμε. Στην παρουσίαση των αποτελεσμάτων χρησιμοποιείται το σύμβολο “+” για αυτήν την περίπτωση, ενώ διαφορετικά χρησιμοποιείται το σύμβολο “-”.

Τα αποτελέσματα των αντικειμενοστραφών μετρικών που αναφέρθηκαν στην ενότητα 2.5.1 σύμφωνα με τις επτά μεταμετρικές που περιγράφηκαν παραπάνω παρουσιάζονται στους πέντε πίνακες που ακολουθούν. Οι γραμμές των πινάκων αναπαριστούν τις μετρικές, όπου αναφέρονται με τη συντόμευση που τους δόθηκε,

ενώ οι στήλες αναπαριστούν τις μεταμετρικές, όπου αναφέρονται σύμφωνα με τη δική τους συντόμευση.

Metric	MS	MI	AU	VI	MO	SI	AC
AHF	+	+	+	+	-	+	+
CCO	+	-	+	+	+	=	+
CEC	+	-	-	+	-	-	-
CLM	+	+	+	-	+	+	+
DAM	+	+	+	+	-	+	+
FOC	+	-	+	+	+	+	+
INP	+	+	=	+	+	+	+
LCM	+	+	=	+	+	=	+
MAA	+	+	+	+	-	+	+
MFA	+	+	+	+	-	+	+
MHF	+	+	+	+	-	+	+
NAD	+	-	+	+	+	+	+
NCM	+	+	+	+	+	+	+

Πίνακας 2.1: Μετρικές Κλάσης

Metric	MS	MI	AU	VI	MO	SI	AC
NIV	+	+	+	+	+	+	+
NOA	+	+	+	+	+	+	+
NPA	+	+	+	+	+	+	+
NPM	+	+	+	+	+	+	+
NRA	+	+	+	+	+	=	-
PCM	+	+	+	-	+	+	+
PDA	+	+	-	+	-	+	+
PMR	+	+	-	+	-	=	+
PPD	+	+	+	+	-	+	+
RFC	+	+	+	+	+	+	+
WCS	+	+	+	+	+	=	-
WMC	+	-	+	+	+	+	+

Metric	MS	MI	AU	VI	MO	SI	AC
AMC	+	+	=	+	+	=	+
AMS	+	-	+	+	+	+	+
MAG	+	+	=	+	+	=	+
MCX	+	+	+	+	+	=	+

Πίνακας 2.2: Μετρικές Μεθόδου

Metric	MS	MI	AU	VI	MO	SI	AC
CBO	+	+	+	+	+	+	+
CCP	+	+	+	+	+	+	+
CFA	+	-	-	+	-	=	-

Πίνακας 2.3: Μετρικές Σύνδεσης

Metric	MS	MI	AU	VI	MO	SI	AC
AIF	+	+	+	+	-	+	+
DIT	+	+	=	+	-	+	+
FEF	+	-	+	+	-	+	+
FIN	+	-	+	+	+	+	+
HNL	+	+	=	+	-	+	+
MIF	+	+	+	+	-	+	+
MRE	+	+	+	+	-	+	-
NMI	+	+	+	+	+	+	+

Πίνακας 2.4: Μετρικές Κληρονομικότητας

Metric	MS	MI	AU	VI	MO	SI	AC
NMO	+	+	+	+	+	+	+
NOC	+	+	=	+	+	+	+
PFA	+	-	-	+	-	-	-
PMO	+	-	-	+	-	-	-
RDB	+	+	=	+	-	+	+
RER	+	+	+	+	-	+	-
SIX	-	+	=	+	+	+	+
SPR	+	+	+	+	-	+	-

Metric	MS	MI	AU	VI	MO	SI	AC
ADI	+	+	=	+	-	+	+
ANA	+	-	=	+	-	+	+
APG	+	-	-	+	-	+	+
ASC	+	+	+	+	+	=	+
CAN	-	-	-	-	-	+	-
CRE	+	+	-	+	+	+	+
FDE	+	-	+	+	-	-	-

Πίνακας 2.5: Μετρικές Συστήματος

Metric	MS	MI	AU	VI	MO	SI	AC
NCT	+	+	=	+	+	+	+
NOH	+	-	=	+	+	+	+
OLE	+	+	-	+	-	-	-
PRC	+	+	+	+	+	+	+
PRO	+	+	-	+	-	=	+
SRE	+	+	-	+	-	+	+

Εξωτερικές Μετρικές Λογισμικού

Το κεφάλαιο αυτό αναφέρεται στις εξωτερικές μετρικές λογισμικού, που ως γνωστόν σχετίζονται άμεσα με τα εξωτερικά ποιοτικά χαρακτηριστικά του λογισμικού. Παρουσιάζονται τα πλεονεκτήματα και τα μειονεκτήματα των εξωτερικών μετρικών. Γίνεται ένας διαχωρισμός των πελατών του λογισμικού, ανάλογα με την εμπειρία τους, αλλά και τη φάση του κύκλου ζωής λογισμικού. Περιγράφονται οι διαφορετικές τεχνικές μέτρησης των εξωτερικών ποιοτικών χαρακτηριστικών, δίνοντας μεγαλύτερη έμφαση στην έρευνα με τη χρήση ερωτηματολογίου, η οποία χρησιμοποιήθηκε κατά κόρον στην παρούσα διδακτορική διατριβή. Παρουσιάζεται μία μέθοδος επεξεργασίας και ανάλυσης των αποτελεσμάτων των εξωτερικών μετρήσεων. Τέλος, παρουσιάζεται το εργαλείο CESM, το οποίο έχει ως σκοπό την ταχύτερη συλλογή των αποτελεσμάτων των εξωτερικών μετρήσεων και την αυτοματοποίηση της ανάλυσής τους.

3.1 Πλεονεκτήματα και Μειονεκτήματα

Όπως έχει ήδη αναφερθεί, οι εξωτερικές μετρικές χρησιμοποιούνται για τη μέτρηση χαρακτηριστικών του λογισμικού με υψηλό επίπεδο αφαίρεσης τα οποία σχετίζονται άμεσα με την ποιότητα του λογισμικού. Δίνουν δεδομένα άμεσα αξιοποιήσιμα και ερμηνεύσιμα, αφού μετρούν απευθείας τα ζητούμενα εξωτερικά ποιοτικά χαρακτηριστικά του λογισμικού. Εξυπηρετούν δηλαδή με άμεσο τρόπο τη δυνατότητα μέτρησης της ικανοποίησης των απαιτήσεων του πελάτη, η οποία αποτελεί βασικό στοιχείο κάθε ορισμού της ποιότητας. Επομένως έχουν μεγάλη αξία για κάθε πρόγραμμα εξασφάλισης ποιότητας, αφού στα διεθνή πρότυπα δίνεται μεγάλη έμφαση στη γνώμη του τελικού χρήστη για την ποιότητα του λογισμικού. Επίσης, δίνεται η δυνατότητα για άμεση ανάλυση των ποιοτικών δεδομένων, η οποία αρμόζει σε επιχειρήσεις με υψηλό επίπεδο ωριμότητας ως προς την εξασφάλιση ποιότητας. Τέλος, η μέτρηση εξωτερικών ποιοτικών χαρακτηριστικών συμβαδίζει με το βασικό στόχο της ποιότητας και συμβάλει στην εικόνα της επιχείρησης όπως αυτή εξάγεται στους πελάτες της [XEN96a]. Ταυτόχρονα, αναβαθμίζεται και ο ρόλος του ίδιου του πελάτη, καθώς όχι μόνο γίνεται φανερό ότι τίθεται σε μεγάλη υπόληψη η γνώμη του, αλλά πλέον συμμετέχει και ο ίδιος ενεργά στη διαδικασία παραγωγής λογισμικού.

Ένα από τα βασικότερα μειονεκτήματα των εξωτερικών μετρήσεων είναι ότι βασίζονται σε υποκειμενικά δεδομένα και επομένως τα αποτελέσματά τους είναι πιθανόν να υπόκεινται σε αμφισβητήσεις. Η συχνότητα λαθών είναι σαφώς μεγαλύτερη από αυτή των εσωτερικών μετρήσεων, γεγονός που δυσχεραίνει την ανάλυση των αποτελεσμάτων. Ένα δεύτερο μειονέκτημα είναι η αδυναμία αυτοματοποίησης της διαδικασίας των εξωτερικών μετρήσεων. Η διεξαγωγή τέτοιου είδους μετρήσεων συνεπάγεται και ένα αρκετά αυξημένο κόστος για την επιχείρηση, αφού απαιτείται έντονη συμμετοχή ανθρώπινου δυναμικού. Οι πλέον διαδεδομένοι τρόποι μέτρησης είναι η χρήση ερωτηματολογίων και οι συνεντεύξεις. Για να αποφευχθούν πιθανά λάθη κατά τη μέτρηση, αλλά και για να περιοριστεί το πρόβλημα των υποκειμενικών απαντήσεων, θα πρέπει το δείγμα χρηστών που θα χρησιμοποιηθεί να είναι αρκετά μεγάλο, γεγονός που αυξάνει κατά πολύ το συνολικό κόστος της έρευνας. Τα παραπάνω οδηγούν μερικές φορές στο να μην τίθενται ως άμεσοι στόχοι μέτρησης τα εξωτερικά ποιοτικά χαρακτηριστικά, αλλά να στοχεύεται η έμμεση μέτρησή τους βασισμένη σε εσωτερικές μετρικές.

3.2 Ποιος είναι ο Πελάτης

Συνήθως, όταν αναφερόμαστε στον όρο πελάτης εννοούμε τον τελικό χρήστη που θα αγοράσει ένα προϊόν λογισμικού από μία εταιρεία παραγωγής λογισμικού. Όμως, καθόλη τη διάρκεια παραγωγής λογισμικού μπορούμε να συναντήσουμε διάφορα είδη πελατών, μιας που αυτή απαρτίζεται από διακριτές φάσεις. Συγκεκριμένα, όπως είναι γνωστό, στην τεχνολογία λογισμικού υπάρχουν αρκετά μοντέλα που περιγράφουν τη διαδικασία παραγωγής λογισμικού, με πιο γνωστό το μοντέλο καταρράκτη [ROY70]. Στον κύκλο ζωής λογισμικού διακρίνουμε τις εξής φάσεις: Ανάλυση Απαιτήσεων, Σχεδίαση, Υλοποίηση, Έλεγχος και Συντήρηση [PRE97]. Τα δεδομένα εξόδου κάθε φάσης αποτελούν δεδομένα εισόδου για την επόμενη φάση. Σε μεγάλες εταιρείες παραγωγής λογισμικού οι φάσεις αυτές ανατίθενται συνήθως σε διαφορετικές ομάδες εργασίας ή και σε διαφορετικά τμήματα της εταιρείας. Με αυτή τη λογική λοιπόν οι προγραμματιστές που θα αναλάβουν την υλοποίηση μπορούν να θεωρηθούν πελάτες των αναλυτών. Παρόμοια, το τμήμα ελέγχου και το τμήμα συντήρησης μπορεί να θεωρηθεί ως πελάτης του τμήματος υλοποίησης. Αλλά και η ίδια η εταιρεία μπορεί να θεωρηθεί πελάτης του εαυτού της όταν επαναχρησιμοποιεί κώδικά της ή τμήματα λογισμικού. Κατά συνέπεια, η έννοια πελάτης σε μία εταιρεία είναι αρκετά ευρεία.

Οι ποιοτικές απαιτήσεις ποικίλουν ανάλογα με το είδος του πελάτη. Έτσι, ο πελάτης που αγοράζει το προϊόν λογισμικού θέλει για παράδειγμα να είναι αξιόπιστο και φιλικό στη χρήση, το τμήμα ελέγχου θέλει από το λογισμικό να είναι εύκολο να ελεγχθεί, ενώ το τμήμα συντήρησης θέλει από το λογισμικό να είναι εύκολα συντηρήσιμο. Επομένως, κατά τη διαδικασία μέτρησης της γνώμης του πελάτη για την ποιότητα του λογισμικού θα πρέπει πάντοτε να λαμβάνονται υπόψη οι ιδιαιτερότητες και η βαρύτητα που δίνει το κάθε είδος πελάτη στα διάφορα συστατικά της ποιότητας λογισμικού.

Ένας άλλος διαχωρισμός των πελατών γίνεται με βάση τα προσόντα που διαθέτουν, δηλαδή σχετίζεται με το πόσο καλός αξιολογητής του λογισμικού είναι ο κάθε πελάτης και κατά πόσο η γνώμη του έχει ιδιαίτερη βαρύτητα, λόγω των ειδικών γνώσεων και της εμπειρίας που διαθέτει [XEN96a]. Τα προσόντα αυτά, που είναι ένα ζητούμενο μέτρησης κατά τη διαδικασία μίας διεξαγωγής εξωτερικών μετρήσεων και βοηθούν στον καθορισμό του βαθμού εμπειρίας του κάθε πελάτη, χωρίζονται στις τρεις παρακάτω κατηγορίες:

- προσωπικό υπόβαθρο (personal background)
- συντακτική γνώση (syntactic knowledge)
- σημασιολογική γνώση (semantic knowledge)

Ως **προσωπικό υπόβαθρο** ορίζουμε όλα τα στοιχεία του πελάτη τα οποία δεν σχετίζονται ειδικά με τη συγκεκριμένη εφαρμογή λογισμικού, αλλά σχετίζονται άμεσα με την ικανότητα του πελάτη να αξιολογεί τα εξωτερικά ποιοτικά χαρακτηριστικά του προϊόντος τα οποία επιθυμούμε να μετρήσουμε. Τέτοια στοιχεία για παράδειγμα αποτελούν η γενικότερη παιδεία του πελάτη, η παιδεία του σε σχέση με ηλεκτρονικούς υπολογιστές, τα χρόνια εκπαίδευσης σε σχέση με ηλεκτρονικούς υπολογιστές, το επίπεδο των σπουδών του κ.ά.

Ως **συντακτική γνώση** ορίζουμε τη γνώση των εφαρμογών λογισμικού αντίστοιχων με το συγκεκριμένο λογισμικό προς αξιολόγηση και γενικότερα την εμπειρία του πελάτη στη χρήση προγραμμάτων ηλεκτρονικών υπολογιστών. Η εμπειρία αυτή σχετίζεται τόσο με τη γενική χρήση προγραμμάτων σε ηλεκτρονικό υπολογιστή, όσο και με τη χρήση προγραμμάτων της ίδιας κατηγορίας με το συγκεκριμένο λογισμικό για το οποίο διεξάγεται η εξωτερική μέτρηση. Η γνώση και η εμπειρία δεν σχετίζονται με τη γνώση της διαδικασίας η οποία αυτοματοποιείται από το λογισμικό, αλλά μόνο με την ευκολία στη χρήση του λογισμικού.

Ως **σημασιολογική γνώση** ορίζουμε τη γνώση της συγκεκριμένης διαδικασίας η οποία αυτοματοποιείται με τη χρήση του ηλεκτρονικού υπολογιστή. Στην πλειοψηφία τους οι εφαρμογές λογισμικού αυτοματοποιούν μία διαδικασία η οποία προϋπήρχε και εκτελούνταν χωρίς τη βοήθεια του ηλεκτρονικού υπολογιστή. Η γνώση αυτής της διαδικασίας, τα χρόνια εμπειρίας και ο βαθμός εξειδίκευσης συνθέτουν τη σημασιολογική γνώση ενός πελάτη.

Με βάση τα παραπάνω οι τελικοί χρήστες ενός προγράμματος λογισμικού μπορούν να διαχωριστούν στις εξής κατηγορίες:

- Αρχάριος ή χρήστης χαμηλής εμπειρίας. Θεωρείται ο χρήστης που έχει μέτρια ή μικρή συντακτική και σημασιολογική γνώση.
- Μέτριος ή χρήστης μέτριας εμπειρίας. Θεωρείται α) ο χρήστης που έχει χαμηλή συντακτική αλλά υψηλή σημασιολογική γνώση και β) ο χρήστης που έχει υψηλή συντακτική αλλά χαμηλή σημασιολογική γνώση. Για παράδειγμα, για μία ιατρική εφαρμογή λογισμικού, ένας γιατρός με ελάχιστες γνώσεις περί ηλεκτρονικών υπολογιστών ανήκει στην (α) περίπτωση, ενώ ένας χειριστής υπολογιστών με χρόνια εμπειρία στη χρήση ανάλογων εφαρμογών λογισμικού, αλλά με ελάχιστες γνώσεις περί ιατρικής ανήκει στη (β) περίπτωση.
- Έμπειρος ή χρήστης υψηλής εμπειρίας. Θεωρείται ο χρήστης που έχει μεγάλη και συντακτική και σημασιολογική γνώση.

Κατά τη διαδικασία της μέτρησης της γνώμης των χρηστών για την ποιότητα ενός προϊόντος λογισμικού είναι συνήθως επιθυμητό από αυτούς που διεξάγουν τη μέτρηση η συμμετοχή χρηστών όσο το δυνατόν μεγαλύτερης εμπειρίας. Εξάλλου, θα μπορούσαμε να περιοριστούμε μόνο στους έμπειρους χρήστες, που η γνώμη τους έχει σαφώς μεγαλύτερη βαρύτητα από αυτή των άπειρων, θεωρώντας ότι ακόμα και οι άπειροι χρήστες, ύστερα από κάποιον εύλογο χρόνο χρήσης της εφαρμογής λογισμικού, θα γίνουν και αυτοί έμπειροι. Δυστυχώς, η εύρεση ενός ικανοποιητικού αριθμού από έμπειρους χρήστες, έτσι ώστε το δείγμα των χρηστών που συμμετέχει

στη μέτρηση να είναι αξιόλογο, είναι αρκετά δύσκολη. Παράλληλα, σε αρκετές περιπτώσεις εφαρμογών λογισμικού που εστιάζουν κυρίως σε άπειρους χρήστες, όπως είναι οι εφαρμογές εκπαιδευτικού λογισμικού, η συμμετοχή άπειρων χρηστών στη διαδικασία της μέτρησης μπορεί να αναδείξει ορισμένες ατέλειες ή ασάφειες της εφαρμογής λογισμικού, τις οποίες ένας έμπειρος χρήστης συνήθως τις αγνοεί γιατί τις θεωρεί ανούσιες. Επομένως, το δείγμα των χρηστών που συμμετέχει στη διαδικασία των εξωτερικών μετρήσεων για τον καθορισμό της ποιότητας μίας εφαρμογής λογισμικού να μην είναι επιθυμητό να απαρτίζεται από χρήστες με όσο το δυνατόν μεγαλύτερη εμπειρία, αλλά σε ορισμένες περιπτώσεις είναι προτιμότερο να συμμετέχουν εξίσου και χρήστες με χαμηλή εμπειρία.

3.3 Τεχνικές Μέτρησης

Διάφορες τεχνικές έχουν αναπτυχθεί κατά καιρούς για τη μέτρηση της γνώμης των χρηστών για την ποιότητα ενός προϊόντος λογισμικού. Ένας πρώτος διαχωρισμός τους [ABO00] μπορεί να γίνει σε **αναλυτικές** και **εμπειρικές τεχνικές**. Οι αναλυτικές στηρίζονται σε θεωρητικά μοντέλα που προσομοιώνουν τη συμπεριφορά του χρήστη ή σε πρότυπα και κανόνες. Συνήθως χρησιμοποιούνται κατά τη φάση σύνταξης προδιαγραφών, συχνά ακόμη και πριν από την κατασκευή πρωτοτύπων και δεν απαιτείται η συμμετοχή των τελικών χρηστών. Αντίθετα, οι εμπειρικές στηρίζονται στην κατασκευή και αξιολόγηση της συμπεριφοράς ή των χαρακτηριστικών ενός πρωτοτύπου ή ενός ολοκληρωμένου συστήματος. Στην αξιολόγηση αυτή μπορεί να συμμετάσχουν αντιπροσωπευτικοί χρήστες καθώς και έμπειροι αξιολογητές της ποιότητας ενός προϊόντος λογισμικού. Οι εμπειρικές τεχνικές μπορούν να χωριστούν σε **πειραματικές** και σε **διερευνητικές**.

Στη συνέχεια παρουσιάζονται χαρακτηριστικά παραδείγματα για την κάθε ομάδα τεχνικών μέτρησης. Μερικές από αυτές τις τεχνικές έχουν προταθεί κυρίως για τη μέτρηση της ευχρηστίας λογισμικού. Παρόλα αυτά, μπορούν επίσης να χρησιμοποιηθούν για τη μέτρηση της γνώμης των χρηστών και για τους υπόλοιπους παράγοντες της ποιότητας λογισμικού.

3.3.1 Αναλυτικές Τεχνικές

Χαρακτηριστικά παραδείγματα αναλυτικών τεχνικών αναφέρονται παρακάτω:

- **Ανάλυση πληκτρολογήσεων.** Χρησιμοποιείται κατά τη φάση σύνταξης προδιαγραφών και αρχικού σχεδιασμού για την εκτίμηση της απόδοσης ενός συστήματος και στηρίζεται στο μοντέλο πληκτρολόγησης (KLM) [CAR83].
- **Γνωστικό περιδιάβασμα.** Εξετάζει την ευχρηστία του συστήματος όσον αφορά την ικανότητά του να βοηθήσει πρωτόπειρους χρήστες κατά τη φάση διερευνητικής εκμάθησης της λειτουργίας του.
- **Μέθοδοι ευρετικής αξιολόγησης.** Εστιάζουν στην αξιολόγηση εφαρμογών στις οποίες οι ενέργειες του χρήστη δεν είναι απόλυτα προβλέψιμες και ο χρήστης δεν είναι πρωτόπειρος.
- **Έλεγχος συμβατότητας με κανόνες σχεδιασμού και πρότυπα.** Εξετάζει αν ικανοποιείται από το σύστημα ένα σύνολο από λεπτομερείς κανόνες σχεδιασμού, οι οποίοι παίρνουν τη μορφή ενός καταλόγου ελέγχου (checklist).

3.3.2 Πειραματικές Τεχνικές

Οι πειραματικές τεχνικές απαιτούν την ενεργό συμμετοχή στην αξιολόγηση των χρηστών του συστήματος σε συνθήκες εργαστηρίου και οι πιο γνωστές είναι οι εξής:

- **Μέτρηση απόδοσης.** Είναι μία κλασική τεχνική αξιολόγησης λογισμικού που παρέχει ποσοτικές μετρήσεις της απόδοσης του συστήματος όταν οι χρήστες εκτελούν προκαθορισμένες ενέργειες ή και ολόκληρες εργασίες. Οι χρήστες αφήνονται συνήθως με ελάχιστη καθοδήγηση να εκτελέσουν αυτές τις ενέργειες, επομένως η αλληλεπίδραση ανάμεσα σε αυτούς και τους υπεύθυνους της έρευνας είναι μικρή.
- **Πρωτόκολλο ομιλούντων χρηστών.** Σύμφωνα με αυτή την τεχνική, που εστιάζει στη μέτρηση της αποτελεσματικότητας του συστήματος και της ικανοποίησης που παρέχει στο χρήστη, ένας μικρός αριθμός χρηστών (τρεις με τέσσερις) καλείται να εκφράσει μεγαλόφωνα τις σκέψεις, τις απόψεις και τα συναισθήματά του, ενώ αλληλεπιδρά με το σύστημα. Τα παραπάνω καταγράφονται ώστε να αναλυθούν σε συνδυασμό με τις ενέργειες του χρήστη, οι οποίες πρέπει επίσης να καταγραφούν.
- **Καταγραφή ενεργειών των χρηστών.** Οι συνήθως χρησιμοποιούμενοι τρόποι της καταγραφής των ενεργειών των χρηστών είναι οι σημειώσεις του ερευνητή, η ηχογράφηση των χρηστών, η βιντεοσκόπηση των χρηστών, η καταγραφή συμβάντων στον υπολογιστή και η καταγραφή συμβάντων από τους χρήστες.

3.3.3 Διερευνητικές Τεχνικές

Οι διερευνητικές τεχνικές αφορούν τη διερεύνηση των ποιοτικών χαρακτηριστικών του συστήματος μέσω καταγραφής των απόψεων του χρήστη. Η έρευνα διεξάγεται συνήθως στο φυσικό χώρο του χρήστη όπου αξιολογείται είτε το τελικό προϊόν λογισμικού είτε ένα αρκετά προχωρημένο πρωτότυπό του. Αυτές οι τεχνικές απαιτούν τη συμμετοχή ενός αξιόλογου δείγματος χρηστών, μεγάλου συνήθως αριθμού, ενώ οι πιο διαδεδομένες είναι οι εξής:

- **Συμπλήρωση ερωτηματολογίων.** Οι χρήστες καλούνται να απαντήσουν ένα δομημένο ερωτηματολόγιο, όπου οι καταγεγραμμένες ερωτήσεις είναι συνήθως πολλαπλής επιλογής (multiple choice) και μέσω του οποίου οι χρήστες εκφράζουν την άποψή τους για τα ποιοτικά χαρακτηριστικά του προϊόντος λογισμικού.
- **Συνεντεύξεις χρηστών.** Πρόκειται για μία δομημένη τεχνική αξιολόγησης του συστήματος κατά την οποία η επαφή του ατόμου που διεξάγει την έρευνα με το χρήστη είναι άμεση.
- **Ομαδική αξιολόγηση.** Είναι μία παραλλαγή της προηγούμενης τεχνικής κατά την οποία γίνεται μία σύσταση και λειτουργία μίας ομάδας από 5 έως 10 χρήστες υπό την επίβλεψη ενός συντονιστή, ο οποίος έχει την αρμοδιότητα της προετοιμασίας των προς συζήτηση θεμάτων, καθώς και της συναγωγής συμπερασμάτων για την ποιότητα του λογισμικού στο τέλος της συζήτησης.
- **Παρατήρηση πεδίου.** Εμπίπτει στην κατηγορία των λεγόμενων εθνογραφικών μελετών και αφορά την παρατήρηση των χρηστών στο φυσικό χώρο όπου επιτελούν το έργο τους ενώ χρησιμοποιούν το σύστημα.

Στην παρούσα διδακτορική διατριβή ακολουθήθηκαν διερευνητικές τεχνικές μέτρησης της γνώμης των χρηστών για την ποιότητα του λογισμικού και κυρίως χρησιμοποιήθηκαν η συμπλήρωση ερωτηματολογίων από το χρήστη και οι συνεντεύξεις. Στη συνέχεια, οι δύο αυτές τεχνικές αναλύονται περισσότερο, όπου αναφέρονται τα πλεονεκτήματα και μειονεκτήματά τους.

Και οι δύο τεχνικές βασίζονται σε ένα ερωτηματολόγιο σχετικό με τα εξωτερικά ποιοτικά χαρακτηριστικά του λογισμικού, αλλά η βασική διαφορά τους είναι ότι στη μία η συμπλήρωση του ερωτηματολογίου γίνεται απευθείας από το χρήστη, χωρίς καμία επαφή με το ερευνητή, ενώ στην άλλη γίνεται από τον ερευνητή, ο οποίος έρχεται σε άμεση επαφή με το χρήστη. Και στις δύο περιπτώσεις, οι απαντήσεις που θα δώσει ο τελικός χρήστης κατά την έρευνα θα πρέπει να κριθούν σύμφωνα με τα ακόλουθα κριτήρια [JAV96]:

1. Εκείνο της **ικανότητας** του ερωτώμενου. Το αντικείμενο των ερωτήσεων του είναι γνωστό; Πρόκειται για μία γνώση κατανοητή ή μη, επιφανειακή ή βαθιά, παρελθούσα ή πρόσφατη;
2. Εκείνο της **κατανόησης** από την πλευρά του ερωτώμενου. Καταλαβαίνει το περιεχόμενο των ερωτήσεων όλου του ερωτηματολογίου; Κατέχει το χρησιμοποιούμενο λεξιλόγιο; Η προσωπική του κατάσταση τη στιγμή της έρευνας του επιτρέπει να απαντήσει;
3. Εκείνο της **ελικρίνειας** του ερωτώμενου. Απαντάει σύμφωνα με τη συνείδησή του ή ψεύδεται εν γνώσει του ή ακόμα και εν αγνοία του;
4. Εκείνο της **αξιοπιστίας** από πλευράς του ερωτώμενου. Εξωτερικεύει με λόγια κατάλληλα τα αληθινά του συναισθήματα; Η μνήμη του δεν τον εγκαταλείπει;

Στα παραπάνω κριτήρια μπορούν να προστεθούν και οι διάφορες πολιτισμικές παραδοχές που αφορούν κυρίως τον ιδιαίτερο τρόπο συμπεριφοράς και τις συνήθειες του χρήστη.

3.3.4 Τα στάδια μίας έρευνας με ερωτηματολόγιο

Η σύνθεση των ερωτηματολογίων με τα οποία διεξάγεται η έρευνα και ο σχεδιασμός των ερωτήσεων βασίζεται σε γνωστές θεωρίες ανάλυσης μη ποσοτικών δεδομένων [PAY51] [KIR86] [SRA90] [MLS94], οι οποίες παρέχουν τις βασικές διαδικασίες για τη σχεδίαση ερωτηματολογίων γενικής φύσεως. Θα πρέπει να γίνει κατανοητό ότι το ερωτηματολόγιο μέτρησης των εξωτερικών ποιοτικών χαρακτηριστικών δεν είναι μία απλή παράθεση των χαρακτηριστικών για τα οποία ζητούμε τη γνώμη των πελατών, αλλά ένα παραδοτέο της επιχείρησης, η παραγωγή του οποίου πρέπει να περνά από φάσεις σχεδίασης και ελέγχου και πρέπει να πληροί κι αυτό με τη σειρά του ποιοτικές προδιαγραφές. Αναλυτικότερα, τα στάδια που πρέπει να ακολουθήσει ο υπεύθυνος της διεξαγωγής έρευνας με ερωτηματολόγιο, τα περισσότερα εκ των οποίων εφαρμόζονται και στην περίπτωση της έρευνας με συνέντευξη, είναι τα παρακάτω [JAV96]:

1. **Προσδιορισμός του αντικειμένου έρευνας.** Προσδιορίζεται το θέμα της έρευνας και οι κατηγορίες των χρηστών που θα ερωτηθούν.
2. **Επιλογή των υλικών μέσων που θα διατεθούν στους πραγματοποιούντες την έρευνα.** Προβλήματα διάρκειας της έρευνας, οικονομικά κονδύλια, διαθέσιμο προσωπικό κ.τ.λ.

3. **Προηγούμενες έρευνες.** Αναζήτηση πηγών που αφορούν το θέμα: άλλες έρευνες, συγκέντρωση ποσοτικών, αριθμητικών στοιχείων κ.τ.λ.
4. **Καθορισμός του αντικειμενικού σκοπού της έρευνας και των υποθέσεων εργασίας.** Τι θέλει κάποιος να δείξει με την έρευνα αυτή; Ποιες μέθοδοι οδηγούν στη διατύπωσή τους: ατομικές συνεντεύξεις ή ομαδικές;
5. **Καθορισμός του πληθυσμού ή του πεδίου έρευνας.** Τα ερωτώμενα άτομα σε ποια κατηγορία πληθυσμού πρέπει να ανήκουν (π.χ. ηλικίας, επαγγέλματος, κοινωνικής τάξης);
6. **Κατασκευή του δείγματος.** Τρόπος δημοσκόπησης που θα χρησιμοποιηθεί, έκταση του δείγματος κ.τ.λ.
7. **Σύνταξη του πλάνου του ερωτηματολογίου.** Επιλογή του τρόπου χειρισμού του ερωτηματολογίου, επιλογή και διατύπωση των ερωτήσεων κ.τ.λ.
8. **Δοκιμή του πλάνου ερωτηματολογίου.** Προ-έρευνα ή πιλοτική έρευνα στο χώρο όπου θα διεξαχθεί η έρευνα, ώστε να δοκιμαστεί το ερωτηματολόγιο και να δοθεί η τελική μορφή του.
9. **Σύνταξη του οριστικού ερωτηματολογίου.** Σελιδοποίηση και αρχική κωδικοποίηση.
10. **Εκπαίδευση των ερευνητών-συνεντευκτών.** Εκπαίδευση των ατόμων που θα μοιράζουν και θα συλλέγουν τα ερωτηματολόγια ή των ατόμων που θα διεξάγουν τη συνέντευξη.
11. **Υλοποίηση της έρευνας.** Επίβλεψη των ερευνητών, συγκέντρωση των συμπληρωμένων ερωτηματολογίων, έλεγχος των ερευνητών καθώς και των ερωτηματολογίων κ.τ.λ.
12. **Κωδικοποίηση των ερωτηματολογίων.** Επεξεργασία του κώδικα, εκπαίδευση των κωδικογράφων κ.τ.λ.
13. **Ανίχνευση των ερωτηματολογίων.** Σχέδιο ανίχνευσης, επιλογή του τρόπου επεξεργασίας κ.τ.λ.
14. **Επαλήθευση του δείγματος και ανάλυση των δεδομένων.** Επιλογή στατιστικών μεθόδων, επιλογή βασικών πινάκων κ.τ.λ.
15. **Τελική σύνταξη.** Σύνταξη των αποτελεσμάτων της έρευνας και παρουσίαση.

Τα παραπάνω στάδια δεν αποτελούν απαραίτητα μία αυστηρή χρονική ακολουθία. Δηλαδή, μερικά στάδια μπορούν να διεξαχθούν συγχρόνως. Για παράδειγμα το 1^ο μαζί με το 2^ο, το 6^ο μαζί με το 7^ο κ.τ.λ.

3.3.5 Πλεονεκτήματα και μειονεκτήματα

Οι τεχνικές μέτρησης της γνώμης του χρήστη για την ποιότητα ενός προϊόντος λογισμικού με τη χρήση προσωπικής συνέντευξης (πρόσωπο με πρόσωπο, τηλεφωνική, κ.ά.) και με τη χρήση ερωτηματολογίου (επιστολή, e-mail, Web-page, κ.ά.) παρουσιάζουν όλα τα πλεονεκτήματα και μειονεκτήματα των εξωτερικών μετρήσεων που αναφέρθηκαν στην ενότητα 3.1. Συγκρίνοντας όμως μεταξύ τους αυτές τις δύο τεχνικές μπορούμε να διακρίνουμε και τα παρακάτω:

Όσον αφορά τη συνέντευξη υπάρχει πάντα η σιγουριά για το ποιος απαντάει, αφού βασίζεται στην προσωπική επαφή ερευνητή και πελάτη. Η άμεση αυτή επικοινωνία μάλιστα διαμορφώνει εύλογα την αντίληψη στον πελάτη ότι συμμετέχει

και ο ίδιος ενεργά στην όλη διαδικασία παραγωγής του προϊόντος λογισμικού. Επίσης, σύμφωνα με αυτή την τεχνική, δίνεται η δυνατότητα στον ερευνητή για περαιτέρω διευκρίνιση ερωτήσεων που πιθανώς να παρουσιάζουν ασάφειες ή να είναι δυσνόητες στο χρήστη. Τέλος, άλλο ένα πλεονέκτημα της συνέντευξης είναι η ευελιξία που παρουσιάζει και η δυνατότητα που έχει να καταγράφει λεπτομερώς τις απόψεις του χρήστη για την ποιότητα ενός προϊόντος λογισμικού.

Στα μειονεκτήματα της συνέντευξης πρέπει να αναφερθεί πρώτο από όλα ότι σε σχέση με την έρευνα με ερωτηματολόγιο αποτελεί μία ιδιαίτερος χρονοβόρα διαδικασία. Αρκεί κάποιος να σκεφτεί τις ανθρωποώρες που απαιτούνται από τα άτομα που παίρνουν τη συνέντευξη για να υπολογίσει τόσο το κόστος όσο και το χρόνο που χρειάζονται από πλευράς της εταιρείας παραγωγής λογισμικού. Επιπλέον, σε αυτή την τεχνική παρουσιάζεται μία δυσκολία τόσο στην ανάλυση και στατιστική επεξεργασία των απαντήσεων όσο και στη συνεπαγωγή γενικών συμπερασμάτων. Μερικές φορές ακόμα παρουσιάζεται μία αδυναμία της έρευνας με τη χρήση της συνέντευξης να καταγράφει ποσοτικά χαρακτηριστικά και στοιχεία που αφορούν την απόδοση του συστήματος [ABO00]. Τέλος, πάντα υπάρχει ένας εύλογος φόβος επιρροής ή καθοδήγησης του χρήστη στις απαντήσεις που δίνει από τον υπεύθυνο της συνέντευξης, λόγω αυτής της άμεσης προσωπικής επαφής. Έτσι, πολλές φορές, έρευνες βασισμένες σε συνεντεύξεις αντικατοπτρίζουν σε μεγάλο βαθμό τη γνώμη των ατόμων τα οποία παίρνουν τη συνέντευξη και παρατηρείται το φαινόμενο να υπάρχουν παρόμοιες απόψεις ανά ομάδα χρηστών με το άτομο που πήρε τις συνεντεύξεις [THO94].

Όσον αφορά τη χρήση ερωτηματολογίου, μπορούμε να διακρίνουμε ότι σε σχέση με τη συνέντευξη αποτελεί μία διαδικασία που είναι σαφώς λιγότερο χρονοβόρα και απαιτεί μικρότερο κόστος από την εταιρεία παραγωγής λογισμικού. Είναι μία διαδικασία που μπορεί εν μέρει να αυτοματοποιηθεί με τη χρήση ηλεκτρονικού ταχυδρομείου και ιστοσελίδων. Μία πρόταση για αυτήν την αυτοματοποίηση δίνεται στην ενότητα 3.5, όπου παρουσιάζεται το εργαλείο CESM. Το ερωτηματολόγιο στοχεύει σε αρκετά μεγαλύτερο δείγμα χρηστών και τα αποτελέσματα που παρουσιάζει είναι άμεσα αναλύσιμα και στατιστικώς επεξεργάσιμα. Επομένως, με αυτή την τεχνική η συναγωγή συμπερασμάτων είναι αρκετά εύκολη. Τέλος, οι χρήστες δίνουν απαντήσεις χωρίς επιρροές από τον υπεύθυνο της έρευνας.

Στα μειονεκτήματα του ερωτηματολογίου μπορούμε να αναφέρουμε ότι είναι λιγότερο ευέλικτο σε σχέση με τη συνέντευξη, αφού οι ερωτήσεις είναι καταγεγραμμένες και δεν μπορούν να αλλάξουν κατά περίπτωση. Έτσι, υπάρχει δυστυχώς πάντα μία πιθανότητα ασάφειας στις ερωτήσεις, που μπορεί να οδηγήσει το χρήστη σε απάντηση διαφορετική από αυτή που θα ήθελε. Η τεχνική αυτή παρουσιάζει ένα μικρό αριθμό επιστροφής απαντήσεων, αφού συνήθως μόνο ένα 10 με 20% των χρηστών ανταποκρίνεται στην έρευνα. Άλλα μειονεκτήματα του ερωτηματολογίου είναι η αβεβαιότητα για το ποιος απάντησε, καθώς και η αβεβαιότητα για την προσοχή, τη σιγουριά και το ενδιαφέρον του χρήστη κατά τη διάρκεια της συμπλήρωσης του ερωτηματολογίου. Πιθανές αιτίες για λάθη που παρουσιάζονται στα αποτελέσματα των μετρήσεων αυτής της τεχνικής είναι οι παρακάτω [XEN95]:

- Ο πελάτης δεν απάντησε ο ίδιος το ερωτηματολόγιο, αλλά το έδωσε σε κάποιον άλλο τελείως αναρμόδιο να απαντήσει.

- Ο πελάτης απάντησε στο ερωτηματολόγιο βιαστικά και απρόσεκτα και επέλεξε τυχαία σε περιπτώσεις που δεν κατάλαβε το ερώτημα ή που δεν διάβασε τις οδηγίες.
- Ο πελάτης άρχισε να απαντά στο ερωτηματολόγιο με ενδιαφέρον, αλλά στη συνέχεια κουράστηκε, έχασε το ενδιαφέρον του και έκανε τυχαίες επιλογές για να το ολοκληρώσει.
- Ο πελάτης απάντησε στο ερωτηματολόγιο με ενδιαφέρον και προσοχή σε όλη τη διάρκειά του, αλλά παρόλα αυτά παρεξήγησε μερικές ερωτήσεις και χωρίς πρόθεση έκανε επιλογές που δεν συμβαδίζουν με την πραγματική του άποψη.

Στις παραπάνω αιτίες, με εξαίρεση την πρώτη, ο αριθμός των λαθών θα μπορούσε να μειωθεί σημαντικά με έναν καλό σχεδιασμό του ερωτηματολογίου.

Πέρα από τα παραπάνω, ο Mucchielli [MUC68] διατύπωσε μία σειρά από κατηγορίες από ακούσιες διαστρεβλώσεις, που προέρχονται από τον τρόπο απάντησης των ερωτώμενων ατόμων και δεν ανήκουν στις προαναφερθείσες αιτίες για λάθη. Αναφορικά, οι επτά διαφορετικές κατηγορίες είναι οι εξής:

1. Η αντίδραση για λόγους γοήτρου.
2. Η αυτοάμυνα σε προσωποποιημένες ερωτήσεις
3. Οι υπαγορευμένες από τη διατύπωση της ερώτησης απαντήσεις
4. Η έλξη της θετικής απάντησης
5. Ο φόβος ορισμένων λέξεων
6. Η έλξη από τις αναφορές σε προσωπικότητες
7. Ο φόβος μίας αλλαγής

Θα πρέπει να σημειωθεί ότι οι παραπάνω κατηγορίες που μπορούν να οδηγήσουν σε ακούσια διαστρέβλωση στις απαντήσεις που δίνονται από τους ερωτώμενους απαντώνται τόσο στην έρευνα με τη χρήση συνέντευξης όσο και στην έρευνα με τη χρήση ερωτηματολογίου.

3.3.6 Τύποι και βάρη ερωτήσεων

Ένας βασικός παράγοντας στη σύνταξη ενός σωστού και επιτυχημένου ερωτηματολογίου για την έρευνα και καταγραφή της άποψης του πελάτη για την ποιότητα λογισμικού είναι η σαφής και κατανοητή δόμηση των ερωτήσεων που το αποτελούν. Όπως προαναφέρθηκε στην ενότητα 1.4, η έννοια της ποιότητας λογισμικού είναι αρκετά αφηρημένη ως οντότητα για να κατανοηθεί. Για το λόγο αυτό έχουν αναπτυχθεί διάφορα μοντέλα, στα οποία συντελείται ένας επιμερισμός της ποιότητας σε συνιστώσες και χαρακτηριστικά. Είδαμε λοιπόν στο μοντέλο του McCall ότι η έννοια της ποιότητας χωρίζεται σε 11 παράγοντες και στη συνέχεια αυτοί χωρίζονται σε 25 διαφορετικά κριτήρια ποιότητας. Κατ' αναλογία, στο πρότυπο ISO9126 η ποιότητα χωρίζεται σε έξι ποιοτικά χαρακτηριστικά, τα οποία στη συνέχεια διασπώνται σε περαιτέρω υπο-χαρακτηριστικά.

Με αντίστοιχο τρόπο θα πρέπει να γίνεται και ο διαχωρισμός και η δόμηση των ερωτήσεων κατά την έρευνα με ερωτηματολόγιο. Συγκεκριμένα, θα πρέπει ο ερευνητής να καθορίσει τις ερωτήσεις που αντιστοιχούν σε κάθε μία συνιστώσα της ποιότητας λογισμικού και στη συνέχεια να προχωρήσει σε μία λογική ομαδοποίησή τους, έτσι ώστε να αποσαφηνίζεται και να διευκολύνεται η ροή και η απάντηση του ερωτηματολογίου από τους χρήστες. Για παράδειγμα, θα ήταν όχι μόνο κουραστικό,

αλλά και παράλογο να ρωτάμε αρχικά το χρήστη σχετικά με την ακρίβεια του λογισμικού (χαρακτηριστικό που ανήκει στη Λειτουργικότητα), στη συνέχεια να τον ρωτάμε σχετικά με την ευκολία μάθησης (χαρακτηριστικό που ανήκει στην Ευχρηστία) και τέλος να τον ρωτάμε για την ασφάλεια του λογισμικού (χαρακτηριστικό που ανήκει πάλι στη Λειτουργικότητα). Η σωστή δόμηση εξάλλου θα διευκολύνει στη συνέχεια και τη στατιστική επεξεργασία των απαντήσεων.

Κατά τη σύνταξη του ερωτηματολογίου οι ερωτήσεις που θα το αποτελούν μπορούν να είναι ενός από τους παρακάτω τύπους:

1. **Κλειστές Ερωτήσεις.** Σε αυτές τις ερωτήσεις οι απαντήσεις καθορίζονται από πριν και ο ερωτώμενος οφείλει υποχρεωτικά να επιλέξει μεταξύ όλων αυτών που προτείνονται. Αυτός ο τύπος προσφέρεται καλύτερα για στατιστική ανάλυση και ανίχνευση, αφού δεν αφήνει περιθώρια για διφορούμενες απαντήσεις. Απαντώνται πολύ γρήγορα και εγγυώνται μία σχετική ανωνυμία., αλλά παρουσιάζουν μερικές φορές τον κίνδυνο να υπαγορεύσουν την απάντηση στην έρευνα. Επίσης, μπορούν να χρησιμοποιηθούν και σαν **ερωτήσεις φίλτρα**, δηλαδή μέσο διάκρισης και κατηγοριοποίησης των απαντώντων μεταξύ ποικίλων μεταγενέστερων απαντήσεων. Στις ερωτήσεις φίλτρα ο ερωτώμενος απαντάει συνήθως με ΝΑΙ ή ΟΧΙ και στη συνέχεια, ανάλογα με την απάντησή του, καλείται να απαντήσει σε διαφορετικές ερωτήσεις που ακολουθούν.
2. **Ανοιχτές Ερωτήσεις.** Σε αυτές τις ερωτήσεις η απάντηση δεν προβλέπεται ούτε προκαθορίζεται. Αντίθετα, ο ερωτώμενος είναι ελεύθερος να εκφραστεί όπως θέλει. Αν οι ερωτήσεις έχουν διατυπωθεί με σωστό τρόπο, μπορούν να συγκεντρωθούν ικανοποιητικές πληροφορίες για κάθε θέμα, ιδιαίτερα σε λεπτεπίλεπτα ζητήματα. Η χρήση ανοιχτών απαντήσεων επιβάλλεται όταν δεν είναι δυνατή η πρόβλεψη πιθανών απαντήσεων και η ανάλυσή τους είναι σαφώς δυσκολότερη από τις κλειστές ερωτήσεις.
3. **Ερωτήσεις ημι-ανοιχτές ή ημι-κλειστές.** Σε αυτές τις ερωτήσεις, που συχνά καλούνται και ερωτήσεις “cafeteria”, προβλέπονται οι κυριότερες πιθανές απαντήσεις, όπως σε μία κλειστή ερώτηση, αλλά αφήνεται εξίσου το περιθώριο να προσθέσει ο ερωτώμενος, από μόνος του, και άλλες απαντήσεις έξω από τα καθοριζόμενα πλαίσια. Με αυτόν τον τρόπο, επιτρέπουν σε ορισμένες περιπτώσεις μία περισσότερο ελεύθερη έκφραση στον ερωτώμενο. Σε αυτόν τον τύπο ερωτήσεων οι προκαθορισμένες απαντήσεις θα πρέπει να επιλέγονται με ιδιαίτερη επιμέλεια και να προκύπτουν από προηγούμενες έρευνες, ώστε ένα σημαντικό μέρος των απαντήσεων να μη συγκεντρώνεται στη θέση των ανοιχτών απαντήσεων.

Όσον αφορά την απεικόνιση των προκαθορισμένων απαντήσεων στις κλειστές ερωτήσεις, η πιο συνηθισμένη μέθοδος είναι η παράθεση των απαντήσεων τη μία κάτω από την άλλη. Σε κάθε μία απάντηση ο ερευνητής αντιστοιχεί ένα συγκεκριμένο βάρος βασισμένο σε μία αριθμητική κλίμακα (π.χ. από 0 ως 10 ή από 0 ως 5 ή από -2 ως +2 κ.τ.λ.), αντιστοιχώντας στο μικρότερο αριθμό της κλίμακας την πιο αρνητική απάντηση, ενώ στο μεγαλύτερο αριθμό την πιο θετική.

Παράδειγμα: Πώς κρίνετε το πρόγραμμα ως προς την ακρίβεια και την ορθότητα του αποτελέσματός του;

α) άριστο	→	10
β) πολύ καλό	→	9

γ) καλό	→	7
δ) μέτριο	→	5
ε) σχετικά κακό	→	3
ζ) κακό	→	1
η) απαράδεκτο	→	0

Μία εναλλακτική απεικόνιση της παραπάνω ερώτησης θα μπορούσε να γίνει με τη χρήση της κλίμακας, όπου ο ερωτώμενος θα σημειώνει επάνω της με ένα X τον αριθμό που επιθυμεί.

Σχήμα 3.1: Παράδειγμα κλίμακας

Έχει παρατηρηθεί μερικές φορές ότι οι ερωτώμενοι, κυρίως όταν πρόκειται για ερωτήσεις που αφορούν την προσωπική τους εμπειρία, προτιμούν να παρουσιάσουν υπερεκτιμημένες τις γνώσεις τους και δεν επιλέγουν μία απάντηση που πιθανώς να τους κατέτασσε στους μη έμπειρους χρήστες, ακόμα και αν αυτή είναι η αληθινή επιλογή. Με άλλα λόγια, οι ερωτώμενοι σε αυτές τις περιπτώσεις διστάζουν να επιλέξουν την απάντηση της χαμηλότερης βαθμολογίας. Για να αντιμετωπιστεί αυτό το φαινόμενο, ο ερευνητής μπορεί να διασπάσει μία επιλογή χαμηλού βαθμού σε περισσότερες, δίνοντας όμως τον ίδιο βαθμό σε κάθε μία από αυτές, κάτι που ο ερωτώμενος προφανώς δεν το γνωρίζει.

Έτσι, για παράδειγμα στην ερώτηση “Πόσες περίπου γραμμές κώδικα έχετε γράψει σε C/C++”, αν οι επιλογές είναι οι εξής:

α) Καμία	→	0
β) Κάτω από 2000	→	1
γ) 2000-4000	→	2
δ) 4000-8000	→	3
ε) 8000-15000	→	4
ζ) Πάνω από 15000	→	5

τότε ένας ερωτώμενος που έχει γράψει κάτω από 2000 γραμμές κώδικα μπορεί σε ειδικές περιπτώσεις λόγω ντροπής ή αμηχανίας να μην επιλέξει το (β), αλλά το (γ). Για αυτό το λόγο, η πρώτη επιλογή μπορεί να χωριστεί σε τρεις και τελικά να έχουμε:

α) Καμία	→	0
β) Κάτω από 500	→	1
γ) 500-1000	→	1
δ) 1000-2000	→	1
ε) 2000-4000	→	2
ζ) 4000-8000	→	3
η) 8000-15000	→	4
θ) Πάνω από 15000	→	5

όπου ο ίδιος ερωτώμενος σε αυτήν την περίπτωση δεν θα διστάσει να επιλέξει το (δ), μιας που δεν γνωρίζει ότι αντιστοιχεί στον ίδιο χαμηλό βαθμό.

Ένα άλλο γνωστό πρόβλημα κατά την απάντηση ενός ερωτηματολογίου, όπως αναφέρθηκε σε προηγούμενη ενότητα, είναι η υποκειμενικότητα της αντίληψης των

πελατών για την ποιότητα, γεγονός που οδηγεί σε διαφορές στην έκφραση των απαντήσεων. Η λύση στο παραπάνω πρόβλημα είναι η χρήση προκαθορισμένων απαντήσεων με ξεκάθαρα όρια ανάμεσα στις επιλογές. Μία πλέον διαδεδομένη τεχνική είναι η χρήση της κλίμακας Likert [LIK32], όπου στις ερωτήσεις δίνονται πέντε απαντήσεις-επιλογές ως εξής: [1 = διαφωνώ ισχυρά, 2 = διαφωνώ, 3 = είμαι αναποφάσιτος, 4 = συμφωνώ, 5 = συμφωνώ ισχυρά]. Η κλίμακα Likert εμφανίζεται σε διάφορες παραλλαγές [TRO01], όπου ο αριθμός των επιλογών ή η διάταξη (αύξουσα ή φθίνουσα) των επιλογών μπορεί να διαφέρει.

Όπως δείχνουν τα πειράματα του Yeh [YEH93], δεν είναι αρκετή η παροχή διακριτών απαντήσεων οι οποίες βρίσκονται σε κλίμακα διάταξης, αφού ο τρόπος που ερμηνεύουν οι πελάτες την απόσταση από απάντηση σε απάντηση είναι διαφορετικός. Έτσι, θα πρέπει είτε να χρησιμοποιείται το παράδειγμα της κλίμακας (όπως στο Σχήμα 3.1) είτε οι προκαθορισμένες απαντήσεις να συνοδεύονται από αντίστοιχες ερμηνείες, ώστε να γίνεται εύκολα και ομοιόμορφα κατανοητή η απόσταση ανάμεσά τους.

Παράδειγμα: Πόσο εύκολα βρίσκετε συνήθως αυτό που ψάχνετε όταν χρησιμοποιείτε τη μηχανή αναζήτησης “AnazitisiS”;

- α) Πάρα πολύ εύκολα (στα πρώτα 10 αποτελέσματα).
- β) Πολύ εύκολα (στα πρώτα 10 αποτελέσματα, αρκεί να έχω βάλει τους σωστούς όρους αναζήτησης).
- γ) Σχετικά εύκολα (στις 2-3 πρώτες σελίδες αποτελεσμάτων, ακόμα και όταν δεν βάζω όλους τους όρους αναζήτησης).
- δ) Μέτρια (χρειάζεται συνήθως να προχωρήσω πάνω από 3 σελίδες αποτελεσμάτων για να βρω αυτό που θέλω).
- ε) Σχετικά δύσκολα (κάποιες φορές πρέπει να προχωρήσω πάνω από 7-8 σελίδες αποτελεσμάτων για να βρω αυτό που θέλω).
- ζ) Δύσκολα (κάποιες φορές δεν βρίσκω αυτό που ψάχνω, αν και ξέρω ότι υπάρχει και οι όροι αναζήτησης που βάζω είναι σωστοί).

Πέρα όμως από τα βάρη που πρέπει να θέσει ο ερευνητής σε κάθε προκαθορισμένη επιλογή μίας ερώτησης, ανάλογα βάρη πρέπει να καθοριστούν τόσο για κάθε παράγοντα ή χαρακτηριστικό στα οποία απαρτίζεται η αφηρημένη έννοια της ποιότητας λογισμικού, όσο και για κάθε ερώτηση από το σύνολο των ερωτήσεων που αναφέρονται στον κάθε συγκεκριμένο παράγοντα ή χαρακτηριστικό της ποιότητας. Θα πρέπει δηλαδή πριν από τη διεξαγωγή της έρευνας να καθοριστούν εκείνα τα συστατικά της ποιότητας λογισμικού στα οποία μία εταιρεία παραγωγής λογισμικού εστιάζει περισσότερο. Είναι προφανές, στηριζόμενοι για παράδειγμα στο πρότυπο ISO9126, ότι σε ένα προϊόν λογισμικού που επιθυμούμε κυρίως να είναι εύχρηστο, στον υπολογισμό της τελικής ποιότητάς του ο παράγοντας της ευχρηστίας θα πρέπει ποσοστιαία να έχει μεγαλύτερο βάρος από τους άλλους. Αντίστοιχα, αν ενδιαφερόμαστε περισσότερο για την ευκολία μάθησης του προϊόντος, τότε οι ερωτήσεις που αναφέρονται σε αυτό το υπο-χαρακτηριστικό της ευχρηστίας θα πρέπει να έχουν μεγαλύτερο βάρος από τις ερωτήσεις σχετικά με την κατανοησιμότητα και την πρακτικότητα.

Συνοψίζοντας, θα πρέπει να αναφέρουμε λοιπόν ότι παράλληλα με τη σύνταξη του ερωτηματολογίου, ο υπεύθυνος της έρευνας θα πρέπει να φροντίσει για: α) τον εντοπισμό των βασικών εξωτερικών ποιοτικών χαρακτηριστικών, στην ικανοποίηση των οποίων στοχεύει η εταιρεία παραγωγής λογισμικού, και το ποσοστό συμμετοχής καθενός στο ερωτηματολόγιο, β) τον καθορισμό των μετρήσιμων στόχων, δηλαδή

των υπο-χαρακτηριστικών τα οποία συνθέτουν τα παραπάνω βασικά χαρακτηριστικά, καθώς και το ποσοστό συμμετοχής κάθε υπο-χαρακτηριστικού στη σύνθεση του βασικού χαρακτηριστικού, γ) τον καθορισμό του ποσοστού με το οποίο συμμετέχει κάθε ερώτηση στη μέτρηση κάθε στοχευόμενου υπο-χαρακτηριστικού και δ) τον καθορισμό της κλίμακας των απαντήσεων, δηλαδή τα βάρη στις διάφορες προκαθορισμένες επιλογές στην περίπτωση των κλειστών ερωτήσεων, ανάλογα με την ανάλυση που θα ακολουθήσει.

3.4 Μέθοδος Ανάλυσης Αποτελεσμάτων

Όπως είδαμε και στην ενότητα 3.3.4, ένα από τα τελευταία στάδια κατά τη διεξαγωγή εξωτερικών μετρήσεων είναι η ανάλυση των αποτελεσμάτων. Σε αυτό το στάδιο περιλαμβάνεται δηλαδή η στατιστική επεξεργασία των απαντήσεων που προήλθαν από τους χρήστες που συμμετείχαν στο δείγμα της έρευνας. Θα πρέπει λοιπόν να καθοριστεί επακριβώς η μέθοδος σύμφωνα με την οποία θα επεξεργαστούν τα δεδομένα που προέκυψαν από τη μέτρηση και θα οδηγήσουν στην εξαγωγή σαφών συμπερασμάτων για την άποψη των χρηστών για την ποιότητα ενός προϊόντος λογισμικού.

Στην παρούσα διδακτορική διατριβή χρησιμοποιήθηκαν τέσσερις εναλλακτικοί τρόποι στατιστικής ανάλυσης των αποτελεσμάτων που προέκυψαν από εξωτερικές μετρήσεις που βασίζονταν σε έρευνα με τη χρήση ερωτηματολογίου ή συνέντευξης, καθένας από τους οποίους έχει διαφορετικό κόστος. Ως κόστος υπολογίζουμε τη διεύρυνση της διαδικασίας ανάλυσης και ελέγχου της ορθότητας των απαντήσεων, αλλά και το κόστος που αντιμετωπίζουν οι πελάτες, δηλαδή την αύξηση του μεγέθους του ερωτηματολογίου. Στη συνέχεια, οι τέσσερις αυτές εναλλακτικές αντιμετώψεις παρουσιάζονται διατεταγμένες με αυξανόμενο κόστος. Δηλαδή, η πρώτη αντιμετώπιση είναι η απλούστερη, ενώ η τελευταία η πιο πολύπλοκη, αλλά και αυτή που είναι σε θέση να αντιπαρέρχεται τα περισσότερα προβλήματα [XEN96a].

3.4.1 Μέση γνώμη

Σύμφωνα με τη μέθοδο μέτρησης της μέσης γνώμης του πελάτη, ACO (Average Customer Opinion), όλοι οι χρήστες αντιμετωπίζονται ως ισοδύναμοι από πλευράς γνώσεων και εμπειρίας και ο αριθμός των λαθών αναμένεται να είναι ελάχιστος. Η σχέση που εκφράζει τη μέθοδο αυτή είναι ένας μέσος όρος των κανονικοποιημένων απόψεων των χρηστών και δίδεται παρακάτω:

$$ACO = \frac{\sum_{i=1}^n O_i}{n} \quad (\text{Σχέση 3.4.1})$$

Στην παραπάνω σχέση το n είναι ο συνολικός αριθμός των πελατών του δείγματος, ενώ ως O_i ορίζεται η άποψη του πελάτη i που προκύπτει από τη μέτρηση της άποψης του πελάτη για τα εξωτερικά ποιοτικά χαρακτηριστικά του λογισμικού. Η τιμή του όρου O_i είναι κανονικοποιημένη και κυμαίνεται από 0 έως 1.

3.4.2 Γνώμη με βαρύτητα προσόντων

Κατά τη μέθοδο μέτρησης της γνώμης του πελάτη με βαρύτητα προσόντων, QWCO (Qualifications Weighted Customer Opinion), γίνεται ένας διαχωρισμός στη βαρύτητα της γνώμης του κάθε πελάτη ανάλογα με την εμπειρία του. Οι κατηγορίες

των προσόντων του πελάτη που καθορίζουν και το βαθμό της εμπειρίας του διακρίνονται στο προσωπικό του υπόβαθρο, τη συντακτική του γνώση και τη σημασιολογική του γνώση, όπως αναφέρθηκε στην ενότητα 3.2. Η σχέση που εκφράζει τη μέθοδο αυτή δίδεται παρακάτω:

$$QWCO = \frac{\sum_{i=1}^n (O_i \cdot E_i)}{\sum_{i=1}^n E_i} \quad (\text{Σχέση 3.4.2})$$

Ο επιπρόσθετος παράγοντας E_i στη σχέση αυτή αντιστοιχεί στα προσόντα του πελάτη i και η τιμή του είναι κανονικοποιημένη και παίρνει τιμές από 0 έως 1. Από αυτή τη σχέση λοιπόν είναι προφανές ότι οι πελάτες με μεγαλύτερη εμπειρία έχουν και μεγαλύτερο βάρος γνώμης.

3.4.3 Γνώμη με βαρύτητα προσόντων και ερωτήσεις ασφαλείας

Κατά τη μέθοδο μέτρησης της γνώμης του πελάτη με βαρύτητα προσόντων και ερωτήσεις ασφαλείας, $QWCO_s$ (Qualifications Weighted Customer Opinion with Safeguards), η βαρύτητα της γνώμης του κάθε πελάτη εξαρτάται όχι μόνο από την εμπειρία του, αλλά και από τον αριθμό των ερωτήσεων ασφαλείας στον οποίο απάντησε σωστά. Η μέθοδος αυτή αποσκοπεί στον εντοπισμό και την αντιμετώπιση λαθών που προκύπτουν από απαντήσεις του πελάτη που δεν αντικατοπτρίζουν τις πραγματικές απόψεις του. Ο ρόλος των ερωτήσεων ασφαλείας δεν είναι η μέτρηση των εξωτερικών ποιοτικών χαρακτηριστικών, αλλά ο έλεγχος της ορθότητας των απαντήσεων.

Οι ερωτήσεις ασφαλείας χωρίζονται στις παρακάτω κατηγορίες: α) ερωτήσεις ελέγχου, β) επαναλαμβανόμενες ερωτήσεις με διαφορετική διατύπωση και γ) επαναλαμβανόμενες ερωτήσεις με διαφορετικά είδη επιλογών απαντήσεων. Η σχέση που εκφράζει τη μέθοδο αυτή δίδεται παρακάτω:

$$QWCO_s = \frac{\sum_{i=1}^n \left(O_i \cdot E_i \cdot \frac{S_i}{S_T} \right)}{\sum_{i=1}^n \left(E_i \cdot \frac{S_i}{S_T} \right)} \quad (\text{Σχέση 3.4.3})$$

Ο επιπρόσθετος παράγοντας S_i στη σχέση αυτή αντιστοιχεί στον αριθμό των ερωτήσεων ασφαλείας τις οποίες ο πελάτης i έχει απαντήσει σωστά, ενώ ο παράγοντας S_T είναι ο συνολικός αριθμός ερωτήσεων ασφαλείας που έχουν ενσωματωθεί στο ερωτηματολόγιο.

3.4.4 Γνώμη με βαρύτητα προσόντων και διπλές ερωτήσεις ασφαλείας

Η μέθοδος μέτρησης της γνώμης του πελάτη με βαρύτητα προσόντων και διπλές ερωτήσεις ασφαλείας, $QWCO_{DS}$ (Qualifications Weighted Customer Opinion with Double Safeguards), είναι μία επέκταση της προηγούμενης μεθόδου, κατά την οποία χρησιμοποιούνται ερωτήσεις ασφαλείας όχι μόνο στη μέτρηση της γνώμης του πελάτη, αλλά και στη μέτρηση της εμπειρίας του. Η σχέση που εκφράζει τη μέθοδο αυτή δίδεται παρακάτω:

$$QWCO_{DS} = \frac{\sum_{i=1}^n \left(O_i \cdot E_i \cdot \frac{S_i}{S_T} \cdot P_i \right)}{\sum_{i=1}^n \left(E_i \cdot \frac{S_i}{S_T} \cdot P_i \right)} \quad (\text{Σχέση 3.4.4})$$

Ο επιπρόσθετος παράγοντας P_i παίρνει τιμές 0 ή 1. Συγκεκριμένα η τιμή του P_i είναι ίση με 0 στην περίπτωση κατά την οποία ο πελάτης i έχει απαντήσει έστω και μία ερώτηση ασφαλείας λάθος από αυτές τις ερωτήσεις ασφαλείας που έχουν ενσωματωθεί στο τμήμα μέτρησης των προσόντων του πελάτη. Αντίθετα, η τιμή του P_i είναι ίση με 1 μόνο στην περίπτωση όπου ο πελάτης i έχει απαντήσει σωστά σε όλες αυτές τις ερωτήσεις ασφαλείας. Όπως προκύπτει από την παραπάνω σχέση, οι πελάτες που έχουν δώσει λάθος απάντηση στο τμήμα του ερωτηματολογίου που σχετίζεται με τη μέτρηση της εμπειρίας τους αποκλείονται πρακτικά και δεν συμμετέχουν στην έρευνα, αφού οι απαντήσεις τους δεν λαμβάνονται υπόψη στην ανάλυση.

3.5 Το Εργαλείο CESM

Το εργαλείο CESM (Collecting External Software Measurements) είναι ένα εργαλείο συλλογής και ανάλυσης των εξωτερικών μετρήσεων ποιότητας λογισμικού, το οποίο αναπτύχθηκε στα πλαίσια της παρούσας διδακτορικής διατριβής [XEN99]. Αποσκοπεί στην εκτίμηση της ποιότητας λογισμικού και μπορεί εύκολα να ενοποιηθεί στη σειρά εργαλείων ενός προγράμματος εξασφάλισης ποιότητας μίας εταιρείας παραγωγής λογισμικού.

Το CESM είναι μία εφαρμογή ανεξάρτητη πλατφόρμας και λειτουργικού που είναι εγκατεστημένο στους υπολογιστές των χρηστών που θα συμμετέχουν στις εξωτερικές μετρήσεις, επειδή βασίζεται σε τεχνολογίες διαδικτύου. Για την ακρίβεια, το μόνο που χρειάζεται από το χρήστη είναι ένας φυλλομετρητής διαδικτύου (browser). Επίσης, έχει σχεδιαστεί με τέτοιο τρόπο ώστε να είναι πολυγλωσσικό. Το CESM χρησιμοποιεί ένα πολλαπλής χρήσης ερωτηματολόγιο βασισμένο είτε στα ποιοτικά χαρακτηριστικά του προτύπου ISO9126 είτε στους παράγοντες ποιότητας του μοντέλου του McCall. Κατά τη σύνταξη αυτού του ερωτηματολογίου καθορίζονται τα βάρη α) για κάθε παράγοντα ποιότητας ή ποιοτικό χαρακτηριστικό, β) για κάθε ερώτηση από το σύνολο ερωτήσεων που αναφέρονται στον κάθε παράγοντα ποιότητας ή ποιοτικό χαρακτηριστικό και γ) για τις προεπιλεγμένες απαντήσεις της κάθε ερώτησης, όπως αναφέρθηκε στην παράγραφο 3.3.6.

Με τη χρήση του παραπάνω ερωτηματολογίου διεξάγεται έρευνα στο διαδίκτυο για την ποιότητα ενός συγκεκριμένου προϊόντος λογισμικού όπως την αντιλαμβάνονται οι τελικοί χρήστες. Το εργαλείο προσφέρει στον ερευνητή ανά πάσα στιγμή τη δυνατότητα να αναθεωρεί τις ερωτήσεις του ερωτηματολογίου, να αλλάζει τα διάφορα βάρη που έχει ορίσει, να διαχειρίζεται τις απαντήσεις των χρηστών, να αποθηκεύει τα αποτελέσματα σε βάση δεδομένων και να τα χρησιμοποιεί για στατιστική ανάλυση και παρουσίαση. Η ανάλυση των αποτελεσμάτων γίνεται με μία από τις μεθόδους που παρουσιάστηκαν στην παράγραφο 3.4, οι οποίες έχουν υλοποιηθεί και ενσωματωθεί στο εργαλείο, έτσι ώστε να περιοριστούν τα πιθανά λάθη και η υποκειμενικότητα στις απαντήσεις.

Το εργαλείο CESM είναι υπεύθυνο για εξωτερικές μετρήσεις και μπορεί να χρησιμοποιηθεί παράλληλα με το εργαλείο ATHENA, που παρουσιάστηκε στην

παράγραφο 2.3.1, το οποίο είναι υπεύθυνο για εσωτερικές μετρήσεις. Το CESM μάλιστα μπορεί να χρησιμοποιηθεί για τη διεξαγωγή περιοδικών εξωτερικών μετρήσεων, τα αποτελέσματα των οποίων μπορούν να βοηθήσουν στην εκτίμηση της αναθεώρησης άποψης των χρηστών, αλλά και στον έλεγχο και την επαναρύθμιση εργαλείων που διεξάγουν εσωτερικές μετρήσεις. Η συχνότητα της συλλογής των εξωτερικών μετρήσεων καθορίζεται από την ανάλυση προηγούμενων αποτελεσμάτων και εκτιμώντας τα αποτελέσματα που είναι αποθηκευμένα στη βάση δεδομένων που χρησιμοποιεί το εργαλείο CESM. Με τη χρήση τεχνικών αναθεώρησης άποψης, οι οποίες θα παρουσιαστούν σε επόμενο κεφάλαιο, ο ερευνητής είναι σε θέση να εκτιμήσει τον κατάλληλο χρόνο για τη διεξαγωγή νέων εξωτερικών μετρήσεων και την ανάκτηση νέων και αναθεωρημένων αποτελεσμάτων.

Πέρα όμως από την ευκολία στην ανάλυση των αποτελεσμάτων, στα πλεονεκτήματα του CESM θα πρέπει να προστεθούν η αμεσότητα στην αποστολή και τη λήψη του ερωτηματολογίου και η ελκυστικότητα στη χρήση. Με το CESM η έρευνα βασίζεται σε ένα ερωτηματολόγιο που είναι αναρτημένο σε μία ιστοσελίδα κάνοντας έτσι χρήση του διαδικτύου και αποφεύγοντας α) την πιθανή σπατάλη που έχει η έκδοση και αποστολή έντυπων ερωτηματολογίων, β) την αβεβαιότητα στην παραλαβή από τον τελικό χρήστη και γ) τα πιθανά λάθη συμπλήρωσης των απαντήσεων, αφού πριν την ηλεκτρονική αποστολή τους γίνεται ο απαραίτητος έλεγχος.

3.5.1 Η αρχιτεκτονική του εργαλείου CESM

Το CESM αποτελείται από τα παρακάτω τρία συστατικά:

- Το Σχεδιαστή Ερωτηματολογίων QD (Questionnaire Designer), που είναι μία εφαρμογή με την οποία ο συντάκτης-ερευνητής συνθέτει το ερωτηματολόγιο, καθορίζει τη δομή των ερωτήσεων και ορίζει τα απαραίτητα βάρη.
- Την Ιστοσελίδα του κάθε ερωτηματολογίου WP (Web Page), η οποία προσπελάζεται από τους χρήστες που συμμετέχουν στην έρευνα.
- Το Στατιστικό Αναλυτή Ερωτηματολογίων QSA (Questionnaire Statistical Analyser), που είναι μία εφαρμογή η οποία παρέχει στον ερευνητή τα στατιστικά αποτελέσματα της έρευνας.

Σχήμα 3.2: Η αρχιτεκτονική του εργαλείου CESM

Άλλα βασικά χαρακτηριστικά του CESM είναι η βάση δεδομένων (CESM Database), όπου αποθηκεύονται τα ερωτηματολόγια και οι απαντήσεις των χρηστών και ο εξυπηρετητής παγκόσμιου ιστού (Web Server), στον οποίο παρουσιάζονται οι ιστοσελίδες με τα αναρτημένα ερωτηματολόγια. Το Σχήμα 3.2 παρουσιάζει τη βασική αρχιτεκτονική του εργαλείου CESM.

3.5.2 Σχεδιαστής Ερωτηματολογίων

Ο Σχεδιαστής Ερωτηματολογίων (QD) είναι μία παραθυρική εφαρμογή, υλοποιημένη σε μορφή «μάγου» (wizard) κατά τα βήματα του οποίου ο υπεύθυνος της έρευνας συντάσσει ένα ερωτηματολόγιο για ένα προϊόν λογισμικού. Εγκαθίσταται μόνο στον υπολογιστή του ερευνητή και συνδέεται κατευθείαν με τη βάση δεδομένων του CESM. Παρέχει επίσης τη δυνατότητα σύνταξης πολυγλωσσικών ερωτηματολογίων, με την προϋπόθεση ότι ο ερευνητής δηλώνει τη γλώσσα του κάθε ερωτηματολογίου.

Κατά το πρώτο βήμα σύνταξης ενός νέου ερωτηματολογίου στον QD δηλώνονται ποια εξωτερικά ποιοτικά χαρακτηριστικά από το πρότυπο ISO9126 (ή παράγοντες ποιότητας από το μοντέλο του McCall) θα χρησιμοποιηθούν. Οι ερωτήσεις του ερωτηματολογίου θα πρέπει να ομαδοποιηθούν ανάλογα με το ποιοτικό χαρακτηριστικό στο οποίο αναφέρονται. Σε κάθε ομάδα ερωτήσεων δίνεται το ανάλογο βάρος, το οποίο εξαρτάται από την έμφαση που θέλει να δώσει ο ερευνητής στο ποιοτικό χαρακτηριστικό που η ομάδα αναφέρεται. Μία αντιπροσωπευτική φόρμα της εφαρμογής QD παρουσιάζεται στο Σχήμα 3.3.

The screenshot shows a software window titled "Group Data Form" with a standard Windows-style title bar. Inside the window, there is a "Language" dropdown menu set to "English". Below this is a section titled "Question Groups" containing a table with two columns: "Group Name" and "Weight". The table lists four groups: "Efficiency" (0.2), "Functionality" (0.3), "Reliability" (0.3), and "Usability" (0.2). The "Functionality" row is highlighted. To the right of the table are two arrow buttons (up and down) and two buttons labeled "Add New" and "Delete...". Below the table is a section titled "Group Notes" with a text area containing the text: "According to ISO 9126, Functionality consists of: 1) Suitability, 2) Accuracy, 3) Interoperability and 4) Security". At the bottom of the window are two buttons: "<< Back" and "Questions >>".

Group Name	Weight
Efficiency	0.2
Functionality	0.3
Reliability	0.3
Usability	0.2

Group Notes

According to ISO 9126, Functionality consists of:
1) Suitability, 2) Accuracy, 3) Interoperability and 4) Security

<< Back Questions >>

Σχήμα 3.3: Μία φόρμα του Σχεδιαστή Ερωτηματολογίων του CESM

Κατά το δεύτερο βήμα συντάσσονται οι ερωτήσεις για κάθε ομάδα ερωτήσεων. Επιπλέον, δηλώνεται το βάρος για κάθε μία ερώτηση, σύμφωνα με τη σπουδαιότητά

της. Οι ερωτήσεις πρέπει να είναι πολλαπλής επιλογής, έτσι ώστε να καθοδηγείται ο χρήστης στην επιλογή μίας από τις προκαθορισμένες απαντήσεις οι οποίες είναι διατεταγμένες σε προκαθορισμένη βαθμολογημένη κλίμακα (π.χ. ερωτήσεις που χρησιμοποιούν ποσοστιαίες εκτιμήσεις, αριθμημένες κλίμακες κτλ). Τέλος, κατά το τρίτο βήμα της εφαρμογής συντάσσονται οι πολλαπλές επιλογές της κάθε ερώτησης και δηλώνονται οι τιμές που αντιπροσωπεύουν.

Το εργαλείο CESM παρέχει τη δυνατότητα να χρησιμοποιηθεί ένα ήδη αποθηκευμένο στη βάση ερωτηματολόγιο ενός συγκεκριμένου προϊόντος για άλλα προϊόντα της εταιρείας λογισμικού, με την προϋπόθεση ότι δηλώνεται ξεκάθαρα για ποιο προϊόν απαντάει το ερωτηματολόγιο ο κάθε χρήστης. Παράλληλα, με τον QD επιτρέπεται η πρόσβαση ενός ερωτηματολογίου που είναι αποθηκευμένο στη βάση, έτσι ώστε να αναδιαρθρωθεί ή να γίνουν πιθανές αλλαγές στα βάρη των ποιοτικών χαρακτηριστικών, στα βάρη των ερωτήσεων ή στις τιμές των πολλαπλών επιλογών. Τέλος, ο QD παρέχει τη δυνατότητα κλωνοποίησης, έτσι ώστε να είναι δυνατή η επαναδημοσίευση ενός ήδη αποθηκευμένου ερωτηματολογίου.

3.5.3 Ιστοσελίδα ερωτηματολογίου

Αφού σχεδιαστεί το ερωτηματολόγιο για ένα προϊόν λογισμικού, μπορεί να εκδοθεί αυτόματα στον εξυπηρετητή παγκόσμιου ιστού της εταιρείας λογισμικού. Η ιστοσελίδα (WP) που παράγεται μπορεί να προσπελαστεί για να απαντηθεί το ερωτηματολόγιο από οποιονδήποτε φυλλομετρητή. Στη συνέχεια, οι χρήστες του συγκεκριμένου προϊόντος που συμμετέχουν στην έρευνα ενημερώνονται για την ύπαρξη αυτού του ερωτηματολογίου και καλούνται να επισκεφτούν την παραπάνω ιστοσελίδα και να το απαντήσουν, ώστε να μετρηθεί η άποψή τους για την ποιότητα του προϊόντος.

Για να προσπελαστεί η ιστοσελίδα του ερωτηματολογίου απαιτείται αρχικά μία πιστοποίηση του κάθε χρήστη με κωδικό που του έχει δοθεί από την εταιρεία λογισμικού. Με αυτόν τον τρόπο είναι δυνατή η διαχείριση των επισκέψεων και των απαντήσεων των χρηστών, αλλά επίσης εμποδίζεται η απάντηση του ερωτηματολογίου από μη εγγεγραμμένους χρήστες. Μετά την επιτυχή εισαγωγή στην ιστοσελίδα μέσω του φυλλομετρητή, ανασύρονται τα απαραίτητα δεδομένα από τη βάση δεδομένων του CESM (δηλαδή, οι διάφορες ερωτήσεις και οι προεπιλεγμένες απαντήσεις τους) και γίνεται η αυτόματη απεικόνιση του ερωτηματολογίου στο χρήστη.

Ο χρήστης καλείται να επιλέξει μία απάντηση για κάθε ερώτηση. Κάτω από τις προεπιλεγμένες απαντήσεις, οι οποίες απεικονίζονται σε φθίνουσα σειρά ανάλογα με την τιμή τους, μπορεί ο ερευνητής να επιλέξει να εμφανίζεται και ένα κενό πεδίο (π.χ. Άλλο:...), όπου ο χρήστης να μπορεί να δώσει μία δική του απάντηση αν δεν τον καλύπτει μία από τις προεπιλεγμένες. Σε αυτή την περίπτωση θα πρέπει ο χρήστης παράλληλα να προσδιορίσει την τιμή της δικής του απάντησης επιλέγοντας ταυτόχρονα και μία από τις προεπιλεγμένες. Στο Σχήμα 3.4 παρουσιάζεται ένα παράδειγμα ιστοσελίδας ενός ερωτηματολογίου.

Αφού απαντηθούν όλες οι ερωτήσεις, πέρα από τις απαντήσεις αποθηκεύονται στη βάση δεδομένων οι απαραίτητες πληροφορίες εισαγωγής του χρήστη, δηλαδή το όνομα χρήστη και η ημερομηνία και ώρα απάντησης του ερωτηματολογίου. Κατά τη διεξαγωγή της έρευνας για τη μέτρηση της ποιότητας ενός συγκεκριμένου προϊόντος, κάθε χρήστης μπορεί να απαντήσει το ανάλογο ερωτηματολόγιο μόνο μία φορά. Αν ο ερευνητής επιθυμεί να επαναλάβει αυτή την έρευνα ύστερα από ένα εύλογο χρονικό

διάστημα χρήσης του προϊόντος, θα πρέπει να δώσει το δικαίωμα στους χρήστες να προσπελάσουν την ιστοσελίδα του ερωτηματολογίου για δεύτερη φορά ή να δημιουργήσει έναν κλώνο του ερωτηματολογίου.

Σχήμα 3.4: Παράδειγμα ιστοσελίδας ερωτηματολογίου

3.5.4 Στατιστικός Αναλυτής Ερωτηματολογίων

Το τρίτο συστατικό του εργαλείου CESM είναι ο Στατιστικός Αναλυτής Ερωτηματολογίων (QSA), μία εφαρμογή που εγκαθίσταται μόνο στον υπολογιστή του ερευνητή και του παρέχει τα στατιστικά αποτελέσματα της έρευνας. Συνδέεται άμεσα με τη βάση δεδομένων του CESM από όπου ανακτώνται οι απαντήσεις των χρηστών. Κατά τη χρήση του QSA ο ερευνητής θα πρέπει αρχικά να επιλέξει το προϊόν για το οποίο έχει ήδη διεξαχθεί έρευνα, το ερωτηματολόγιο που χρησιμοποιήθηκε για αυτό και τη χρονική διάρκεια κατά την οποία διεξάχθηκε η έρευνα (σε περίπτωση πολλαπλών ερευνών μέσω του ίδιου ερωτηματολογίου για το ίδιο προϊόν).

Οι βασικές συναρτήσεις που έχουν υλοποιηθεί στον QSA είναι ο υπολογισμός των σχέσεων 3.4.1, 3.4.2, 3.4.3 και 3.4.4 ανάλογα με τη μέθοδο που έχει επιλέξει ο ερευνητής. Η παρακάτω σχέση 3.5.1, GOC_i (Group Opinion of Customer 'i'), υπολογίζει τη γνώμη του χρήστη i για την ποιότητα του προϊόντος σύμφωνα με ένα ποιοτικό χαρακτηριστικό.

$$GOC_i = \frac{\sum_{j=1}^m (Q_j \cdot V_j)}{\sum_{j=1}^m Q_j} \quad (\text{Σχέση 3.5.1})$$

Στην παραπάνω σχέση το m είναι ο συνολικός αριθμός των ερωτήσεων στο ερωτηματολόγιο για το συγκεκριμένο ποιοτικό χαρακτηριστικό, ως Q_j ορίζεται το βάρος που έχει δοθεί στην ερώτηση j και V_j η τιμή της απάντησης που επέλεξε ο χρήστης i .

Η επόμενη σχέση 3.5.2, TOC_i (Total Opinion of Customer 'i'), υπολογίζει τη γνώμη του χρήστη i για την ποιότητα του προϊόντος σύμφωνα με όλα τα ποιοτικά χαρακτηριστικά που επέλεξε ο ερευνητής στο ερωτηματολόγιο.

$$TOC_i = \frac{\sum_{l=1}^k (G_l \cdot GOC_l)}{\sum_{l=1}^k G_l} \quad (\text{Σχέση 3.5.2})$$

Στην παραπάνω σχέση το k είναι ο αριθμός των διαφόρων ομάδων ερωτήσεων, επομένως και των ποιοτικών χαρακτηριστικών που επιλέχτηκαν, ως G_l ορίζεται το βάρος που δόθηκε σε κάθε ποιοτικό χαρακτηριστικό l και το GOC_l είναι η γνώμη του χρήστη i για αυτό το ποιοτικό χαρακτηριστικό.

Χρησιμοποιώντας τα αποτελέσματα που προκύπτουν από τις σχέσεις 3.5.1 και 3.5.2, οι σχέσεις 3.4.1, 3.4.2, 3.4.3 και 3.4.4 μπορούν να χρησιμοποιηθούν είτε για τη μέτρηση της γνώμης των χρηστών για ένα συγκεκριμένο ποιοτικό χαρακτηριστικό (θέτοντας $O_i = GOC_i$) είτε για τη μέτρηση της γνώμης των χρηστών για όλα τα ποιοτικά χαρακτηριστικά (θέτοντας $O_i = TOC_i$). Με αυτό τον τρόπο ο ερευνητής μπορεί να ανακαλύψει σε ποια ποιοτικά χαρακτηριστικά τα αποτελέσματα της έρευνας για ένα προϊόν λογισμικού είναι ικανοποιητικά ή όχι.

Παράλληλα, ο QSA παρέχει στον ερευνητή διάφορα ακόμα στατιστικά και διαχειριστικά δεδομένα που προκύπτουν από την έρευνα που διεξήχθηκε. Η εφαρμογή αυτή υποστηρίζει επίσης ιστογράμματα και πίτες, έτσι ώστε να γίνεται οπτικοποιημένη και περισσότερο αντιληπτή η παρουσίαση των αποτελεσμάτων.

Κεφάλαιο**4**

Συσχέτιση Εσωτερικών και Εξωτερικών Μετρικών Λογισμικού

Στο κεφάλαιο αυτό παρουσιάζεται η μέθοδος συσχέτισης εσωτερικών μετρικών λογισμικού με τα εξωτερικά ποιοτικά χαρακτηριστικά ενός προγράμματος λογισμικού. Η μέθοδος αυτή βασίζεται στις μεθόδους και τις τεχνικές που αναφέρθηκαν στα προηγούμενα κεφάλαια. Παράλληλα, παρουσιάζεται μία περιπτωσιολογική μελέτη που διεξήχθη με την ταυτόχρονη εφαρμογή εσωτερικών και εξωτερικών μετρήσεων και γίνεται μελέτη της συσχέτισης των αποτελεσμάτων που προέκυψαν από τις μετρήσεις αυτές [XEN96b].

4.1 Συσχέτιση εσωτερικών και εξωτερικών μετρήσεων

Σύμφωνα με το πρότυπο ISO9126, η ποιότητα λογισμικού χωρίζεται σε έξι βασικά χαρακτηριστικά [ISO01], όπως αναφέρθηκε στην ενότητα 1.4.2. Αυτά τα ποιοτικά χαρακτηριστικά, μαζί με τα υπο-χαρακτηριστικά τους, μπορούν να διακριθούν σε δύο διαφορετικές ομάδες: σε αυτά που προσανατολίζονται περισσότερο στους χρήστες και σε αυτά που προσανατολίζονται περισσότερο στους κατασκευαστές προγραμμάτων λογισμικού. Οι τελικοί χρήστες ενδιαφέρονται κυρίως για χαρακτηριστικά όπως η λειτουργικότητα, η αξιοπιστία, η ευχρηστία και η αποδοτικότητα, καθώς και για τα υπο-χαρακτηριστικά αυτών, επειδή αυτά μπορούν να εκτιμηθούν με τη χρήση ενός προϊόντος λογισμικού. Αντίθετα, οι κατασκευαστές λογισμικού ενδιαφέρονται περισσότερο για χαρακτηριστικά όπως η συντηρησιμότητα και η μεταφερσιμότητα, καθώς και για τα υπο-χαρακτηριστικά αυτών, αφού αυτά σχετίζονται άμεσα με το κόστος της προσπάθειας κατά την ανάπτυξη ενός προϊόντος λογισμικού.

Όπως αναφέρθηκε στα προηγούμενα κεφάλαια, οι εσωτερικές μετρήσεις με τη χρήση μετρικών μπορούν να γίνουν αυτόματα και τα αποτελέσματά τους είναι εύκολο να συλλεχθούν και να αναλυθούν με στατιστικό τρόπο, σε αντίθεση με τις εξωτερικές μετρήσεις. Παρόλα αυτά, οι εσωτερικές μετρικές δεν προσφέρουν συνήθως πληροφορίες υψηλού επιπέδου που σχετίζονται με την ποιότητα του προϊόντος, κάτι που επιτυγχάνεται σαφώς με τη διεξαγωγή εξωτερικών μετρήσεων. Επιπρόσθετα, η εύρεση εσωτερικών μετρικών οι οποίες ενσωματωμένες σε μία μέθοδο εσωτερικών

μετρήσεων θα είχαν απόλυτη συσχέτιση με τις μετρήσεις των εξωτερικών ποιοτικών χαρακτηριστικών που προσανατολίζονται περισσότερο στους τελικούς χρήστες είναι πολύ δύσκολη. Για την ακρίβεια, εάν υπήρχε ένα τέτοιο σύνολο μετρικών, δεν θα υπήρχε ανάγκη για διεξαγωγή εξωτερικών μετρήσεων, αφού η τήρηση των ορίων που τίθενται από αυτές τις μετρικές θα υποκαθιστούσε το πρόγραμμα εξασφάλισης ποιότητας. Ο λόγος όμως που δεν μπορεί να βρεθεί ένα τέτοιο σύνολο μετρικών είναι ότι η ποιότητα συντίθεται από παράγοντες με υψηλό επίπεδο αφαίρεσης, κυρίως όταν οι παράγοντες αυτοί αφορούν χαρακτηριστικά που προσανατολίζονται στους τελικούς χρήστες. Έτσι, δεν μπορούν να σχετιστούν πάντοτε άμεσα με χαρακτηριστικά που μετρώνται βασισμένα στις εσωτερικές μετρικές. Για παράδειγμα, η αξιοπιστία ενός προγράμματος θα μπορούσε να σχετιστεί άμεσα με το μετρήσιμο αριθμό των λαθών που αναφέρθηκαν κατά τη χρήση του, αλλά δεν μπορεί να συμβεί κάτι ανάλογο στην περίπτωση της ευχρηστίας.

Όπως αναφέρει ο Fenton στο αξίωμά του για την Τεχνολογία Λογισμικού, η καλή εσωτερική δομή συνεπάγεται καλή εξωτερική ποιότητα. Μάλιστα, αν η παραπάνω υπόθεση δεν ισχύει, τότε όλα τα χρόνια έρευνας στην Τεχνολογία Λογισμικού είναι χωρίς νόημα [FEN97]. Το ζητούμενο είναι κατά πόσο μία μέθοδος εσωτερικών μετρήσεων μπορεί να προβλέψει δυσάρεστες καταστάσεις, δηλαδή κατά πόσο οι εσωτερικές μετρήσεις μπορούν να εντοπίσουν τμήματα του λογισμικού τα οποία ενδέχεται να δημιουργήσουν προβλήματα, τόσο κατά τις εσωτερικές στην επιχείρηση φάσεις αξιολόγησης (έλεγχο και συντήρηση), όσο και κατά την τελική αξιολόγηση από τους πελάτες. Φυσιολογικά, ένα πρόγραμμα εξασφάλισης ποιότητας βοηθάει μία εταιρεία να βελτιώσει τη διαδικασία παραγωγής λογισμικού, βελτιώνοντας αρκετά ποιοτικά χαρακτηριστικά. Είναι προφανές ότι ένα βασικό μέρος της εξωτερικής ποιότητας είναι η ίδια η γνώμη του τελικού χρήστη για την ποιότητα ενός προγράμματος λογισμικού. Όμως, το ζητούμενο είναι το πώς και σε ποιο βαθμό αυτή η γνώμη σχετίζεται με την τήρηση των ορίων στις εσωτερικές μετρικές. Αυτό το ζητούμενο μπορεί να αναλυθεί στα δύο ερωτήματα που ακολουθούν:

1. Ένα έργο το οποίο ικανοποιεί ένα πρόγραμμα εξασφάλισης ποιότητας βασισμένο σε εσωτερικές μετρήσεις θα έχει αντίστοιχη επιτυχία στις μετρήσεις της άποψης των πελατών για την ποιότητά του;
2. Ένα έργο το οποίο απέτυχε να ικανοποιήσει ένα πρόγραμμα εξασφάλισης ποιότητας βασισμένο σε εσωτερικές μετρήσεις θα γνωρίσει επίσης αποτυχία στις μετρήσεις της άποψης των πελατών για την ποιότητά του;

Η απάντηση στο πρώτο ερώτημα είναι πως δεν υπάρχει εγγύηση ότι ένα έργο το οποίο ικανοποίησε το πρόγραμμα εσωτερικών μετρήσεων θα έχει αντίστοιχα ικανοποιητικά αποτελέσματα στις εξωτερικές μετρήσεις. Αυτό οφείλεται στο γεγονός ότι οι εσωτερικές μετρήσεις δεν μπορούν να καλύψουν όλους τους εξωτερικούς ποιοτικούς παράγοντες. Αντίθετα, για το δεύτερο ερώτημα η απάντηση είναι ότι η αποτυχία στο πρόγραμμα εσωτερικών μετρήσεων κατά κανόνα σημαίνει και αποτυχία στις εξωτερικές μετρήσεις. Αυτό οφείλεται στο γεγονός ότι αποτυχία στις εσωτερικές μετρήσεις σημαίνει αποτυχία τήρησης κανόνων και αρχών στην παραγωγή λογισμικού, αδυναμία ελέγχου του κώδικα, αδυναμία εντοπισμού εσωτερικών στόχων κτλ. [XEN96b]. Όλα αυτά σχεδόν πάντα αντικατοπτρίζονται στην ποιότητα του τελικού προϊόντος όπως την αντιλαμβάνεται ο πελάτης. Είναι σχεδόν αδύνατο για ένα έργο να μην πληροί εσωτερικούς ποιοτικούς περιορισμούς, όπως τήρηση γενικών κανόνων προγραμματισμού και αρχών της Τεχνολογίας Λογισμικού, και παρόλα αυτά να κρίνεται ικανοποιητικό από τους πελάτες. Κατά συνέπεια, αποτυχία σε

εσωτερικές μετρικές θα σημαίνει και αποτυχία και στην άποψη των πελατών για την ποιότητα.

Οι επιτυχείς εσωτερικές μετρήσεις μπορούν να χαρακτηριστούν ως επαλήθευση της τήρησης κανόνων και αρχών κατά τη διαδικασία παραγωγής λογισμικού και συνέπεια με τις μετρήσιμες ποιοτικές προδιαγραφές, όπως αυτές καθορίζονται από την επιχείρηση. Αυτό συνεπάγεται ότι η επιχείρηση είναι σε θέση να παράγει κώδικα τηρώντας μία μεθοδολογία παραγωγής και έχοντας τη διαδικασία παραγωγής κάτω από έλεγχο, έτσι ώστε το παραγόμενο προϊόν να πληροί τις εσωτερικές προδιαγραφές. Στην περίπτωση όμως αποτυχίας των εσωτερικών μετρήσεων παρουσιάζονται αδυναμίες στην τήρηση των εσωτερικών ποιοτικών προδιαγραφών και παράγεται κώδικας που χαρακτηρίζεται ως χαμηλής ποιότητας. Κατά κανόνα τέτοιος κώδικας δεν ικανοποιεί τους πελάτες και συνήθως αυτό έχει ως συνέπεια να υπάρχει ανάλογη αποτυχία και στις εξωτερικές μετρήσεις. Η περιπτωσιολογική μελέτη που ακολουθεί είναι ένα πείραμα που επιβεβαιώνει αυτή τη διαπίστωση.

4.2 Περιπτωσιολογική μελέτη

Ο βασικός σκοπός της περιπτωσιολογικής μελέτης ήταν η πειραματική εξέταση των δύο ερωτημάτων που αναφέρθηκαν στην προηγούμενη παράγραφο. Η περιπτωσιολογική μελέτη ασχολήθηκε με 46 όμοια έργα, τα οποία υλοποιούσαν λογισμικό με ακριβώς τις ίδιες προδιαγραφές και στόχους. Συγκεκριμένα, υλοποιούσαν διαδικασίες μορφοποίησης κειμένου και αναζήτησης και ήταν αυτόνομα εργαλεία με περιβάλλον επικοινωνίας από το χρήστη, τα οποία λειτουργούσαν κάτω από το λειτουργικό σύστημα Ms-DOS. Όπως παρατηρήθηκε, κάθε έργο αποτελούταν κατά μέσο όρο από 4 τμήματα (modules) και κάθε τμήμα αποτελούταν από 1 έως 14 ρουτίνες. Τα 46 αυτά έργα δημιουργούσαν τις κατάλληλες προϋποθέσεις για τη διεξαγωγή της περιπτωσιολογικής μελέτης λόγω της ύπαρξης των παρακάτω ιδιαιτεροτήτων:

- Ύπαρξη πολλών διαφορετικών έργων με τις ίδιες τυπικές προδιαγραφές που υλοποιούσαν λογισμικό για τον ίδιο σκοπό.
- Χρήση της ίδιας γλώσσας προγραμματισμού (ANSI C) για την υλοποίηση του κάθε έργου.
- Εντοπισμένες μεγάλες διαφορές στο στυλ προγραμματισμού κάθε έργου και στον τρόπο σχεδίασης και υλοποίησής τους, αφού δεν είχαν παραχθεί από τους ίδιους προγραμματιστές.
- Κανένα από τα έργα δεν παράχθηκε βασισμένο σε κάποιο πρόγραμμα εξασφάλισης ποιότητας.
- Κανένα από τα έργα δεν είχε ελεγχθεί μεθοδικά από ανάλογη ομάδα ελέγχου, επομένως αρκετά από αυτά παρουσίαζαν ποιοτικά προβλήματα που ήταν εύκολο να εντοπιστούν.

Οι παραπάνω ιδιαιτερότητες δεν θα μπορούσαν να συμβαίνουν εύκολα σε μία επιχείρηση και δημιουργούσαν ένα κατάλληλο περιβάλλον μελέτης λόγω της μεγάλης διαφοροποίησης και απόκλισης μεταξύ τους. Η απόκλιση αυτή εντοπιζόταν τόσο ως προς την εκπλήρωση των προδιαγραφών του έργου όσο και ως προς τα εξωτερικά ποιοτικά χαρακτηριστικά.

4.2.1 Εσωτερικές μετρήσεις

Για τη διεξαγωγή των εσωτερικών μετρήσεων χρησιμοποιήθηκαν τρία είδη μετρικών: οι μετρικές της επιστήμης του λογισμικού του Halstead [HAL75], η μετρική πολυπλοκότητας του McCabe [MCB76] και η μετρική πολυπλοκότητας δομών δεδομένων του Tsai [TSA86]. Οι παραπάνω μετρικές είναι αντιπροσωπευτικές για κάθε μία από τις τρεις διαφορετικές κατηγορίες μετρικών, δηλαδή μετρικών μεγέθους, δομής και δομών δεδομένων. Συνεπώς, η ταυτόχρονη εφαρμογή τους αποτελεί μία ομάδα που καλύπτει το τρίπτυχο μέγεθος – δομή – δομές δεδομένων.

Οι εσωτερικές μετρήσεις έγιναν αυτοματοποιημένα και τα αποτελέσματα που προέκυψαν για κάθε μία από τις μετρήσιμες ποσότητες αναλύθηκαν με τη χρήση ενός στατιστικού εργαλείου. Για τον εντοπισμό της ύπαρξης ή μη συσχέτισης ανάμεσα στις εσωτερικές και τις εξωτερικές μετρήσεις δόθηκε ιδιαίτερη βαρύτητα όχι μόνο στις μετρικές που επιλέχθηκαν, οι οποίες παρουσιάστηκαν στην ενότητα 2.2, αλλά και στο συνδυασμό εφαρμογής των μετρικών του παραπάνω τρίπτυχου. Στη συνέχεια παρουσιάζονται ενδεικτικά τρεις μετρικές, καθώς και μία συνδυαστική μέτρηση που ενσωματώνει βασικές μετρήσιμες ποσότητες από το τρίπτυχο αυτό.

Η πρώτη μετρική που χρησιμοποιήθηκε ήταν το συνολικό επίπεδο γλώσσας ενός προγράμματος λ_{sum} και κατά τον υπολογισμό του δόθηκε μεγαλύτερη βαρύτητα στις μεγαλύτερες ρουτίνες που το αποτελούν. Ο υπολογισμός του γίνεται από τη σχέση 4.2.1, όπου λαμβάνεται υπόψη το επίπεδο γλώσσας λ_i κάθε ρουτίνας i επί το μήκος N_i αυτής της ρουτίνας προς το συνολικό μήκος του προγράμματος (που είναι το άθροισμα των μηκών όλων των ρουτινών του προγράμματος). Επομένως, οι μεγαλύτερες ρουτίνες επηρεάζουν το συνολικό επίπεδο γλώσσας του προγράμματος λ_{sum} σε μεγαλύτερο βαθμό από τις μικρότερες.

$$\lambda_{sum} = \frac{\sum (N_i \cdot \lambda_i)}{\sum N_i} \quad (\text{Σχέση 4.2.1})$$

Όπως είναι γνωστό, μία χαμηλή τιμή στο επίπεδο γλώσσας είναι ένδειξη πιθανών προβλημάτων στον κώδικα. Ειδικότερα, τα έργα για τα οποία το μετρούμενο επίπεδο γλώσσας απέχει πολύ από το μέσο όρο της γλώσσας ANSI C (που είναι περίπου 1,9) μπορεί πιθανώς να εμφανίσουν προβλήματα.

Η δεύτερη μετρική που χρησιμοποιήθηκε ήταν ο λόγος του αθροίσματος των εκτιμητών μήκους όλων των ρουτινών του προγράμματος δια το συνολικό μήκος τους προγράμματος. Ο υπολογισμός του λόγου αυτού R γίνεται από τη σχέση 4.2.2, όπου λαμβάνεται υπόψη ο εκτιμητής μήκους $Nest_i$ και το μήκος N_i κάθε ρουτίνας i . Η συσχέτιση του εκτιμητή μήκους με το μήκος είναι επιθυμητός παράγοντας και αποτελεί μία ένδειξη για πιθανές προγραμματιστικές ατέλειες του κώδικα. Συγκεκριμένα, δεν είναι επιθυμητό ο εκτιμητής μήκους να είναι πολύ μικρότερος από το μήκος προγράμματος.

$$R = \frac{\sum Nest_i}{\sum N_i} \quad (\text{Σχέση 4.2.2})$$

Η τρίτη μετρική που χρησιμοποιήθηκε ήταν το πηλίκο του αριθμού 10 προς τη μέση κυκλωματική πολυπλοκότητα ενός προγράμματος $V(g)_{sum}$ κατά τον υπολογισμό της οποίας δόθηκε μεγαλύτερη βαρύτητα στις μεγαλύτερες ρουτίνες που το αποτελούν. Δηλαδή, η μετρική που χρησιμοποιήθηκε ήταν το πηλίκο $10/V(g)_{sum}$ και ο παράγοντας $V(g)_{sum}$ δίνεται από τη σχέση 4.2.3, όπου λαμβάνεται υπόψη η

κυκλωματική πολυπλοκότητα $V(g)_i$ και το μήκος N_i κάθε ρουτίνας i προς το συνολικό μήκος του προγράμματος (που είναι το άθροισμα των μηκών όλων των ρουτινών του προγράμματος). Επομένως, οι μεγαλύτερες ρουτίνες επηρεάζουν τη μέση κυκλωματική πολυπλοκότητα του προγράμματος $V(g)_{sum}$ σε μεγαλύτερο βαθμό από τις μικρότερες.

$$V(g)_{sum} = \frac{\sum (N_i \cdot V(g)_i)}{\sum N_i} \quad (\text{Σχέση 4.2.3})$$

Ο αριθμός 10 που χρησιμοποιήθηκε σε αυτή τη μετρική έχει προταθεί από τον McCabe ως η μέγιστη αποδεκτή κυκλωματική πολυπλοκότητα μίας ρουτίνας. Έτσι, τα προγράμματα που έχουν αποτέλεσμα μικρότερο της μονάδας σημαίνει πως έχουν υψηλή κυκλωματική πολυπλοκότητα και όσο μικρότερη τιμή, τόσο περισσότερο αυξάνει η πιθανότητα να παρουσιάζουν προβλήματα.

Όπως ήδη αναφέρθηκε, για την καλύτερη μελέτη της συσχέτισης εσωτερικών και εξωτερικών μετρικών λογισμικού χρησιμοποιήθηκε μία συνδυαστική μετρική που ενσωματώνει βασικές μετρήσιμες ποσότητες από το τρίπτυχο μέγεθος – δομή – δομές δεδομένων. Η χρήση της μετρικής αυτής έγινε καθαρά για λόγους οπτικοποίησης των δεδομένων, αφού αποτελεί μία συνολική μέτρηση των βασικών στοιχείων των μετρικών του τριπτύχου. Δεν μετράει κάτι συγκεκριμένο, αλλά δίνει τη γενική εικόνα κάθε προγράμματος όσον αφορά τις εσωτερικές μετρήσεις. Ο υπολογισμός της συνδυαστικής αυτής μετρικής TIM (Total Internal Measurements) γίνεται από τη σχέση 4.2.4, όπου λ_{sum} , R και $V(g)_{sum}$ είναι οι μετρικές που υπολογίστηκαν από τις σχέσεις 4.2.1, 4.2.2 και 4.2.3 αντίστοιχα και T είναι ο μεγαλύτερος βαθμός πολωνύμου πολυπλοκότητας δομών δεδομένων του έργου που προκύπτει σύμφωνα με τη μετρική του Tsai (ενότητα 2.2.3).

$$TIM = 0,2 \cdot \lambda_{sum} + 0,2 \cdot R + 0,4 \cdot \frac{10}{V(g)_{sum}} + 0,2 \cdot \frac{1}{T+1} \quad (\text{Σχέση 4.2.4})$$

4.2.2 Εξωτερικές μετρήσεις

Οι εξωτερικές μετρήσεις βασίστηκαν στη μέση γνώμη των πελατών με βαρύτητα προσόντων, QWCO (Qualifications Weighted Customer Opinion, ενότητα 3.4.2), όπου γίνεται ένας διαχωρισμός στη βαρύτητα της γνώμης του κάθε πελάτη ανάλογα με την εμπειρία του. Στη συγκεκριμένη μελέτη κάθε χρήστης έπρεπε να αξιολογήσει και τα 46 διαφορετικά προγράμματα. Άρα κάθε χρήστης έπρεπε να απαντήσει σε 47 ερωτηματολόγια (46 ερωτηματολόγια για τη μέτρηση της γνώμης του O_i για κάθε πρόγραμμα και ένα ερωτηματολόγιο για τη μέτρηση των προσόντων του E_i). Λόγω του μεγάλου αριθμού ερωτηματολογίων που έπρεπε να απαντηθούν ο αριθμός των χρηστών που συμμετείχε στην περιπτώσιολογική μελέτη έμεινε χαμηλός, αφού περισσότερη έμφαση έπρεπε να δοθεί στην κάλυψη μεγάλου αριθμού προγραμμάτων για τη μελέτη των διαφορών οι οποίες προκύπτουν από πρόγραμμα σε πρόγραμμα.

Τα ερωτηματολόγια μέτρησης γνώμης των χρηστών για τα 46 προγράμματα εστίασαν στα εξωτερικά ποιοτικά χαρακτηριστικά του προτύπου ISO9126 που αφορούν τον τελικό χρήστη, δηλαδή τη λειτουργικότητα, την αξιοπιστία, την ευχρηστία και την αποδοτικότητα. Από τους χρήστες ζητήθηκε να αξιολογήσουν όλα τα προγράμματα με τον ίδιο βαθμό αυστηρότητας, δηλαδή αν είναι αυστηροί στην κρίση ενός προγράμματος, να είναι ανάλογα αυστηροί στην κρίση και των υπολοίπων.

Όπως είναι προφανές, η αξιολόγηση βασίστηκε αποκλειστικά στη χρήση των προγραμμάτων που θα μπορούσε να γίνει σε καθημερινή βάση. Θα πρέπει να σημειωθεί ότι σύμφωνα με τον Thomas (THO94), όταν οι πελάτες πρόκειται να αγοράσουν ένα προϊόν λογισμικού, η κρίση τους επηρεάζεται όχι μόνο από την ποιότητά του, αλλά και από άλλους παράγοντες, όπως η αγοραστική τιμή του, η εταιρεία που το παρήγαγε, η αναγνώριση ή η επίκριση που δέχτηκε από κριτικές για το ίδιο το λογισμικό, η δυνατότητα συνεργασίας του με άλλα προγράμματα κτλ. Στη συγκεκριμένη μελέτη, αφού τα προγράμματα που χρησιμοποιήθηκαν δεν ήταν προς πώληση, οι παραπάνω παράγοντες δεν είχαν καμία βαρύτητα. Επομένως, η γνώμη των χρηστών εστίασε αποκλειστικά στην ποιότητα των προγραμμάτων ως τελικό προϊόν.

4.2.3 Αποτελέσματα και συσχετίσεις

Στον πίνακα 4.1 παρουσιάζονται τα αποτελέσματα των εξωτερικών μετρήσεων για τα 46 προγράμματα. Τα αποτελέσματα είναι διατεταγμένα κατά αύξουσα σειρά στις γραμμές του πίνακα, δηλαδή πρώτα παρουσιάζονται τα προγράμματα που έλαβαν τη χαμηλότερη βαθμολογία και ακολουθούν τα προγράμματα που έλαβαν υψηλότερη βαθμολογία.

Αποτελέσματα εξωτερικών μετρήσεων QWCO						
0,06	0,12	0,12	0,14	0,16	0,19	0,19
0,21	0,23	0,26	0,28	0,30	0,30	0,30
0,31	0,33	0,33	0,34	0,34	0,39	0,40
0,41	0,42	0,43	0,43	0,44	0,45	0,46
0,47	0,48	0,48	0,48	0,51	0,54	0,56
0,60	0,60	0,60	0,67	0,75	0,76	0,78
0,80	0,86	0,88	0,94			

Πίνακας 4.1: Εξωτερικές μετρήσεις

Τα αποτελέσματα είναι κανονικοποιημένα και οι δυνατές τιμές για κάθε πρόγραμμα κυμαίνονται από 0 έως 1. Τιμές κοντά στο 0 σημαίνουν ότι οι χρήστες εκτίμησαν πως η ποιότητα του προγράμματος σε σχέση με τα εξωτερικά χαρακτηριστικά που ενδιαφέρουν τον τελικό χρήστη ήταν πολύ χαμηλή. Αντίθετα, τιμές κοντά στο 1 σημαίνουν ότι οι χρήστες εκτίμησαν πως η ποιότητα αυτή του προγράμματος είναι αρκετά υψηλή.

Σχήμα 4.1: Κατανομή των εξωτερικών μετρήσεων

Τα αποτελέσματα των εξωτερικών μετρήσεων ακολουθούν μία κανονική κατανομή, έχοντας μέση τιμή 0,437 και τυπική απόκλιση 0,219. Στο Σχήμα 4.1 παρουσιάζεται η κατανομή των εξωτερικών μετρήσεων για τα 46 προγράμματα.

Στη συνέχεια παρουσιάζονται τα αποτελέσματα των εσωτερικών μετρήσεων. Θα πρέπει να σημειωθεί εκ των προτέρων ότι οι τιμές των μετρήσεων που παρουσιάζονται στους παρακάτω πίνακες είναι διατεταγμένες με τον ίδιο τρόπο όπως αυτόν του πίνακα 4.1, ώστε να αντιστοιχούν στο ίδιο πρόγραμμα και να είναι δυνατή η σύγκριση με τις εξωτερικές μετρήσεις. Δηλαδή η πρώτη τιμή στον κάθε πίνακα αντιστοιχεί στο ίδιο πρόγραμμα, αντίστοιχα για τη δεύτερη τιμή, την τρίτη κ.ο.κ.

Συγκεκριμένα, στον πίνακα 4.2 παρουσιάζονται τα αποτελέσματα της ποσότητας λ_{sum} για τα 46 προγράμματα. Η συσχέτιση των αποτελεσμάτων αυτών με τις εξωτερικές μετρήσεις ήταν θετική και βρέθηκε ίση με 81,43%.

Αποτελέσματα λ_{sum}						
0,609	0,540	0,559	0,627	0,440	1,085	0,524
0,690	1,181	1,060	1,189	1,052	1,207	1,146
1,064	1,542	1,540	1,656	1,550	1,049	1,072
2,789	1,581	1,130	1,872	1,802	1,160	1,660
1,460	1,278	1,620	2,110	1,901	1,880	1,670
1,758	2,045	1,770	2,090	1,963	1,412	2,603
1,803	2,450	2,340	2,575			

Πίνακας 4.2: Εσωτερικές μετρήσεις του επιπέδου γλώσσας

Στον πίνακα 4.3 παρουσιάζονται τα αποτελέσματα του λόγου R (εκτιμητής μήκους προγράμματος προς μήκος προγράμματος) για τα 46 προγράμματα. Η συσχέτιση αυτών των αποτελεσμάτων με τις εξωτερικές μετρήσεις ήταν θετική και βρέθηκε ίση με 66,62%.

Αποτελέσματα $R=N_{\text{est}}/N$						
0,594	0,464	0,798	0,657	0,337	0,402	0,769
0,580	0,592	0,552	0,971	0,279	1,355	0,996
0,966	1,221	0,584	0,565	0,556	1,720	0,443
0,871	1,405	0,934	1,571	1,238	0,934	0,834
0,845	1,167	1,110	0,913	1,444	0,933	1,123
0,799	1,482	1,040	1,330	1,146	1,112	1,152
0,885	1,553	1,770	1,550			

Πίνακας 4.3: Εσωτερικές μετρήσεις του εκτιμητή μήκους ανά μήκος

Στον πίνακα 4.4 παρουσιάζονται τα αποτελέσματα του πηλίκου $10/V(g)_{\text{sum}}$ για τα 46 προγράμματα. Η συσχέτιση αυτών των αποτελεσμάτων με τις εξωτερικές μετρήσεις ήταν θετική και βρέθηκε ίση με 54,94%.

Τέλος, στον πίνακα 4.5 παρουσιάζονται τα αποτελέσματα της συνδυαστικής εσωτερικής μετρικής TIM για τα 46 προγράμματα. Και σε αυτή την περίπτωση, η συσχέτιση των αποτελεσμάτων της TIM με τις εξωτερικές μετρήσεις ήταν θετική και βρέθηκε ίση με 70,44%.

Αποτελέσματα $10/V(g)_{sum}$						
0,351	1,410	2,683	0,712	0,222	0,574	1,912
0,599	0,563	0,670	1,028	0,566	2,190	0,443
1,041	1,024	1,330	0,680	1,780	2,113	0,691
1,170	1,297	0,833	0,969	1,715	0,833	1,450
1,480	0,902	1,230	1,540	0,685	1,550	1,880
0,798	1,322	2,770	1,990	1,516	1,633	2,188
1,887	2,770	3,110	1,900			

Πίνακας 4.4: Εσωτερικές μετρήσεις της κυκλωματικής πολυπλοκότητας

Αποτελέσματα TIM						
0,581	0,965	1,545	0,742	0,444	0,727	1,223
0,694	0,780	0,660	1,043	0,693	1,488	0,806
1,022	1,162	0,980	0,916	1,000	1,599	0,779
1,400	1,316	0,946	1,276	1,494	0,952	0,930
0,970	1,050	1,100	0,934	1,143	1,200	1,220
1,031	1,434	1,234	2,000	1,428	1,358	1,826
1,493	1,750	1,730	1,785			

Πίνακας 4.5: Εσωτερικές μετρήσεις της συνδυαστικής μετρικής TIM

Τα αποτελέσματα της μετρικής TIM ακολουθούν επίσης μία κανονική κατανομή, έχοντας μέση τιμή 1,149 και τυπική απόκλιση 0,363. Στο Σχήμα 4.2 παρουσιάζεται η κατανομή των μετρήσεων της μετρικής αυτής για τα 46 προγράμματα.

Σχήμα 4.2: Κατανομή των εσωτερικών μετρήσεων της TIM

Για να μελετήσουμε καλύτερα τη σχέση εσωτερικών και εξωτερικών μετρικών χρησιμοποιήθηκε η τεχνική της γραφικής απεικόνισης με scatter plots (διαγράμματα διασποράς). Το scatter plot είναι η γραφική απεικόνιση των δεδομένων από δύο μετρήσεις για μία συλλογή από δεδομένα, για τα οποία οι μετρήσεις της μίας κατηγορίας απεικονίζονται στον οριζόντιο άξονα (άξονας x) και οι μετρήσεις της άλλης κατηγορίας απεικονίζονται στον κατακόρυφο άξονα (άξονας y). Κάθε πρόγραμμα απεικονίζεται με ένα σημάδι στο επίπεδο που ορίζεται από τις συντεταγμένες των μετρήσεων σε κάθε άξονα.

Στο Σχήμα 4.3 παρουσιάζεται το scatter plot για τις εξωτερικές μετρήσεις (στον άξονα x) και το λ_{sum} (στον άξονα y). Η διαγώνιος, που ονομάζεται και γραμμή συσχέτισης, είναι ο γεωμετρικός τύπος των σημείων που αντιστοιχούν σε συσχέτιση 100% ανάμεσα στις δύο μετρικές.

Σχήμα 4.3: Scatter plot QWCO και λ_{sum}

Με αντίστοιχο τρόπο παρουσιάζονται στα ακόλουθα σχήματα τα scatter plots όπου στον άξονα x απεικονίζονται οι εσωτερικές μετρήσεις και στον άξονα y απεικονίζεται ο λόγος R (Σχήμα 4.4), το πηλίκο $10/V(g)_{\text{sum}}$ (Σχήμα 4.5) και η συνδυαστική μετρική TIM (Σχήμα 4.6).

Σχήμα 4.4: Scatter plot QWCO και R

Σχήμα 4.5: Scatter plot QWCO και $10/V(g)_{\text{sum}}$

Σχήμα 4.6: Scatter plot QWCO και TIM

4.2.4 Συμπεράσματα

Όλες οι εσωτερικές μετρήσιμες ποσότητες που χρησιμοποιήθηκαν στη μελέτη, τόσο για τις μετρικές που αναφέρθηκαν παραπάνω όσο και για τις υπόλοιπες μετρικές που απαρτίζουν το επιλεγμένο τρίπτυχο, ήταν σε θετική συσχέτιση με τις εξωτερικές μετρικές. Ο βαθμός συσχέτισης κυμαινόταν από 54,94% έως 81,43%. Αυτό σημαίνει ότι η μέθοδος εσωτερικών μετρήσεων βρίσκεται σε αντιστοιχία με τις εξωτερικές μετρήσεις. Δηλαδή, οι εσωτερικές μετρήσεις συμπίπτουν με την άποψη των χρηστών. Παρόλα αυτά, δεν μετρήθηκε πολύ υψηλός βαθμός συσχέτισης, ώστε να μπορούμε να ισχυριστούμε πως ένα πρόγραμμα εσωτερικών μετρήσεων είναι ικανό να αντικαταστήσει τις εξωτερικές μετρήσεις.

Όσον αφορά τη μελέτη των scatter plots, παρατηρούμε ότι τα περισσότερα σημεία βρίσκονται κοντά στη διαγώνιο, δηλαδή τη γραμμή συσχέτισης. Πολύ λίγα σημεία βρίσκονται κάτω από τη διαγώνιο και ελάχιστα σημεία αρκετά κάτω από τη διαγώνιο. Αντίθετα, αρκετά σημεία βρίσκονται πάνω από τη διαγώνιο και μάλιστα μερικά από αυτά αρκετά πάνω από τη διαγώνιο. Τα σημεία που βρίσκονται κάτω από τη διαγώνιο αντιστοιχούν σε προγράμματα που είχαν αναλογικά πολύ υψηλότερη βαθμολογία στις εξωτερικές μετρικές από ό,τι στις εσωτερικές, ενώ τα σημεία που βρίσκονται πάνω από τη διαγώνιο αντιστοιχούν σε έργα που είχαν αναλογικά πολύ υψηλότερη βαθμολογία στις εσωτερικές μετρικές από ό,τι στις εξωτερικές.

Είναι φανερό λοιπόν πως προγράμματα με χαμηλή βαθμολογία στις εσωτερικές μετρικές έχουν αντίστοιχη χαμηλή βαθμολογία και στις εξωτερικές. Ακόμα και για τα λίγα προγράμματα όπου η εξωτερική τους εικόνα είναι καλύτερη από αυτή που παρουσιάζουν οι εσωτερικές μετρήσεις, η διαφορά απόκλισης είναι ελάχιστη. Αντιθέτως, προγράμματα με υψηλή βαθμολογία στις εσωτερικές μετρικές δεν παρουσιάζουν απαραίτητα αντίστοιχη υψηλή βαθμολογία στις εξωτερικές. Πράγματι, στα scatter plots παρατηρήθηκαν αρκετά σημεία που βρίσκονται πάνω από τη γραμμή συσχέτισης.

Αφού λοιπόν αποτυχία στα αποτελέσματα των εσωτερικών μετρήσεων οδηγεί σε αντίστοιχη αποτυχία στις εξωτερικές, είναι γενικώς αποδεκτό να θέτουμε ένα όριο στις εσωτερικές μετρικές, ώστε να εξασφαλίζουμε μία αποδεκτή εξωτερική ποιότητα του προγράμματος. Για παράδειγμα, ρουτίνες με πολύ χαμηλό επίπεδο προγράμματος ή πολύ υψηλή κυκλωματική πολυπλοκότητα μπορούν αυτομάτως να απορρίπτονται, επειδή θα οδηγήσουν σε προγράμματα χαμηλής εξωτερικής ποιότητας. Όμως, έχοντας απορρίψει όλες αυτές τις ρουτίνες, δεν υπάρχει αξιόπιστη μέθοδος να

επιβεβαιώσουμε ότι ρουτίνες με υψηλή βαθμολογία στις εσωτερικές μετρικές θα εμφανίσουν αντίστοιχα υψηλή βαθμολογία όσον αφορά την εξωτερική τους εικόνα. Με άλλα λόγια, οι εσωτερικές μετρικές αποτελούν μία πρώτη ένδειξη, και μάλιστα πολύ καλή, για την εξωτερική ποιότητα των προγραμμάτων λογισμικού, που μπορούν να φανερώσουν αρκετά νωρίς και με αυτόματο τρόπο την πιθανή αποτυχία στην εξωτερική ποιότητα ενός προγράμματος, αλλά δεν μπορούν να εξασφαλίσουν ανάλογα την επιτυχημένη εικόνα στις εξωτερικές μετρήσεις.

Συνοψίζοντας, μπορούμε να πούμε ότι αν και η απάντηση στο δεύτερο ερώτημα της ενότητας 4.1 είναι θετική, δεν μπορούμε να απαντήσουμε θετικά με καμία ακρίβεια για το πρώτο ερώτημα. Δηλαδή, η επιτυχία για ένα έργο σε ένα πρόγραμμα εξασφάλισης ποιότητας βασισμένο σε εσωτερικές μετρήσεις αποτελεί συνθήκη αναγκαία, αλλά όχι ικανή για την εξασφάλιση της επιτυχίας στις μετρήσεις της άποψης των πελατών για την ποιότητα του έργου.

Κεφάλαιο**5**

Συσχέτιση Εσωτερικών Μετρικών και Συντηρησιμότητας

Στο κεφάλαιο αυτό παρουσιάζεται η συσχέτιση εσωτερικών μετρικών λογισμικού με τη συντηρησιμότητα. Η συσχέτιση αυτή βασίζεται στις μεθόδους και τις τεχνικές που αναφέρθηκαν στα προηγούμενα κεφάλαια. Αναλύονται τα κριτήρια της συντηρησιμότητας, καθώς και οι διάφορες μέθοδοι μέσω των οποίων τα κριτήρια αυτά γίνονται αντιληπτά και ερμηνεύονται από τους προγραμματιστές. Παράλληλα, παρουσιάζονται δύο περιπτωσιολογικές μελέτες που διεξήχθησαν με την ταυτόχρονη εφαρμογή εσωτερικών μετρήσεων και μετρήσεων της γνώμης των προγραμματιστών για τη συντηρησιμότητα, ενώ επιπλέον γίνεται μελέτη του βαθμού συσχέτισης των αποτελεσμάτων που προέκυψαν από τις μετρήσεις αυτές [STA99b].

5.1 Η έννοια της συντηρησιμότητας

Ως συντηρησιμότητα λογισμικού ορίζεται ο βαθμός στον οποίο το λογισμικό μπορεί να κατανοηθεί, να διορθωθεί, να προσαρμοστεί και/ή να επεκταθεί [PRE97]. Ασφαλώς, σε έναν ιδανικό κόσμο, όπου οι αρχές και οι οδηγίες της εξασφάλισης ποιότητας λογισμικού ακολουθούνται κατά γράμμα, η συντηρησιμότητα δεν θα έχει τόσο μεγάλη βαρύτητα στη διαδικασία παραγωγής λογισμικού. Επιπλέον, η μείωση του κόστους της συντήρησης είναι ένας βασικός στόχος των προγραμμάτων ποιότητας λογισμικού. Παρόλα αυτά, στην πραγματικότητα οι αρχές της εξασφάλισης ποιότητας λογισμικού δεν εφαρμόζονται τόσο αυστηρά όσο χρειάζεται ή ακόμα και αν εφαρμόζονται, ο ρόλος της συντηρησιμότητας παραμένει σημαντικός.

Παρόλο που η συντηρησιμότητα είναι ένας από τους επιμέρους τομείς της ποιότητας που συνήθως ενδιαφέρουν έμμεσα τον πελάτη, τα δεδομένα που έχουν συλλεχθεί τις τελευταίες δεκαετίες δείχνουν ότι οι κατασκευαστές λογισμικού μπορεί να δαπανήσουν μέχρι και 70% του προϋπολογισμού ενός έργου στη συντήρηση του λογισμικού. Με άλλα λόγια, στον κύκλο ζωής λογισμικού η φάση της συντήρησης εμφανίζεται να παρουσιάζει το μεγαλύτερο κόστος από τις άλλες. Επιπλέον, είναι αληθές ότι οι προδιαγραφές κατά την κατασκευή του λογισμικού όντως αλλάζουν, αλλά το μέγεθος της επίδρασης των αλλαγών αυτών κυμαίνεται ανάλογα με τη χρονική στιγμή που παρουσιάζονται. Μάλιστα, μία αλλαγή που ζητείται κατά τη

φάση της συντήρησης μπορεί να προκαλέσει ένα κόστος περισσότερο από 10 φορές μεγαλύτερο από το κόστος που θα προκαλέσει μία αντίστοιχη αλλαγή κατά της φάση της σχεδίασης (βλ. Σχήμα 1.2). Δηλαδή, το κόστος για να συντηρηθεί μία ρουτίνα μπορεί να φτάσει πάνω από 10 φορές μεγαλύτερο από το κόστος για να κατασκευαστεί αρχικά η ρουτίνα αυτή.

Ως αποτέλεσμα, μία εταιρεία παραγωγής λογισμικού πρέπει να επινοήσει διάφορους τρόπους μέτρησης της ευκολίας της συντήρησης, όχι μόνο για να μειώσει το κόστος της ίδιας της συντήρησης, αλλά και για να εξακριβώσει αν συμφέρει οικονομικά ένα λογισμικό να συντηρηθεί ή όχι. Για τον παραπάνω λόγο, έχει προταθεί ένας μεγάλος αριθμός από μετρικές λογισμικού που σχετίζονται με τα κριτήρια της συντηρησιμότητας. Ο βασικός σκοπός αυτών των μετρικών είναι να καθορίσουν το βαθμό της συντηρησιμότητας ενός προϊόντος λογισμικού.

5.1.1 Οι τύποι της συντήρησης λογισμικού

Όπως ήδη αναφέρθηκε, το κόστος και η προσπάθεια που καταβάλλεται κατά τη διάρκεια της φάσης της συντήρησης του λογισμικού είναι μεγαλύτερη από κάθε άλλη φάση του κύκλου ζωής λογισμικού. Παρόλο που θεωρούνταν μέχρι πρόσφατα ως μία παραμελημένη φάση στην διαδικασία παραγωγής λογισμικού, η συντηρησιμότητα αποτελεί ένα στόχο κλειδί που μπορεί να κατευθύνει τα στάδια αυτής της παραγωγής. Στο λογισμικό διακρίνονται τέσσερις διαφορετικοί τύποι συντήρησης:

- Η **διορθωτική συντήρηση** που έχει σκοπό τη διάγνωση και διόρθωση λαθών που ανακαλύφθηκαν αφότου το λογισμικό άρχισε να χρησιμοποιείται.
- Η **προσαρμοστική συντήρηση** που έχει σκοπό την προσαρμογή σε ένα νέο μεταβαλλόμενο περιβάλλον όπου εφαρμόζεται το υπάρχον λογισμικό, όπως για παράδειγμα ανάγκη για νέο υλικό, νέο λειτουργικό σύστημα, περιφερειακό εξοπλισμό, μεταφορά του λογισμικού σε διαφορετικό σύστημα κ.ά.
- Η **τελειοποιητική συντήρηση** που έχει σκοπό τον εμπλουτισμό του λογισμικού με νέες δυνατότητες κατόπιν πρότασης των χρηστών του ή πρωτοβουλίας της ίδιας της ομάδας που κατασκεύασε το λογισμικό.
- Η **προληπτική συντήρηση** που έχει σκοπό τη βελτίωση της απόδοσης του λογισμικού, της συντηρησιμότητάς του ή της αξιοπιστίας του, ώστε να παρέχεται μία βάση για μελλοντική επέκτασή του.

Όπως αναφέρθηκε στην ενότητα 1.4, σύμφωνα με το μοντέλο του McCall, ο παράγοντας συντηρησιμότητα αποτελείται από τα κριτήρια συνοχή, απλότητα, περιεκτικότητα, αυτοπεριγραφικότητα και τμηματοποίηση [MCC77]. Σύμφωνα με το πρότυπο IEEE για τις μετρικές ποιότητας, η συντηρησιμότητα απαρτίζεται από τη διορθωσιμότητα, την ελεγχσιμότητα και την επεκτασιμότητα [IEEE89]. Τέλος, σύμφωνα με το πρότυπο ISO9126, η συντηρησιμότητα αποτελείται από τα υπο-χαρακτηριστικά αναλυσιμότητα, τροποποιησιμότητα, σταθερότητα και ελεγχσιμότητα [ISO01].

5.1.2 Απόψεις χειρισμού της διαδικασίας συντήρησης

Καθώς όλο και περισσότερα προγράμματα λογισμικού αναπτύσσονται, το μέγεθος της προσπάθειας και το πλήθος των πόρων που απαιτούνται για τη συντήρησή τους διαρκώς αυξάνονται. Η συντηρησιμότητα του λογισμικού επηρεάζεται από αρκετούς παράγοντες, όπως η διαθεσιμότητα ικανού προσωπικού για να αναλάβει τη διαδικασία αυτή, η ευκολία χειρισμού του συστήματος, η γλώσσα

προγραμματισμού που επιλέχτηκε, η μεθοδολογία που ακολουθήθηκε κατά τις προηγούμενες φάσεις της ανάπτυξης λογισμικού κ.ά. Λάθη και απροσεξίες που γίνονται κατά τη σχεδίαση, την υλοποίηση και τον έλεγχο παρουσιάζουν ένα προφανές αρνητικό αντίκτυπο στη δυνατότητα να συντηρηθεί το απορρέον λογισμικό [KOP79]. Επιπλέον, μία εταιρεία παραγωγής λογισμικού μπορεί να φτάσει στο σημείο να είναι υποχρεωμένη να μείνει καθηλωμένη στο να συντηρεί έργα που έχει αναπτύξει στο παρελθόν, επειδή όλοι οι πόροι της θα έχουν αφοσιωθεί στη συντήρησή τους. Ως αποτέλεσμα, θα είναι αδύνατο να αναλάβει την ανάπτυξη νέων έργων.

Η γνώμη των προγραμματιστών σχετικά με το επίπεδο της συντηρησιμότητας συνήθως ποικίλει. Σε γενικές γραμμές πάντως, κατά την άποψή τους ένα πρόγραμμα με υψηλό επίπεδο συντηρησιμότητας πρέπει να αποτελείται από τμήματα με υψηλή συνοχή και χαλαρή σύνδεση, που θα είναι αναγνώσιμα, απλά, σωστά δομημένα, θα περιέχουν κώδικα με επαρκή σχολιασμό, θα ακολουθούν αυστηρά ένα καθορισμένο στυλ προγραμματισμού και οι μεταβλητές τους θα έχουν μία εύκολα κατανοήσιμη ορολογία. Επιπλέον, οι ρουτίνες που θα αναπτυχθούν πρέπει να έχουν ένα λογικό μέγεθος (κατά προτίμηση λιγότερο από 80 γραμμές κώδικα), καθώς και περιορισμένο αριθμό εισόδων και εξόδων. Πρέπει να ακολουθείται μία ξεκάθαρη λογική και να χρησιμοποιούνται απλές και κατανοήσιμες εκφράσεις. Τέλος, είναι προτιμότερο το τμήμα δήλωσης και το τμήμα υλοποίησης της κάθε ρουτίνας να είναι αυστηρώς διαχωρισμένα [KER99].

Σύμφωνα με την άποψη των υπευθύνων προγραμμάτων για τη συντηρησιμότητα, τα προγράμματα που αναπτύσσονται πρέπει να αποτελούνται από ανεξάρτητα τμήματα, τα οποία πρέπει να είναι τεκμηριωμένα και να μπορούν να ελεγχθούν από μόνα τους. Πρέπει να ακολουθείται ένα σαφές πλαίσιο για τον καθορισμό των διαδοχικών εκδόσεων αυτών των τμημάτων σε ένα συγκεκριμένο περιβάλλον. Οι υπεύθυνοι έργων στοχεύουν πάντοτε στον περιορισμό της προσπάθειας που θα δαπανηθεί κατά τη διαδικασία της συντήρησης. Επιπλέον, επιθυμούν την επίτευξη ενός υψηλού βαθμού επαναχρησιμοποίησης στα προγράμματα που αναπτύσσονται, έτσι ώστε τα τμήματα λογισμικού που τα αποτελούν να μπορούν με μεγαλύτερη ευκολία να χρησιμοποιηθούν ως έχουν και σε επόμενα προγράμματα.

Η συντηρησιμότητα είναι ένας παράγοντας της ποιότητας που συνήθως ενδιαφέρει έμμεσα τον πελάτη. Ωστόσο στην πραγματικότητα, ένας μικρός αριθμός πελατών προσθέτει στις απαιτήσεις του προς τους προγραμματιστές κατευθυντήριες οδηγίες σχετικά με τη συντηρησιμότητα, εκτός φυσικά από την περίπτωση που οι κατασκευαστές του λογισμικού πρόκειται να αναλάβουν οι ίδιοι τη συντήρησή του. Σύμφωνα με τον Ince, μία από τις ενδείξεις που έχει παραδόξως ο κατασκευαστής λογισμικού ότι έχει αναπτυχθεί ένα ποιοτικό σύστημα είναι να συλλέξει έναν αριθμό από αιτήματα από τους πελάτες για τροποποιήσεις που προκύπτουν από τις αλλαγές στις απαιτήσεις τους [INC94]. Και όπως έχει παρατηρηθεί, οι πελάτες μπορεί να αναθεωρούν την άποψή τους για την ποιότητα ενός προϊόντος λογισμικού πολύ συχνά [STA99a]. Σήμερα, εξαιτίας της μεγάλης ζήτησης που έχουν οι πελάτες από τα επιτυχημένα συστήματα λογισμικού και εξαιτίας αλλαγών σε διάφορους εξωτερικούς παράγοντες, ένα μεγάλο ποσοστό των τροποποιήσεων μπορεί να αποδοθεί στις αλλαγές των αρχικών απαιτήσεων.

5.1.3 Προσεγγίσεις μέτρησης της συντηρησιμότητας

Υπάρχουν δύο γενικές προσεγγίσεις για τη μέτρηση της συντηρησιμότητας που εκφράζουν την εξωτερική και την εσωτερική όψη αυτού του χαρακτηριστικού. Η συντηρησιμότητα είναι ένας εξωτερικός παράγοντας της ποιότητας, καθώς εξαρτάται όχι μόνο από το ίδιο το προϊόν λογισμικού, αλλά και από την ομάδα συντήρησης του λογισμικού αυτού. Κατά την εξωτερική προσέγγιση μέτρησης της συντηρησιμότητας, που είναι και περισσότερο άμεση, καθορίζονται μετρήσιμες ποσότητες της διαδικασίας της συντήρησης και μετά συλλέγονται οι απόψεις των προγραμματιστών που συμμετέχουν σε αυτή τη διαδικασία. Δυστυχώς, η εξωτερική προσέγγιση απαιτεί πολύ χρόνο και βασίζεται στη διεξαγωγή μίας έρευνας που μπορεί να έχει μεγάλο κόστος. Επιπρόσθετα, είναι πιθανόν σε έρευνες αυτού του είδους να χρειάζεται βελτίωση της ορθότητας και της ερμηνείας των δεδομένων εισόδου και των αποτελεσμάτων που προκύπτουν. Εναλλακτικά, κατά την εσωτερική προσέγγιση μέτρησης της συντηρησιμότητας γίνεται χρήση εσωτερικών μετρικών, ελπίζοντας ότι μπορούν να προβλέψουν την άποψη των προγραμματιστών για τη συντηρησιμότητα ενός προϊόντος λογισμικού. Επιπλέον, στην εσωτερική προσέγγιση τα αποτελέσματα των μετρήσεων μπορούν να συλλεχτούν σαφώς ταχύτερα και ευκολότερα.

Κατά τη διάρκεια της φάσης της συντήρησης η δομή των συστημάτων λογισμικού συνήθως υποβαθμίζεται. Τα μέλη της ομάδας συντήρησης κάνουν συνήθως τυχαίες προσθήκες στον κώδικα με αποτέλεσμα να μειώνεται η ποιότητα της δομής των συστημάτων. Επομένως, βαθμιαία το σύστημα γίνεται όλο και πιο δύσκολο να συντηρηθεί. Σε αυτή την περίπτωση, οι μετρικές λογισμικού μπορούν να χρησιμοποιηθούν για να ελεγχθεί σε κάποια όρια η πιθανή υποβάθμιση του συστήματος, έτσι ώστε να διατηρηθεί η ποιότητα της διαδικασίας της συντήρησης σε υψηλά επίπεδα.

5.2 Χρήση μετρικών λογισμικού για τη βελτίωση της συντηρησιμότητας

Η συντηρησιμότητα του λογισμικού αποτελεί έναν παράγοντα της ποιότητας που είναι δύσκολο να υπολογιστεί. Μπορεί όμως να μετρηθεί έμμεσα, λαμβάνοντας υπόψη τα αποτελέσματα από εσωτερικές μετρικές λογισμικού. Όπως υποστηρίζεται εξάλλου, η πολυπλοκότητα και η δομή ενός προγράμματος έχουν μεγάλη συσχέτιση με τη συντηρησιμότητα του παραγόμενου λογισμικού [KAF87], [ROM87]. Αυτό που πρέπει όμως να καθοριστεί είναι σε ποιο βαθμό και σε ποιες περιπτώσεις μπορούμε να βασιστούμε στις μετρικές λογισμικού για να προσδιορίσουμε τη συντηρησιμότητα ενός προϊόντος λογισμικού.

Μελετώντας τις διάφορες απόψεις για τις ενέργειες που πρέπει να ληφθούν κατά την παραγωγή λογισμικού ώστε να επιτευχθεί ένας υψηλός βαθμός συντηρησιμότητας, γίνεται εύκολα αντιληπτό ότι οι μετρικές λογισμικού μπορούν να εκτιμήσουν μόνο την άποψη των προγραμματιστών για τη συντηρησιμότητα. Όπως αναφέρθηκε και στο προηγούμενο κεφάλαιο, παρόλο που οι εσωτερικές μετρικές λογισμικού εμφανίζουν μεγάλη συσχέτιση με τη γνώμη των πελατών για την ποιότητα, τα υψηλά αποτελέσματα στις εσωτερικές μετρήσεις δεν μπορούν να εξασφαλίσουν εκ των προτέρων αντίστοιχα υψηλά αποτελέσματα όσον αφορά τη γνώμη των τελικών χρηστών. Οι εσωτερικές μετρικές μπορούν με σιγουριά να διακρίνουν μόνο πιθανές περιπτώσεις χαμηλής ποιότητας. Στην περίπτωση όμως που συγκρίνουμε τα αποτελέσματα των μετρικών λογισμικού με τη γνώμη των

προγραμματιστών για τη συντηρησιμότητα μπορούμε να συμπεράνουμε ότι υπάρχει μία μεγάλη συσχέτιση μεταξύ τους. Με άλλα λόγια, στις περισσότερες περιπτώσεις οι μετρικές λογισμικού μπορούν όχι μόνο να προσδιορίσουν τμήματα με χαμηλό επίπεδο συντηρησιμότητας, αλλά και να προβλέψουν το υψηλό επίπεδο συντηρησιμότητας ενός προϊόντος χωρίς την ανάγκη διεξαγωγής έρευνας για άμεση μέτρηση της γνώμης των προγραμματιστών για αυτόν τον παράγοντα της ποιότητας.

Η επιλογή κατάλληλων εσωτερικών μετρικών για τη μέτρηση της συντηρησιμότητας ενός συγκεκριμένου προϊόντος εξαρτάται από τη φύση του προϊόντος και τη γλώσσα προγραμματισμού στην οποία έχει υλοποιηθεί. Παραδοσιακές και ευρέως χρησιμοποιούμενες μετρικές, όπως οι μετρικές του Halstead [HAL75], η κυκλωματική πολυπλοκότητα του McCabe [MCB76], οι μετρικές του Tsai [TSA86], οι γραμμές κώδικα, οι γραμμές σχολίων ανά γραμμές κώδικα, ο αριθμός εισόδων και εξόδων κ.ά., μπορούν πάντοτε να εφαρμόζονται για να εκτιμηθεί το επίπεδο της συντηρησιμότητας ενός προγράμματος. Επιπρόσθετα, για την εκτίμηση της συντηρησιμότητας των προγραμμάτων που έχουν υλοποιηθεί σε ένα συγκεκριμένο τύπο γλωσσών προγραμματισμού μπορούν να χρησιμοποιηθούν πέρα από τις παραπάνω μετρικές και άλλες μετρικές που εφαρμόζονται μόνο σε προγράμματα αυτού του συγκεκριμένου τύπου. Έτσι για παράδειγμα, στις περιπτώσεις προγραμμάτων που υλοποιήθηκαν σε μία αντικειμενοστραφή γλώσσα προγραμματισμού οι μετρικές που μπορούν επίσης να εφαρμοστούν είναι το άθροισμα των βαρών όλων των μεθόδων μίας κλάσης, το επίπεδο συνοχής ανάμεσα στις μεθόδους, ο αριθμός των κλάσεων με τις οποίες συνδέεται μία κλάση, ο αριθμός των προγόνων μίας κλάσης, ο συνολικός αριθμός των παιδιών μίας κλάσης κ.ά. [HUD94].

Θα πρέπει να σημειωθεί επίσης ότι μερικές από τις μετρικές που έχουν προταθεί εστιάζουν ακριβώς στο ζήτημα που μας ενδιαφέρει. Για παράδειγμα, υπολογίζουν άμεσα το μέγεθος, το πλήθος των εμφωλιασμένων επιπέδων, τον αριθμό εισόδων και εξόδων μίας ρουτίνας κ.ά. που είναι σημαντικά κριτήρια στη διαμόρφωση της άποψης για τη συντηρησιμότητα της ρουτίνας. Επιπλέον, οι μετρικές λογισμικού αποσκοπούν να είναι αποδεκτές ενδείξεις όχι μόνο στην εκτίμηση της δομής, της συνοχής και της σύνδεσης ανάμεσα στα διάφορα τμήματα ενός προγράμματος, αλλά και στην εκτίμηση της αναγνωσιμότητας, της σαφήνειας, της επάρκειας σχολίων και της απλότητας του κώδικα αυτών των τμημάτων.

Η χρήση εσωτερικών μετρικών λογισμικού αποκτά νόημα μόνο όταν εφαρμόζονται σε μεγάλα τμήματα κώδικα. Για να βγουν σαφή και χρήσιμα συμπεράσματα οι μετρήσεις δεν θα πρέπει να περιορίζονται σε μικρά τμήματα ενός προγράμματος λογισμικού. Παρόλα αυτά, χρησιμοποιώντας μία συλλογή από μετρικές οι προγραμματιστές μπορούν να εντοπίσουν τμήματα ή ακόμα και ρουτίνες με χαμηλό επίπεδο συντηρησιμότητας. Η χρήση των μετρικών βοηθάει τους προγραμματιστές να επιθεωρούν τον κώδικά τους και να κάνουν τις απαραίτητες διορθώσεις και βελτιώσεις κατά τη διάρκεια της φάσης της υλοποίησης. Αν και αληθεύει ότι η εφαρμογή εσωτερικών μετρήσεων μπορεί να καθυστερήσει τη φάση της υλοποίησης, η καθυστέρηση αυτή είναι σαφώς προτιμότερη από την αντιμετώπιση ενός μεγάλου μεγέθους προσπάθειας και πόρων που πιθανόν να απαιτήσει η φάση της συντήρησης. Με άλλα λόγια, η επιτυχία υψηλών αποτελεσμάτων στις εσωτερικές μετρικές λογισμικού πρέπει να αποτελεί μία βασική απαίτηση για την επιτυχή ολοκλήρωση της υλοποίησης ενός προγράμματος.

Τα αποδεκτά όρια που πρέπει να καθοριστούν στα αποτελέσματα της κάθε μετρικής που θα χρησιμοποιηθεί στην παραπάνω συλλογή για τη μέτρηση της συντηρησιμότητας εξαρτώνται από τη φύση του προϊόντος που πρόκειται να παραχθεί, τη γλώσσα προγραμματισμού που θα επιλεγεί, το προφίλ και τους αντικειμενικούς στόχους της εταιρείας λογισμικού, την εμπειρία των προγραμματιστών κ.ά. Για παράδειγμα, στην υλοποίηση ενός αναλυτή (parser) ή ενός γραμματικού ελεγκτή είναι προφανές ότι κάποια τμήματα στο πρόγραμμα θα έχουν πολύ μεγάλη κυκλωματική πολυπλοκότητα και μεγάλο αριθμό εμφωλιασμένων επιπέδων. Στην περίπτωση αυτή, το παραπάνω γεγονός είναι αποδεκτό. Σε άλλες περιπτώσεις προγραμμάτων όμως, αυτού του είδους τα αποτελέσματα θα χαρακτηρίζονταν απαγορευτικά και θα ήταν προτιμότερο να μη γίνει η μετάβαση στην επόμενη φάση του κύκλου ζωής λογισμικού, αλλά να υποχρεωθούν οι προγραμματιστές να υλοποιήσουν ξανά τα κακής εσωτερικής ποιότητας τμήματα.

Θα πρέπει να σημειωθεί ότι τα διαφορετικά κριτήρια που συνθέτουν τη συντηρησιμότητα μπορεί να έρχονται σε σύγκρουση σε μερικές περιπτώσεις. Η απλότητα και η περιεκτικότητα μπορεί να έρχονται σε αντίθεση με την αναλυσιμότητα, την τροποποιησιμότητα και την επεκτασιμότητα. Για παράδειγμα, οι ρουτίνες A και B του Σχήματος 5.1 έχουν γραφτεί σε Object Pascal για τον ίδιο ακριβώς σκοπό.

Ρουτίνα A

```

Procedure TChoiceForm.ListView1Change(Sender: TObject; Item:
TListItem; Change: TItemChange);
Begin
  If ListViewChoices.Selected <> nil then
  begin
    DelBtn.Enabled:=True;
    MoveUp.Enabled:=True;
    MoveDown.Enabled:=True;
  End
  Else
  Begin
    DelBtn.Enabled:=False;
    MoveUp.Enabled:=False;
    MoveDown.Enabled:=False;
  End;
end;

```

Ρουτίνα B

```

Procedure TChoiceForm.ListView1Change(Sender: TObject; Item:
TListItem; Change: TItemChange);
var HasSelection : boolean;
begin
  if ListViewChoices.Selected <> nil then HasSelection:=true
  else HasSelection:=false;

  DelBtn.Enabled:=HasSelection;
  MoveUp.Enabled:=HasSelection;
  MoveDown.Enabled:=HasSelection;
end;

```

Σχήμα 5.1: Παράδειγμα με παρόμοιες ρουτίνες

Σύμφωνα με τις μετρικές του Halstead, η ρουτίνα B είναι καλύτερη από τη ρουτίνα A, επειδή έχει υψηλότερο επίπεδο προγράμματος ($L_A=0.2$ και $L_B=0.33$) και

υψηλότερο επίπεδο γλώσσας ($\lambda_A=4.45$ και $\lambda_B=8.98$). Παράλληλα, η ρουτίνα Β έχει λιγότερες γραμμές κώδικα από τη ρουτίνα Α. Αυτό φυσικά είναι ένα πολύ μικρό κομμάτι κώδικα για να μπορούμε να βγάλουμε αποδεκτά συμπεράσματα. Παρόλα αυτά, είναι προφανές ότι αν και η ρουτίνα Β είναι απλούστερη και πιο περιεκτική, η ρουτίνα Α είναι ευκολότερο να αναλυθεί, να τροποποιηθεί ή να επεκταθεί. Η τελική επιλογή πρέπει να γίνει σύμφωνα με το κριτήριο της συντηρησιμότητας όπου θα δοθεί μεγαλύτερη βαρύτητα.

Η εταιρεία παραγωγής λογισμικού που θέλει να εφαρμόσει ένα περιβάλλον μετρικών λογισμικού για να ελέγξει το επίπεδο της συντηρησιμότητας των προϊόντων της πρέπει πρώτα να επιλέξει μία συλλογή από κατάλληλες μετρικές για το κάθε προϊόν. Στη συνέχεια, πρέπει να καθορίσει τα όρια για τα αποτελέσματα της κάθε μετρικής που θα χαρακτηρίζουν τον κώδικα α) ως αποδεκτό, β) ως αποδεκτό αν γίνουν κάποιες διορθώσεις ή γ) ως απορριπτέο. Μετά, αφού διεξαχθούν οι εσωτερικές μετρήσεις, αν κάποιο τμήμα ή μέρος του προγράμματος αποτύχει σύμφωνα με τα προκαθορισμένα όρια στα αποτελέσματα των μετρικών, θα πρέπει να ελεγχθεί από τους ίδιους τους προγραμματιστές για να αποφασιστεί το επόμενο βήμα. Αυτά τα στάδια προλαμβάνουν την ομάδα των προγραμματιστών από σοβαρά προβλήματα και ανάγκη μεγάλης προσπάθειας κατά τη διάρκεια της συντήρησης. Επιπλέον, αν τα αποτελέσματα των μετρικών εμφανίζουν μεγάλη απόκλιση από τα επιθυμητά επίπεδα στην πλειοψηφία των τμημάτων, είναι προτιμότερο το προϊόν λογισμικού να επανασχεδιαστεί. Για το λόγο αυτό οι υπεύθυνοι των έργων θα πρέπει να είναι ιδιαίτερος προσεκτικοί όταν καθορίζουν τους περιορισμούς και τα όρια για την κάθε μετρική που θα χρησιμοποιηθεί.

5.3 Περιπτώσιολογικές μελέτες

Όπως έχει ήδη αναφερθεί, ένα βασικό ζήτημα που πρέπει να καθοριστεί είναι σε ποιο βαθμό και σε ποιες περιπτώσεις μπορούμε να βασιστούμε στις μετρικές λογισμικού για να προσδιορίσουμε τη συντηρησιμότητα ενός προϊόντος λογισμικού. Για να εξεταστεί λοιπόν η συσχέτιση ανάμεσα στις μετρικές λογισμικού και τη συντηρησιμότητα διεξήχθησαν ανάλογες περιπτώσιολογικές μελέτες στις οποίες συμμετείχαν προγραμματιστές που είτε συμμετείχαν στη διαδικασία της συντήρησης είτε επρόκειτο υποθετικά στο μέλλον να συντηρήσουν τον κώδικα που τους δόθηκε. Στη συνέχεια παρουσιάζονται τα αποτελέσματα από δύο περιπτώσιολογικές μελέτες που διεξήχθησαν με την ταυτόχρονη εφαρμογή εσωτερικών μετρήσεων και μετρήσεων της γνώμης των παραπάνω προγραμματιστών για τη συντηρησιμότητα.

5.3.1 Πρώτη περιπτώσιολογική μελέτη

Κατά την πρώτη περιπτώσιολογική μελέτη μετρήθηκαν 5 μεγάλα έργα, καθένα από τα οποία αποτελούνταν από 10 μέχρι 20 τμήματα και έγινε μελέτη του επιπέδου συντηρησιμότητάς τους και με τις δύο προσεγγίσεις: άμεσα, όπου συλλέχθηκαν οι εκτιμήσεις από 50 προγραμματιστές, και έμμεσα, όπου έγινε η χρήση εσωτερικών μετρικών. Κάθε τμήμα του κάθε έργου μετρήθηκε και με τις δύο προσεγγίσεις ξεχωριστά. Για να μετρηθεί η γνώμη των προγραμματιστών για τη συντηρησιμότητα των έργων (εξωτερική μέτρηση) έγινε η χρήση ενός κατάλληλου ερωτηματολογίου, το οποίο αποτελούνταν από ερωτήσεις (πολλαπλής επιλογής) για κάθε έναν παράγοντα της συντηρησιμότητας. Επομένως, οι προγραμματιστές εκτίμησαν τα διάφορα τμήματα των έργων σχετικά με τη συνοχή, την απλότητα, την περιεκτικότητα, την επεκτασιμότητα, τη διορθωσιμότητά τους κ.ά. Οι εσωτερικές μετρήσεις για την

περιπτωσιολογική μελέτη διεξήχθησαν αυτόματα χρησιμοποιώντας το περιβάλλον ATHENA [TSL91]. Επομένως, τα αποτελέσματα των εσωτερικών μετρήσεων συλλέχθηκαν άμεσα, γρήγορα και χωρίς μεγάλο κόστος, όπως θα πρέπει άλλωστε να συμβαίνει σε κάθε εταιρεία που διεξάγει ανάλογες εσωτερικές μετρήσεις.

Από τα αποτελέσματα των δύο ειδών μετρήσεων έγινε φανερό ότι οι εσωτερικές μετρικές και η άποψη των προγραμματιστών για τη συντηρησιμότητα βρίσκονται σε ισχυρή συσχέτιση μεταξύ τους. Αναλυτικότερα, τα τμήματα με υψηλά αποτελέσματα στις εσωτερικές μετρήσεις εκτιμήθηκαν αντίστοιχα υψηλά και στις εξωτερικές μετρήσεις. Επιπλέον, η άποψη των προγραμματιστών για τη συντηρησιμότητα των τμημάτων που απέτυχαν στις εσωτερικές μετρικές ήταν αντίστοιχα χαμηλή. Παραδείγματα από τμήματα της κάθε περίπτωσης παρουσιάζονται στα Σχήματα 5.2 (τμήμα με υψηλά αποτελέσματα) και 5.3 (τμήμα με χαμηλά αποτελέσματα), όπου για κάθε ρουτίνα του κάθε τμήματος αναφέρονται τα αποτελέσματα στις γραμμές κώδικα (LOC), στο επίπεδο προγράμματος (L), στο επίπεδο γλώσσας (λ) και στην κυκλωματική πολυπλοκότητα (Vg). Στην πραγματικότητα, ένας μεγάλος αριθμός από διαφορετικές μετρικές χρησιμοποιήθηκε στη συγκεκριμένη μελέτη. Εξαιτίας όμως της διαφορετικής φύσης των 5 προς εξέταση έργων, ορισμένες από τις επιλεγμένες μετρικές δεν μπορούσαν να εφαρμοστούν σε όλα τα έργα. Τα ιστογράμματα δίπλα σε κάθε τμήμα παρουσιάζουν τη γνώμη των προγραμματιστών για τη συντηρησιμότητά τους, σύμφωνα με τις απαντήσεις που έδωσαν στα ερωτηματολόγια. Στον οριζόντιο άξονα απεικονίζεται το αποτέλεσμα της άποψής τους, ενώ στον κατακόρυφο άξονα αναγράφεται ο αριθμός των προγραμματιστών.

Σχήμα 5.2: Παράδειγμα τμήματος με υψηλά αποτελέσματα στις μετρήσεις

Σχήμα 5.3: Παράδειγμα τμήματος με χαμηλά αποτελέσματα στις μετρήσεις

Στη συγκεκριμένη μελέτη πάντως παρατηρήθηκαν μερικά τμήματα που ενώ εμφάνισαν χαμηλά αποτελέσματα στις εσωτερικές μετρικές, παρουσίασαν μεγάλη απόκλιση στις μετρήσεις της άποψης των προγραμματιστών για τη συντηρησιμότητά τους. Με άλλα λόγια, τα χαμηλά αποτελέσματα στις εσωτερικές μετρήσεις δεν συνεπάγονται απαραίτητα αντίστοιχα χαμηλά αποτελέσματα στις εξωτερικές μετρήσεις. Η απόκλιση αυτή παρατηρήθηκε όχι μόνο λόγω της εμπειρίας των προγραμματιστών στη γλώσσα προγραμματισμού όπου υλοποιήθηκαν τα συγκεκριμένα τμήματα ή λόγω των προσωπικών ικανοτήτων των προγραμματιστών, αλλά κυρίως εξαιτίας της φύσης των έργων στα οποία τα τμήματα αυτά ανήκουν, καθώς και των λειτουργιών που υλοποιούνται σε αυτά. Στο Σχήμα 5.4 παρουσιάζονται τα αποτελέσματα σε ένα τμήμα αυτού του τύπου. Η παρατήρηση αυτή έρχεται σε αντίθεση με τη συσχέτιση ανάμεσα στις μετρικές λογισμικού και της άποψης των πελατών για την ποιότητα ενός προγράμματος, η οποία παρουσιάστηκε στο προηγούμενο κεφάλαιο. Στην περίπτωση των πελατών οι μετρικές μπορούν να εντοπίσουν πιθανές περιπτώσεις χαμηλής ποιότητας, ενώ αντίθετα τα ικανοποιητικά αποτελέσματα στις μετρικές δεν μπορούν να εγγυηθούν πάντοτε την επιτυχία στην εκπλήρωση των αιτημάτων των πελατών για ποιότητα.

Σχήμα 5.4: Παράδειγμα τμήματος με χαμηλές εσωτερικές και υψηλές εξωτερικές μετρήσεις

5.3.2 Δεύτερη περιπτωσιολογική μελέτη

Στην περιπτωσιολογική μελέτη που παρουσιάζεται εδώ ακολουθήθηκε παρόμοια διαδικασία με αυτή της προηγούμενης ενότητας. Χρησιμοποιήθηκαν δεδομένα από $N=29$ τμήματα από εφαρμογές λογισμικού που αναπτύχθηκαν στα πλαίσια δύο μεγάλων έργων. Καθένα από αυτά τα N_i ($i \in [1, 29]$) τμήματα μπορούσε να αντιμετωπιστεί ως μία αυτόνομη εφαρμογή, επομένως μπορούσε να μετρηθεί και εκτιμηθεί ξεχωριστά από τα άλλα τμήματα του κάθε έργου. Οι εφαρμογές αυτές είχαν υλοποιηθεί σε C++ ή Object Pascal, ενώ κατά τη διάρκεια του κύκλου ζωής λογισμικού έπρεπε να συντηρηθούν ή ακόμα και να τροποποιηθούν ώστε να επαναχρησιμοποιηθούν σε μελλοντικά έργα. Τα αποτελέσματα που παρουσιάζονται σε αυτή τη μελέτη βασίζονται σε εξωτερικές μετρήσεις που έγιναν από P_i προγραμματιστές για κάθε τμήμα N_i . Ο αριθμός P_i κάθε τμήματος N_i κυμαινόταν από 28 μέχρι 54. Οι απόψεις των προγραμματιστών συνδυάστηκαν σε ένα μετρήσιμο αποτέλεσμα για το κάθε τμήμα, χρησιμοποιώντας τη μέθοδο μέτρησης της γνώμης του χρήστη με βαρύτητα προσόντων και ερωτήσεις ασφαλείας που παρουσιάστηκε στην ενότητα 3.4.3. Επομένως, βάσει αυτής της μεθόδου, προέκυπτε ένας απλός αριθμός EM_i στον οποίο συνδυάζονταν όλες οι εξωτερικές μετρήσεις για το τμήμα N_i . Είναι προφανές από τη σχέση 3.4.3 ότι $EM_i \in [0, 1]$. Σε αυτή την περιπτωσιολογική

μελέτη παρατηρήθηκαν 4 ακραίες τιμές (outliers) από τα 29 τμήματα και έτσι δεν λήφθηκαν υπόψη, αφού αποτελούνταν είτε από ρουτίνες δήλωσης μεταβλητών είτε από πολύ μικρές ρουτίνες που δεν περιείχαν καθόλου εντολές διακλάδωσης ή ελέγχου ροής δεδομένων, αλλά απλά καλούσαν άλλες ρουτίνες.

Σχήμα 5.5: Ιστόγραμμα εξωτερικών μετρήσεων

Τα αποτελέσματα των εξωτερικών μετρήσεων ακολούθησαν μία κανονική κατανομή (*Επίπεδο σημαντικότητας του ελέγχου κανονικότητας Shapiro-Wilk* = 0,439 >> 0,05, *Ασυμμετρία* = -0,516, *Κυρτότητα* = 0,488). Στο Σχήμα 5.5 παρουσιάζεται το ιστόγραμμα, ενώ στο Σχήμα 5.6 παρουσιάζεται το διάγραμμα κανονικής πιθανότητας των αποτελεσμάτων αυτών (*Μέση τιμή* = 0,638, *Τυπική απόκλιση* = 0,117).

Παρόμοια με τις εξωτερικές μετρήσεις, τα αποτελέσματα στις εσωτερικές μετρικές αναλύθηκαν και συγκρίθηκαν με αυτά των εξωτερικών μετρήσεων. Στην περιπτωσιολογική μελέτη αυτή έγινε ξανά χρήση της συνδυαστικής μετρικής TIM (Total Internal Measurements) της σχέσης 4.2.4. Πριν από κάθε σύγκριση τα αποτελέσματα σε κάθε μία εσωτερική μετρική ελέγχθηκαν για πιθανές ακραίες τιμές, καθώς και για κανονική κατανομή. Έτσι στο Σχήμα 5.7 παρουσιάζεται το ιστόγραμμα και στο Σχήμα 5.8 το διάγραμμα κανονικής πιθανότητας για το επίπεδο γλώσσας λ.

Σχήμα 5.6: Διάγραμμα κανονικής πιθανότητας των εξωτερικών μετρήσεων

Σχήμα 5.7: Ιστόγραμμα για το επίπεδο γλώσσας λ

Σχήμα 5.8: Διάγραμμα κανονικής πιθανότητας για το επίπεδο γλώσσας λ

Τα αποτελέσματα στο επίπεδο γλώσσας λ για τα N_i τμήματα ακολούθησαν κανονική κατανομή (*Επίπεδο σημαντικότητας του ελέγχου κανονικότητας Shapiro-Wilk* = 0,526 >> 0,05, *Ασυμμετρία* = 0,526, *Κυρτότητα* = 0,026) και βρέθηκε ότι $\lambda_i \in [0,44, 2,93]$. (*Μέση τιμή* = 1,469, *Τυπική απόκλιση* = 0,633).

Με τον ίδιο τρόπο, στο Σχήμα 5.9 παρουσιάζεται το ιστόγραμμα και στο Σχήμα 5.10 παρουσιάζεται το διάγραμμα κανονικής πιθανότητας για την κυκλωματική πολυπλοκότητα $V(g)$ και ιδιαίτερα για το πηλίκο $10/V(g)$. Τα αποτελέσματα της κυκλωματικής πολυπλοκότητας για τα N_i τμήματα ακολούθησαν κανονική κατανομή (*Επίπεδο σημαντικότητας του ελέγχου κανονικότητας Shapiro-Wilk* = 0,805 >> 0,05, *Ασυμμετρία* = 0,229, *Κυρτότητα* = -0,582), ενώ βρέθηκε ότι $10/V(g)_i \in [0,22, 3,33]$ (*Μέση τιμή* = 1,581, *Τυπική απόκλιση* = 0,815).

Όσον αφορά τη συνδυαστική μετρική TIM, τα αποτελέσματα που προέκυψαν για τα N_i τμήματα ακολούθησαν κανονική κατανομή (*Επίπεδο σημαντικότητας του ελέγχου κανονικότητας Shapiro-Wilk* = 0,732 >> 0,05, *Ασυμμετρία* = 0,107, *Κυρτότητα* = -0,467) και βρέθηκε ότι $TIM_i \in [0,44, 2,28]$. (*Μέση τιμή* = 1,367, *Τυπική απόκλιση* = 0,506). Στο Σχήμα 5.11 παρουσιάζεται το ιστόγραμμα και στο

Σχήμα 5.12 παρουσιάζεται το διάγραμμα κανονικής πιθανότητας για τη συνδυαστική μετρική TIM.

Σχήμα 5.9: Ιστόγραμμα για το πηλίκο $10/V(g)$

Σχήμα 5.10: Διάγραμμα κανονικής πιθανότητας για το πηλίκο $10/V(g)$

Σχήμα 5.11: Ιστόγραμμα για τη συνδυαστική μετρική TIM

Σχήμα 5.12: Διάγραμμα κανονικής πιθανότητας για τη συνδυαστική μετρική TIM

Όπως και στην προηγούμενη περιπτωσιολογική μελέτη, η συσχέτιση ανάμεσα στις εσωτερικές μετρικές και τις εξωτερικές μετρήσεις βρέθηκε αξιοπρόσεκτα υψηλή. Στον πίνακα 5.1 παρουσιάζεται ο βαθμός συσχέτισης ανάμεσα στα αποτελέσματα των εξωτερικών μετρήσεων και στα αντίστοιχα αποτελέσματα των μετρικών επιπέδου γλώσσας λ, κυκλωματικής πολυπλοκότητας $V(g)$ και της συνδυαστικής μετρικής TIM.

Εσωτερικές μετρικές	Συσχέτιση με εξωτερικές μετρήσεις
λ	0.679
$10/V(g)$	0.683
TIM	0.730

Πίνακας 5.1: Αποτελέσματα συσχετίσεων

Τα scatter plots στα Σχήματα 5.13, 5.14 και 5.15 παρουσιάζουν αυτές τις συσχετίσεις. Στα scatter plots οι εξωτερικές μετρήσεις απεικονίζονται στον οριζόντιο άξονα, ενώ οι εσωτερικές μετρήσεις στον κατακόρυφο άξονα. Η διαγώνιος, που ονομάζεται και γραμμή συσχέτισης, είναι ο γεωμετρικός τύπος των σημείων που αντιστοιχούν σε συσχέτιση 100% ανάμεσα στις δύο μετρήσεις.

Σχήμα 5.13: Scatter plot εξωτερικών μετρήσεων και λ

Σχήμα 5.14: Scatter plot εξωτερικών μετρήσεων και $10/V(g)$

Από τα αποτελέσματα της περιπτώσιολογικής μελέτης είναι φανερό ότι τα αποτελέσματα των εσωτερικών μετρικών λογισμικού και η γνώμη των προγραμματιστών για τη συντηρησιμότητα βρίσκονται σε υψηλή συσχέτιση μεταξύ τους. Συγκεκριμένα, τμήματα με υψηλές εσωτερικές μετρήσεις εμφανίζουν ανάλογα υψηλά αποτελέσματα και στις εξωτερικές. Επιπλέον, ήταν κατά κύριο λόγο χαμηλή η εκτίμηση των προγραμματιστών για τη συντηρησιμότητα των τμημάτων που απέτυχαν όσον αφορά τις εσωτερικές μετρικές.

Παράλληλα, παρατηρήθηκε ότι μερικά τμήματα με χαμηλά αποτελέσματα στις μετρικές λογισμικού παρουσίασαν μεγάλη απόκλιση όσον αφορά τη γνώμη των προγραμματιστών για τη συντηρησιμότητά τους. Με άλλα λόγια, οι χαμηλές εσωτερικές μετρήσεις δεν συνεπάγονται απαραίτητα χαμηλές εξωτερικές μετρήσεις. Αυτή η απόκλιση οφειλόταν κυρίως στη φύση της εφαρμογής στην οποία ανήκαν τα παραπάνω τμήματα.

Σχήμα 5.15: Scatter plot εξωτερικών μετρήσεων και TIM

5.3.3 Συμπεράσματα

Το βασικό συμπέρασμα των περιπτώσιολογικών μελετών που παρουσιάστηκαν σε αυτό το κεφάλαιο είναι ότι οι εσωτερικές μετρικές αποτελούν μία πολύ καλή

αρχική ένδειξη για τη συντηρησιμότητα του λογισμικού. Όπως παρατηρήθηκε, η συσχέτισή τους με τη γνώμη των προγραμματιστών για τη συντήρηση είναι αρκετά υψηλή. Επομένως, οι μετρικές λογισμικού παρέχουν έναν εύκολο, αυτόματο και οικονομικό τρόπο για τον εντοπισμό και τη διόρθωση περιπτώσεων χαμηλής ποιότητας σύμφωνα με αυτό το χαρακτηριστικό της ποιότητας.

Σύμφωνα με τα αποτελέσματα από τις περιπτωσιολογικές μελέτες, ένα πρόγραμμα που εμφανίζει υψηλές εσωτερικές μετρήσεις θα παρουσιάσει υψηλή συντηρησιμότητα σύμφωνα με τα κριτήρια των προγραμματιστών για αυτό το χαρακτηριστικό. Οι χαμηλές εσωτερικές μετρήσεις όμως σε ένα πρόγραμμα δεν έχουν πάντοτε ως επακόλουθο αντίστοιχα χαμηλά αποτελέσματα για τη συντηρησιμότητα, παρόλο που αυτό συνήθως αναμένεται.

Οι προγραμματιστές πρέπει να διατηρούν τα αποτελέσματα των εσωτερικών μετρήσεων των προγραμμάτων τους σε κάποια αποδεκτά όρια. Αυτό μπορεί να επιτευχθεί με τη χρήση ενός κατάλληλου περιβάλλοντος προγραμματισμού το οποίο να επιτρέπει βασικές λειτουργίες εξασφάλισης ποιότητας, όπως την εφαρμογή ενός προγράμματος για τις μετρήσεις του παραγόμενου κώδικα. Παράλληλα, θα πρέπει να γίνεται μία σωστή επιλογή από εσωτερικές μετρικές για το εκάστοτε πρόγραμμα που πρόκειται να υλοποιηθεί και να καθορίζονται τα επιθυμητά όρια σε αυτές. Αυτά τα περιβάλλοντα προγραμματισμού μπορούν επίσης να προσφέρουν τη δυνατότητα συγκρίσεων ανάμεσα σε διαφορετικές εκδόσεις του ίδιου προγράμματος. Επιπλέον, επιτρέπουν στους προγραμματιστές να παρακολουθούν την πρόοδο της απόδοσής τους σε σχέση με τους μετρούμενους στόχους που έχουν ορίσει ή σε σχέση με τις προδιαγραφές του προγράμματος ποιότητας που ακολουθείται. Σε γενικές γραμμές, όσο υψηλότερα αποτελέσματα καταφέρουν στις μετρήσεις αυτές, τόσο λιγότερη προσπάθεια θα χρειαστεί να δαπανήσουν κατά τη φάση της συντήρησης. Αυτό μάλιστα έχει παρατηρηθεί και σε μετρήσεις διαφορετικών εκδόσεων του ίδιου έργου, όπου οι μεταγενέστερες και βελτιωμένες εκδόσεις εμφανίζουν υψηλότερα αποτελέσματα στις εσωτερικές μετρικές από ό,τι οι προγενέστερες εκδόσεις.

Αν όμως ένα τμήμα αποτύχει στις εσωτερικές μετρήσεις, είναι προτιμότερο να μην απορρίπτεται αυτόματα, παρόλο που πιθανότατα θα αποτύχει και στις εξωτερικές μετρήσεις σχετικά με τη συντηρησιμότητα. Αντιθέτως, αφού σε ορισμένες, έστω και λίγες, περιπτώσεις τα χαμηλά αποτελέσματα στις μετρικές είναι αναμενόμενα, οι προγραμματιστές πρέπει να επιθεωρήσουν προσεκτικά αυτόν τον κώδικα και να αποφασίσουν αν όντως χρειάζεται ή όχι βελτίωση ή ακόμα και αν χρειάζεται να ξαναγραφτεί. Δηλαδή, η επιτυχία για ένα έργο σε ένα πρόγραμμα εξασφάλισης ποιότητας βασισμένο σε εσωτερικές μετρήσεις αποτελεί συνθήκη ικανή, αλλά όχι αναγκαία για την εξασφάλιση της επιτυχίας στις μετρήσεις της άποψης των προγραμματιστών για τη συντηρησιμότητα του έργου. Μερικές φορές είναι προτιμότερο για μία εταιρεία παραγωγής λογισμικού να ρισκάρει όχι μόνο να διορθώνει, αλλά επίσης να ξαναγράφει συγκεκριμένα τμήματα κώδικα, έτσι ώστε να μειωθεί η προσπάθεια που θα χρειαστεί να καταβάλει κατά τη φάση της συντήρησης και παράλληλα θα αποφευχθούν ή θα λυθούν εγκαίρως πιθανά προβλήματα που θα αντιμετώπιζαν οι πελάτες.

Μεταβολές στις Απόψεις για την Ποιότητα

Λογισμικού

Στο κεφάλαιο αυτό παρουσιάζονται οι μεταβολές και οι αναθεωρήσεις στις απόψεις των χρηστών για την ποιότητα προϊόντων λογισμικού με την πάροδο του χρόνου. Τα αποτελέσματα αυτών των μεταβολών προέκυψαν από επαναληπτικές μετρήσεις που διεξήχθησαν στα πλαίσια της διατριβής, στις οποίες έγινε ένας διαχωρισμός των χρηστών σε διάφορες κατηγορίες ανάλογα με το επίπεδο εμπειρίας τους. Παρουσιάζεται η διαφορετικότητα ανάμεσα σε έμπειρους και άπειρους χρήστες και περιγράφονται οι ανάγκες για τη βελτίωση της διαδικασίας των μετρήσεων. Για το λόγο αυτό χρησιμοποιούνται και προσαρμόζονται κατάλληλα κανόνες από τη θεωρία Αναθεώρησης Άποψης και προτείνεται ένα μοντέλο για την εκτίμηση της γνώμης των χρηστών για την ποιότητα λογισμικού με αυτόματο τρόπο, χωρίς την ανάγκη επαναληπτικών μετρήσεων που έχουν συχνά μεγάλο κόστος. Τέλος, περιγράφονται οι προδιαγραφές του μοντέλου και ο τρόπος σχεδιάσής του για κάθε κατηγορία χρηστών.

6.1 Η ανάγκη για μετρήσεις αξιολόγησης

Είναι κοινώς αποδεκτό ότι η αξία των μετρήσεων των ποιοτικών χαρακτηριστικών όπως αυτά αντιλαμβάνονται από τους τελικούς χρήστες είναι πολύ μεγάλη. Τα διεθνή πρότυπα ποιότητας, όπως το ISO9001 [ISO00], το IEEE [IEEE89], τα βραβεία Baldridge [Bro91] και το CMM [PAU93], ενθαρρύνουν τις εταιρείες παραγωγής λογισμικού να μετρούν την άποψη των πελατών τους για την ποιότητα των προϊόντων τους. Οι μετρήσεις αυτές δεν παρέχουν απλά ενδείξεις για την αξιολόγηση του λογισμικού, αλλά επιπλέον επιτρέπουν την εφαρμογή πιο εκλεπτυσμένων τεχνικών ανάλυσης οι οποίες απαιτούνται από οργανισμούς με υψηλό επίπεδο ωριμότητας όσον αφορά τη διασφάλιση ποιότητας. Επίσης, οι μετρήσεις αξιολόγησης επιτρέπουν στην ομάδα ποιότητας μίας εταιρείας να εστιάσει ακριβώς στα ζητήματα που την ενδιαφέρει. Έτσι, τα ζητήματα αυτά μπορούν πλέον να αντιμετωπιστούν ως μετρήσιμες ποσότητες και επομένως παρέχεται η δυνατότητα εφαρμογής κανόνων της στατιστικής ανάλυσης με έναν ενδεδειγμένο και κατάλληλο για την περίπτωση τρόπο.

Παρόλα αυτά, στις παραπάνω μετρήσεις προκύπτουν συχνά διάφορες δυσκολίες σχετικά με την ποιότητα των δεδομένων τους, το υψηλό κόστος διεξαγωγής τους και τις συχνές διακυμάνσεις της γνώμης των χρηστών για το τελικό προϊόν με την πάροδο του χρόνου μέχρι αυτή να οριστικοποιηθεί σε μία τελική τιμή. Επιπρόσθετα, η διαδικασία του υπολογισμού της άποψης του χρήστη για την ποιότητα ενός λογισμικού απαιτεί αρκετό χρόνο, ειδικά στις περιπτώσεις όπου τα αποτελέσματα μίας μέτρησης πρέπει να επιβεβαιωθούν ή και να επαναπροσδιοριστούν από επακόλουθες μετρήσεις. Για το λόγο αυτό θα πρέπει να διεκπεραιωθούν οι ακόλουθοι στόχοι: α) να βελτιωθεί η ορθότητα και η ακρίβεια των δεδομένων εισόδου στις μετρήσεις αξιολόγησης, β) να βελτιωθεί η ανάλυση και η ερμηνεία των αποτελεσμάτων που προκύπτουν και γ) να βρεθεί μία μοντελοποίηση των αναθεωρήσεων άποψης των χρηστών, ώστε να εξάγεται εγκαίρως η γνώμη τους για την ποιότητα ενός προϊόντος ανά πάσα χρονική στιγμή. Δηλαδή, θα πρέπει να υιοθετηθεί μία απλούστερη, ταχύτερη και πιο αυτοματοποιήσιμη μέθοδος για να γίνεται μία έγκαιρη και αποδεκτή εκτίμηση της αναθεώρησης της γνώμης των διαφόρων ομάδων χρηστών, χωρίς να χρειάζεται να επαναλαμβάνεται κάθε τόσο η διαδικασία μέτρησης της άποψής τους.

Σύμφωνα με τη μέθοδο αυτή, θα πρέπει κατά τον υπολογισμό της γνώμης των χρηστών για ένα πρόγραμμα λογισμικού να λαμβάνονται επιπρόσθετα υπόψη τόσο η εμπειρία τους και η συνολική ικανότητά τους στη χρήση προγραμμάτων λογισμικού όσο και η ικανότητά τους στη χρήση του συγκεκριμένου προγράμματος. Επίσης θα πρέπει να υιοθετηθούν και να προσαρμοστούν κατάλληλα αρχές και κανόνες από τη θεωρία Αναθεώρησης Άποψης. Συγκεκριμένα, το μοντέλο που παρουσιάζεται σε αυτό το κεφάλαιο χρησιμοποιεί το σύστημα σφαιρών του Grove (Grove's Systems of Spheres) [GRO88], ώστε να παρουσιάζονται με ένα δόκιμο και παραστατικό τρόπο οι αναθεωρήσεις της γνώμης των χρηστών για την ποιότητα λογισμικού. Παράλληλα, αναλύονται και παρουσιάζονται διάφορες κατηγορίες χρηστών και πώς μπορεί να εφαρμοστεί το σύστημα σφαιρών του Grove σε κάθε περίπτωση. Στο συγκεκριμένο μοντέλο αντιμετωπίζονται όλα τα χαρακτηριστικά της ποιότητας που αφορούν τους τελικούς χρήστες και παρουσιάζεται με μαθηματικό τρόπο πώς η αλλαγή στην άποψη των χρηστών για ένα χαρακτηριστικό μπορεί να επιφέρει αλλαγή στην άποψή τους και για τα υπόλοιπα.

Κατά τη μέτρηση της γνώμης των χρηστών για την ποιότητα λογισμικού εστιάσαμε στα ποιοτικά χαρακτηριστικά (και τα υπο-χαρακτηριστικά αυτών) του προτύπου ISO9126 [ISO01] που αφορούν τους τελικούς χρήστες. Τα χαρακτηριστικά αυτά είναι η λειτουργικότητα, η αξιοπιστία, η αποδοτικότητα και η ευχρηστία. Ωστόσο, η προτεινόμενη μέθοδος μπορεί εύκολα να εφαρμοστεί με παρόμοιο τρόπο αν γίνει χρήση του μοντέλου του McCall [MCC77], του μοντέλου FURPS [GRA87], του μοντέλου CUPRIMDSO [KAN96] ή άλλων. Όπως είναι γνωστό, το πρότυπο ISO9126 ακολουθεί μία ιεραρχική δομή στη διάταξη των χαρακτηριστικών και υπο-χαρακτηριστικών που το αποτελούν, στην οποία θεωρητικά δεν υπάρχει επικάλυψη. Βρέθηκε όμως ότι παρόλο που τα χαρακτηριστικά αυτά είναι ανεξάρτητα, οι χρήστες έχουν την τάση να πιστεύουν πως υπάρχει μία συσχέτιση μεταξύ τους. Το γεγονός αυτό είναι περισσότερο εμφανές σε περιπτώσεις άπειρων χρηστών, όπου η έννοια του κάθε χαρακτηριστικού είναι περισσότερο συγκεχυμένη. Παρόλα αυτά, η αντίληψη για την ύπαρξη συσχέτισης παρατηρήθηκε και σε περιπτώσεις έμπειρων χρηστών, έστω και σε μικρότερο βαθμό. Για τη συλλογή των μετρήσεων της γνώμης των χρηστών έγινε χρήση ερωτηματολογίων με ερωτήσεις πολλαπλής επιλογής, έτσι ώστε να

καθοδηγείται ο χρήστης στην επιλογή μίας από τις προκαθορισμένες απαντήσεις που ήταν διατεταγμένες σε ισομήκη διαστήματα.

Για την ανάλυση των αποτελεσμάτων των μετρήσεων, δηλαδή για τη στατιστική επεξεργασία των απαντήσεων που προήλθαν από τους χρήστες που συμμετείχαν στο δείγμα της κάθε έρευνας, χρησιμοποιήθηκαν οι τέσσερις εναλλακτικοί μέθοδοι που παρουσιάζονται στην ενότητα 3.4. Όπως έχει ήδη αναφερθεί, οι μέθοδοι αυτές βασίζονται σε έρευνα με τη χρήση ερωτηματολογίου ή συνέντευξης, καθεμία από τις οποίες έχει διαφορετικό κόστος. Έτσι λοιπόν, χρησιμοποιήθηκαν στην ανάλυση προϊόντων λογισμικού για τα οποία μετρήθηκε η γνώμη των χρηστών που είχαν διαφορετικό επίπεδο εμπειρίας. Με την πάροδο του χρόνου έγιναν διαδοχικές μετρήσεις, τα αποτελέσματα των οποίων παρουσιάζονται στην επόμενη ενότητα.

6.2 Μετρήσεις με την πάροδο του χρόνου

6.2.1 Αποτελέσματα των μετρήσεων

Για να υπολογιστεί αποτελεσματικά η γνώμη των χρηστών για ένα προϊόν λογισμικού θα πρέπει να διενεργούνται μετρήσεις σε καθορισμένες χρονικές περιόδους. Παρόλο που η πρακτική αυτή δεν είναι δυνατόν να εφαρμοστεί στο επαγγελματικό περιβάλλον μίας εταιρείας παραγωγής λογισμικού λόγω του πολύ υψηλού κόστους, στα πλαίσια της έρευνας της διδακτορικής διατριβής [STA99a], [STA00], [STA05] διενεργήθηκαν μηνιαίες μετρήσεις, όπου χρησιμοποιήθηκε το ίδιο δείγμα χρηστών για το ίδιο υπό μέτρηση προϊόν λογισμικού. Κατά την ανάλυση των αποτελεσμάτων των μετρήσεων αυτών οι χρήστες που συμμετείχαν διαιρέθηκαν σε δύο βασικές κατηγορίες, στους έμπειρους και στους άπειρους. Στο Σχήμα 6.1 παρουσιάζεται ένα παράδειγμα από αποτελέσματα που προέκυψαν από αυτές τις μετρήσεις.

Στο σχήμα αυτό παρουσιάζονται τα όρια μέσα στα οποία μπορεί να μεταβληθεί η άποψη των χρηστών για την ποιότητα ενός προϊόντος λογισμικού με την πάροδο του χρόνου. Στον οριζόντιο άξονα απεικονίζεται ο χρόνος σε μήνες, ενώ στον κατακόρυφο άξονα απεικονίζεται η άποψη των χρηστών, η οποία υπολογίστηκε μέσω των μεθόδων που αναφέρθηκαν στην ενότητα 3.4. Η γνώμη του κάθε χρήστη σε κάθε μέτρηση παίρνει τιμές από 0 μέχρι 10, όπου το 10 αντιστοιχεί στην καλύτερη άποψη για τα ποιοτικά χαρακτηριστικά και το 0 στη χειρότερη. Η γραμμή AvgO παρουσιάζει τη μέση τιμή της γνώμης των χρηστών για την ποιότητα του προϊόντος, η οποία υπολογίζεται αφού μετρηθεί η τελική άποψη των χρηστών. Με την πάροδο του χρόνου η γνώμη των έμπειρων χρηστών κυμαίνεται ανάμεσα στις καμπύλες e1 και e2, ενώ η γνώμη των άπειρων χρηστών κυμαίνεται ανάμεσα στις καμπύλες i1 και i2. Οι τιμές και τα όρια που καθορίζουν αυτές οι καμπύλες διαφέρουν ανάλογα με το προϊόν λογισμικού. Συνήθως, η διαφορά ανάμεσα στις αρχικές τιμές των καμπύλων e1 και e2 είναι 3, η διαφορά ανάμεσα στις αρχικές τιμές των καμπύλων i1 και i2 είναι 6, ενώ η γραμμή AvgO βρίσκεται στη μέση ανάμεσα σε αυτές τις διαφορές. Με την πάροδο του χρόνου όμως, οι διαφορές ανάμεσα στις τιμές των καμπύλων μειώνονται.

Οι έμπειροι χρήστες, σε αντίθεση με τους άπειρους, σχηματίζουν από τα αρχικά στάδια της έκδοσης του προϊόντος μία άποψη για την ποιότητά του, η οποία είναι πολύ κοντά στην τελική τους άποψη. Οι άπειροι χρήστες όμως θα σχηματίσουν μία άποψη που θα πλησιάζει στην τελική τους άποψη αφού χρησιμοποιήσουν το προϊόν λογισμικού για μία μεγάλη χρονική περίοδο. Το μέγεθος αυτής της περιόδου εξαρτάται από το μέγεθος και την πολυπλοκότητα του προϊόντος, από τον αριθμό και

την ποικιλία των λειτουργιών που υποστηρίζει, από τη συχνότητα της χρήσης του προϊόντος και τις συνθήκες κάτω από τις οποίες χρησιμοποιείται, καθώς και από τη χρήση παρόμοιων προϊόντων λογισμικού. Η περίοδος αυτή συνήθως ποικίλει από έξι σε δώδεκα μήνες, οπότε και ο χρήστης αυτός μπορεί πλέον να θεωρηθεί ως έμπειρος για τη χρήση του συγκεκριμένου προϊόντος.

Σχήμα 6.1: Όρια στις απόψεις των χρηστών

Μετά από μία χρονική περίοδο η γραμμή AvgO παρουσιάζει συνήθως μία φθίνουσα πορεία, καθώς οι απαιτήσεις των χρηστών συνήθως αυξάνονται με την πάροδο του χρόνου. Αυτό το φαινόμενο οφείλεται σε παράγοντες, όπως τα διάφορα παρόμοια προϊόντα λογισμικού που ενδέχεται να κυκλοφορήσουν στην αγορά και οι διαρκείς εξελίξεις και βελτιώσεις στις δυνατότητες του υλικού των ηλεκτρονικών υπολογιστών. Επίσης, παρατηρήθηκε ότι όταν ένας έμπειρος χρήστης αξιολογεί αρχικά ένα προϊόν υψηλότερα ή χαμηλότερα από την τελική του άποψη, τότε με την πάροδο του χρόνου η τιμή της άποψής του δεν θα παρουσιάσει μεγάλες διακυμάνσεις, αλλά με σταθερό ρυθμό θα ακολουθεί φθίνουσα ή αύξουσα αντίστοιχα πορεία μέχρι να καταλήξει στην τελική της τιμή. Στην περίπτωση των άπειρων χρηστών όμως δεν παρατηρήθηκε παρόμοια προβλέψιμη μεταβολή. Αντίθετα, η γνώμη ενός άπειρου χρήστη συνήθως παρουσίαζε μεγάλη διακύμανση ανάμεσα στις καμπύλες i1 και i2. Με την πάροδο του χρόνου βέβαια ο βαθμός της διακύμανσης αυτής μειωνόταν μέχρις ότου να καταλήξει η τιμή της άποψης του άπειρου χρήστη για την ποιότητα του προϊόντος λογισμικού στην τελική της τιμή. Η μεταβολή της άποψης ενός άπειρου χρήστη παρουσιάζεται στο Σχήμα 6.2, όπου η γραμμή UO απεικονίζει ένα παράδειγμα από την πορεία της τιμής της άποψης του χρήστη και τις αλλαγές της με την πάροδο του χρόνου.

Η διακύμανση αυτή προέκυπτε όταν ένας άπειρος χρήστης ανακάλυπτε ένα νέο χαρακτηριστικό ή μία νέα δυνατότητα του προϊόντος που μέχρι τότε δεν γνώριζε ή όταν έβρισκε μία λειτουργία του προϊόντος την οποία αναζητούσε αλλά δεν είχε βρει μέχρι εκείνη τη στιγμή, με αποτέλεσμα να βαθμολογεί το προϊόν αρκετά υψηλότερα από την προηγούμενή του εκτίμηση. Παρόμοια, αν ο άπειρος χρήστης ανακάλυπτε ένα ελάττωμα στο προϊόν (είτε επρόκειτο για πραγματικό ελάττωμα είτε το θεωρούσε

ο ίδιος με την υποκειμενική του κρίση ως ελάττωμα, μερικές φορές μάλιστα λανθασμένα), τότε αξιολογούσε το προϊόν αρκετά χαμηλότερα από την προηγούμενή του εκτίμηση, ακόμα και αν επρόκειτο για κάποιον περιορισμό του προϊόντος που δεν μπορούσε να αποφευχθεί κατά το στάδιο της ανάπτυξής του.

Σχήμα 6.2: Μεταβολή στην άποψη των άπειρων χρηστών

Οι άπειροι χρήστες δεν αντιλαμβάνονται και δεν διακρίνουν ξεκάθαρα τους διαφορετικούς παράγοντες της ποιότητας λογισμικού. Αν ανακαλύψουν ένα χαρακτηριστικό που υποδηλώνει ότι το προϊόν λογισμικού αποτυγχάνει σε ένα συγκεκριμένο παράγοντα, τότε θεωρούν ότι το προϊόν αποτυγχάνει επίσης και στους υπόλοιπους. Αντίθετα, οι έμπειροι χρήστες αντιλαμβάνονται καλύτερα και πιο ξεκάθαρα τη φύση των διαφορετικών παραγόντων της ποιότητας και την ανεξαρτησία ανάμεσά τους. Έπειτα από ένα εύλογο χρονικό διάστημα, οι άπειροι χρήστες εξοικειώνονται με τα νέα χαρακτηριστικά ή τα ελαττώματα που ανακάλυψαν στον προϊόν και, ως επακόλουθο, η άποψή τους αρχίζει να κλίνει προς την τελική της τιμή, όπως συμβαίνει στην περίπτωση των έμπειρων χρηστών.

6.2.2 Βελτίωση της διαδικασίας μετρήσεων

Από τα αποτελέσματα των διάφορων μετρήσεων είναι φανερό ότι με την πάροδο του χρόνου: α) η γνώμη των έμπειρων χρηστών για την ποιότητα ενός προϊόντος λογισμικού προσεγγίζει την τιμή της τελικής τους άποψης και β) η απόκλιση της τιμής της άποψης των άπειρων χρηστών από την τελική της τιμή διαρκώς μειώνεται. Επομένως, όσο περισσότερο ένας χρήστης χρησιμοποιεί ένα προϊόν, τόσο μεγαλύτερη βαρύτητα πρέπει να δίνεται στην άποψή του. Με άλλα λόγια, ο παράγοντας τους χρόνου πρέπει επίσης να λαμβάνεται υπόψη για την αποτελεσματική μέτρηση της ποιότητας λογισμικού.

Σύμφωνα με τα παραπάνω, είναι προφανές ότι μετά από ένα μεγάλο χρονικό διάστημα οι άπειροι χρήστες θα σχηματίσουν μία άποψη για την ποιότητα του προϊόντος λογισμικού πολύ κοντά στην τελική τους άποψη. Το χρονικό αυτό διάστημα μπορεί να ξεπεράσει τους έξι μήνες, θεωρώντας ότι κατά το διάστημα αυτό ο πραγματικός χρόνος χρήσης του προϊόντος είναι ικανοποιητικός, δηλαδή είναι μέσα

στα προβλεπόμενα όρια. Ως αποτέλεσμα, για να μπορούμε να καθορίσουμε την ποιότητα ενός προϊόντος λογισμικού στα πρώτα στάδια της έκδοσής του, το δείγμα των χρηστών που θα συμμετέχουν στην έρευνα θα πρέπει να περιοριστεί στους έμπειρους χρήστες.

Επιπρόσθετα, στα αρχικά αυτά στάδια η γνώμη των άπειρων χρηστών παρουσιάζει μεγάλη διακύμανση. Η γνώμη τους μπορεί να ληφθεί υπόψη μόνο αν το δείγμα των χρηστών είναι αρκετά μεγάλο, ώστε να μπορεί να θεωρηθεί αντιπροσωπευτικό και επομένως να μπορεί να εξασφαλίσει αξιόπιστα αποτελέσματα. Παράλληλα, πρέπει να δίνεται μεγαλύτερη βαρύτητα στη γνώμη των έμπειρων χρηστών από ό,τι στη γνώμη των άπειρων, ανεξάρτητα αν οι έμπειροι χρήστες που συμμετέχουν στη μέτρηση είναι σαφώς λιγότεροι. Επιπλέον, από τη μελέτη των αποτελεσμάτων των διαδοχικών μετρήσεων για τον κάθε χρήστη ξεχωριστά που συμμετείχε στην έρευνα, παρατηρήθηκε ότι όσο μεγαλύτερος ήταν ο βαθμός της διακύμανσης που παρουσίαζε η άποψή του με την πάροδο του χρόνου, τόσο δυσκολότερο ήταν για αυτόν να μάθει και να εξοικειωθεί με τα χαρακτηριστικά του προϊόντος λογισμικού.

Κατά τη διάρκεια των δύο πρώτων μηνών της χρήσης ενός προϊόντος από έναν άπειρο χρήστη η διακύμανση αυτή μπορεί να ξεπερνούσε και το 25% της τιμής της τελικής του άποψης. Ένα παράδειγμα ποιοτικών χαρακτηριστικών, όπως αυτά διακρίνονται στο πρότυπο ISO9126, που οι περισσότεροι χρήστες λανθασμένα θεωρούν ότι σχετίζονται μεταξύ τους είναι η “καταλληλότητα” (υπο-χαρακτηριστικό της λειτουργικότητας) και η “πρακτικότητα” (υπο-χαρακτηριστικό της ευχρηστίας). Επιπλέον, όπως δηλώνεται στο ISO9241-11 [BEV97], μεταβολές στην αξιοπιστία, τη λειτουργικότητα και την αποδοτικότητα ενός προγράμματος ενδέχεται να έχουν αντίκτυπο στην απόδοση του χρήστη και την ικανοποίησή του και επομένως να έχουν αντίκτυπο και στην ευχρηστία του προγράμματος.

6.3 Η ανάγκη ενός μοντέλου

Όπως αναφέρθηκε παραπάνω, το πρότυπο ISO9126 ακολουθεί μία ιεραρχική δομή στη διάταξη των χαρακτηριστικών και υπο-χαρακτηριστικών που το αποτελούν, στην οποία θεωρητικά δεν υπάρχει επικάλυψη. Παρόλο που τα χαρακτηριστικά αυτά είναι ανεξάρτητα, οι χρήστες, και περισσότερο οι άπειροι χρήστες, έχουν την τάση να πιστεύουν πως υπάρχει μία συσχέτιση μεταξύ τους. Λαμβάνοντας αυτό υπόψη, μπορούμε να κατανοήσουμε πώς η μεταβολή στη γνώμη του χρήστη σχετικά με ένα ποιοτικό χαρακτηριστικό μπορεί να επηρεάσει και τη γνώμη του για διάφορα άλλα ποιοτικά χαρακτηριστικά.

Στην προηγούμενη ενότητα παρουσιάστηκε πώς η γνώμη των χρηστών μεταβάλλεται με την πάροδο του χρόνου, ιδιαίτερα στην περίπτωση των άπειρων χρηστών, των οποίων η γνώμη παρουσιάζει μεγάλη διακύμανση. Η διαδικασία του εντοπισμού αυτών των μεταβολών μέσω μετρήσεων μπορεί να είναι προβληματική, καθώς απαιτεί πιθανότατα τη διεξαγωγή τόσο συχνών ερευνών και μετρήσεων με τη χρήση ερωτηματολογίων που την καθιστά ως μία διαδικασία ιδιαίτερα υψηλού κόστους ή ακόμα και ανέφικτη. Επίσης, πρόκειται για μία αρκετά χρονοβόρα διαδικασία, ειδικά όταν τα αποτελέσματα μίας μέτρησης θα πρέπει να επιβεβαιώνονται ή να αναθεωρούνται από επακόλουθες μετρήσεις.

Η εναλλακτική λύση που προτείνεται σε αυτή τη διατριβή είναι η χρήση κανόνων αναθεώρησης άποψης για τη μοντελοποίηση των γνωστικών καταστάσεων (belief

states) των χρηστών. Χρησιμοποιώντας αυτό το μοντέλο, όταν η γνώμη ενός χρήστη για ένα συγκεκριμένο ποιοτικό χαρακτηριστικό μεταβάλλεται, μπορούμε να εκτιμήσουμε με αυτόματο τρόπο τη γνώμη του για όλα τα ποιοτικά χαρακτηριστικά που θεωρεί ότι σχετίζονται με αυτό, χωρίς να χρειάζεται να μετρήσουμε ξανά τη γνώμη του για κάθε ένα από αυτά. Επομένως, οι αναθεωρήσεις της άποψης ενός χρήστη για την ποιότητα ενός προϊόντος λογισμικού μπορούν να εντοπιστούν και να εκτιμηθούν χωρίς την ανάγκη διεξαγωγής νέων μετρήσεων τόσο συχνά, αφού μπορούμε εύκολα (π.χ. με ένα απλό τηλεφώνημα) να βρούμε για ποιο ποιοτικό χαρακτηριστικό μεταβλήθηκε η άποψη του χρήστη και στη συνέχεια να εκτιμήσουμε τη συνολική του άποψη για την ποιότητα του προϊόντος αυτού.

6.3.1 Προδιαγραφές του προτεινόμενου μοντέλου

Προφανώς, για να καθοριστεί η αρχική γνώμη του χρήστη για την ποιότητα, θα πρέπει να διεξαχθεί μία ανάλογη εξωτερική μέτρηση. Με τη μέτρηση αυτή θα υπολογιστεί η γνώμη του χρήστη για κάθε ποιοτικό χαρακτηριστικό, γεγονός που θα επιτρέψει τον καθορισμό της αρχικής γνωστικής κατάστασης του χρήστη για την ποιότητα του προϊόντος. Με την πάροδο του χρόνου αυτή η γνωστική κατάσταση μπορεί να μεταβληθεί εξαιτίας των αναθεωρήσεων της άποψης του χρήστη. Ο βασικός σκοπός του προτεινόμενου μοντέλου είναι ο υπολογισμός της νέας γνωστικής κατάστασης του χρήστη, χωρίς να χρειαστεί η διεξαγωγή μίας νέας εξωτερικής μέτρησης. Η βασική φιλοσοφία του μοντέλου απεικονίζεται στο Σχήμα 6.3, όπου οι είσοδοι του μοντέλου είναι η τρέχουσα γνωστική κατάσταση του χρήστη (Initial BS) και η νέα πληροφορία (New Information) για το προϊόν λογισμικού που ανακαλύπτει, ενώ η έξοδος του μοντέλου είναι η νέα γνωστική κατάσταση (New BS) του χρήστη.

Σχήμα 6.3: Η βασική φιλοσοφία του προτεινόμενου μοντέλου

Κάθε ποιοτικό χαρακτηριστικό του προτύπου ISO9126 (ή παράγοντας του μοντέλου του McCall) αποτελείται από διάφορα κριτήρια. Για να καθοριστεί πλήρως η γνώμη των χρηστών για την ποιότητα λογισμικού θα πρέπει να υπολογιστεί η γνώμη τους για κάθε ένα από τα παραπάνω χαρακτηριστικά και τα κριτήριά τους που αφορούν τους χρήστες. Θεωρώντας ότι η γνώμη των χρηστών μπορεί να αναπαρασταθεί ως μία βαθμολογία (π.χ. ένας αριθμός από 0 έως 10, όπου όσο πιο υψηλή είναι η βαθμολογία τόσο μεγαλύτερη και η ικανοποίηση του χρήστη), η γνωστική κατάσταση του κάθε χρήστη σχετικά με την ποιότητα λογισμικού αποτελείται από τη βαθμολογία που έδωσε για το κάθε κριτήριο (χαρακτηριστικό ή υπο-χαρακτηριστικό). Με άλλα λόγια, η γνωστική κατάσταση BS μπορεί να

αναπαρασταθεί με την ακόλουθη μορφή: $BS=\{A_1=x_1, A_2=x_2, \dots, A_n=x_n\}$, όπου A_i είναι το i από τα n κριτήρια που απαιτούνται για να καθοριστεί πλήρως η γνωστική κατάσταση του χρήστη και x_i η βαθμολογία για το συγκεκριμένο κριτήριο σύμφωνα με τη γνώμη του χρήστη, η οποία παίρνει τιμές από 0 έως 10.

6.3.2 Καθορισμός συνόλου από περιορισμούς

Όπως ήδη αναφέρθηκε, οι χρήστες ενός προϊόντος λογισμικού έχουν την άποψη πως υπάρχει μία συσχέτιση ανάμεσα στα ποιοτικά χαρακτηριστικά του λογισμικού. Η αναπαράσταση της συσχέτισης αυτής που ακολουθήθηκε στο προτεινόμενο μοντέλο γίνεται με τον καθορισμό ενός συνόλου από περιορισμούς ανάμεσα στις βαθμολογίες του κάθε χαρακτηριστικού ποιότητας που αφορούν τον τελικό χρήστη. Κάθε περιορισμός καθορίζει την πιθανή σχέση ανάμεσα σε δύο χαρακτηριστικά και ακολουθεί τον παρακάτω φορμαλισμό: $A_i \in [h_1, h_2] \Rightarrow A_j \in [c_1, c_2]$, όπου τα A_i και A_j είναι η βαθμολογίες για τα δύο διαφορετικά ποιοτικά χαρακτηριστικά i και j , ενώ τα $[h_1, h_2]$ και $[c_1, c_2]$ είναι υποσύνολα του συνόλου $[0,10]$, το οποίο σημαίνει ότι αν η βαθμολογία για το χαρακτηριστικό i παίρνει τιμές από h_1 έως h_2 , τότε η βαθμολογία για το χαρακτηριστικό j θα παίρνει τιμές από c_1 έως c_2 .

Για να γίνει ένας διαχωρισμός στις απόψεις των χρηστών για τις σχέσεις ανάμεσα στα ποιοτικά χαρακτηριστικά, θα πρέπει να καθοριστούν διαφορετικά σύνολα από περιορισμούς ανάλογα με το επίπεδο εμπειρίας των χρηστών. Εφόσον, όπως παρουσιάστηκε παραπάνω, οι άπειροι χρήστες θεωρούν σε μεγαλύτερο βαθμό από τους έμπειρους ότι υπάρχει συσχέτιση ανάμεσα σε αυτά τα χαρακτηριστικά, τότε θα πρέπει όσο πιο έμπειρος είναι ο χρήστης τόσο λιγότερο αυστηροί να είναι οι περιορισμοί. Επομένως, οι χρήστες θα πρέπει αρχικά να κατηγοριοποιούνται σε διάφορα επίπεδα εμπειρίας και στο καθένα να αντιστοιχεί διαφορετικό σύνολο από περιορισμούς. Η κατηγοριοποίηση αυτή μπορεί να επιτευχθεί με τη χρήση ενός ανάλογου ερωτηματολογίου μέτρησης και καθορισμού της εμπειρίας του χρήστη. Καθώς η εμπειρία ενός χρήστη αυξάνει με την πάροδο του χρόνου, θα πρέπει μετά από ένα εύλογο χρονικό διάστημα ο χρήστης να αντιστοιχιστεί σε σύνολο περιορισμών μεγαλύτερης εμπειρίας.

Το παρακάτω παράδειγμα παρουσιάζει τους πιθανούς περιορισμούς ανάμεσα σε τρία ποιοτικά χαρακτηριστικά:

$A_1 \in [0,2] \Rightarrow A_2 \in [0,4]$	$A_2 \in [0,2] \Rightarrow A_1 \in [0,4]$
$A_1 \in [4,6] \Rightarrow A_2 \in [3,7]$	$A_2 \in [5,7] \Rightarrow A_1 \in [2,8]$
$A_1 \in [7,10] \Rightarrow A_2 \in [5,10]$	$A_2 \in [9,10] \Rightarrow A_1 \in [7,10]$
$A_1 \in [0,2] \Rightarrow A_3 \in [0,4]$	$A_3 \in [0,1] \Rightarrow A_1 \in [0,5]$
$A_1 \in [8,10] \Rightarrow A_3 \in [5,10]$	$A_3 \in [9,10] \Rightarrow A_1 \in [6,10]$
$A_2 \in [0,2] \Rightarrow A_3 \in [0,5]$	$A_3 \in [0,2] \Rightarrow A_2 \in [0,5]$
$A_2 \in [5,7] \Rightarrow A_3 \in [3,8]$	$A_3 \in [9,10] \Rightarrow A_2 \in [6,10]$
$A_2 \in [9,10] \Rightarrow A_3 \in [6,10]$	

Αν για παράδειγμα η τρέχουσα γνωστική κατάσταση του χρήστη είναι η $CBS=\{A_1=6, A_2=4, A_3=2\}$ και η γνώμη του για το χαρακτηριστικό A_1 αλλάξει σε 7, τότε αυτή η αναθεώρηση θα πρέπει να προκαλέσει τις ελάχιστες δυνατές μεταβολές

στη γνωστική του κατάσταση, σύμφωνα με τους κανόνες της θεωρίας Αναθεώρησης Άποψης. Ως αποτέλεσμα, με βάση τους περιορισμούς του παραδείγματος, η νέα γνωστική κατάσταση του χρήστη θα είναι η $NBS = \{A_1=7, A_2=5, A_3=3\}$.

Είναι προφανές από το παράδειγμα ότι οι προτεινόμενοι περιορισμοί δεν είναι απαραίτητο να καλύπτουν όλες τις πιθανές τιμές των χαρακτηριστικών. Έτσι, στο παραπάνω παράδειγμα δεν υπάρχει περιορισμός για το χαρακτηριστικό A_3 όταν $A_1 \in [3,7]$, δεδομένου ότι όλοι οι υπόλοιποι περιορισμοί ικανοποιούνται. Επιπλέον, το σύνολο των περιορισμών πρέπει να είναι συνεπές. Αυτό σημαίνει ότι α) κάθε αποδεκτή γνωστική κατάσταση θα πρέπει να ικανοποιεί όλους τους περιορισμούς και β) δεν θα πρέπει να υπάρχει περιορισμός που να παρουσιάζει ασυμβατότητα με κάποιον άλλον.

6.4 Έγκαιρη εκτίμηση της γνώμης των χρηστών

Από τα αποτελέσματα των μετρήσεων που παρουσιάστηκαν στις προηγούμενες ενότητες γίνεται προφανές ότι η βαρύτητα της γνώμης του χρήστη για την ποιότητα ενός προϊόντος λογισμικού αυξάνει με την πάροδο του χρόνου. Επομένως, θα πρέπει να λαμβάνεται υπόψη κάθε νέο χαρακτηριστικό του προϊόντος που ανακαλύπτει ο χρήστης, το οποίο θα οδηγήσει αδιαμφισβήτητα σε αναθεώρηση της άποψής του για το προϊόν αυτό. Επιπλέον, καθώς ανακαλύπτουν οι χρήστες νέα χαρακτηριστικά του λογισμικού που χρησιμοποιούν, γίνονται ολοένα και πιο σίγουροι για την άποψή τους για την ποιότητά του.

6.4.1 Σύστημα Σφαιρών του Grove

Όπως έχει ήδη αναφερθεί, για να γίνει εφικτή η εκτίμηση της νέας, αναθεωρημένης άποψης των χρηστών, χωρίς να χρειάζονται επαναλαμβανόμενες μετρήσεις που έχουν μεγάλο κόστος, χρησιμοποιήθηκαν και προσαρμόστηκαν κανόνες από τη θεωρία της Αναθεώρησης Άποψης [GAR88a], [PEP96] και ιδιαίτερα του Συστήματος Σφαιρών του Grove [GRO88].

Σχήμα 6.4: Σύστημα Σφαιρών του Grove

Σε αυτό το σύστημα, κάθε γνωστική κατάσταση K μπορεί να αναπαρασταθεί ως ένα υποσύνολο $[K]$ του M που αποτελείται από όλες τις μέγιστες συλλογές στις

οποιές περιλαμβάνονται όλες οι προτάσεις του K , όπου το M είναι το σύνολο όλων των πιθανών κόσμων που μπορούν να περιγραφούν σε μία προτασιακή γλώσσα L .

Ένα σύστημα σφαιρών με κέντρο το $[K]$ είναι μία συλλογή κόσμων S από υποσύνολα του M τα οποία μπορούν να απεικονιστούν σχηματικά όπως στο Σχήμα 6.4. Σε αυτό το σύστημα όσο πιο εσωτερική είναι μία σφαίρα τόσο πιο πιθανός και ο κόσμος με κέντρο το $[K]$. Όταν μία νέα πρόταση A φαίνεται να είναι αληθής, όπου $A \notin [K]$, και η πρόταση A είναι πάντα αξιόπιστη, τότε ο πιθανός κόσμος θα πρέπει να αναθεωρηθεί ώστε να συμπεριλαμβάνει το A . Επομένως, για να υπάρχουν οι ελάχιστες δυνατές αλλαγές από την αρχική γνωστική κατάσταση, θα πρέπει να χρησιμοποιηθεί η μικρότερη σφαίρα (S_A), όπου $S_A \cap [A] \neq \emptyset$. Η νέα γνωστική κατάσταση, δηλαδή ο νέος κόσμος μετά την αναθεώρηση, είναι το $C(A)$, όπου $C(A) = [K * A] = S_A \cap [A]$.

6.4.2 Σχεδίαση του μοντέλου

Προσαρμόζοντας τους κανόνες του Συστήματος Σφαιρών του Grove μπορεί να σχεδιαστεί ένα εναλλακτικό μοντέλο αναπαράστασης της γνώμης του χρήστη για την ποιότητα λογισμικού. Τα διάφορα σημεία, δηλαδή οι πιθανοί κόσμοι, στο σύστημα αυτού του μοντέλου αντιπροσωπεύουν τις απόψεις του χρήστη για ένα προϊόν λογισμικού σύμφωνα με τα ποιοτικά χαρακτηριστικά που τον ενδιαφέρουν. Για κάθε γνωστική κατάσταση K που έχει ο χρήστης για την ποιότητα του προϊόντος υπάρχει ένα σύστημα σφαιρών S στο M με κέντρο το $[K]$ που δείχνει την απεικόνιση της γνωστικής κατάστασης αυτής μέσα στο M . Επιπλέον, η γνωστική κατάσταση K πρέπει να περιλαμβάνει τις απόψεις του χρήστη για το προϊόν σύμφωνα με όλα τα ποιοτικά χαρακτηριστικά του λογισμικού.

Το πρώτο βήμα στη σχεδίαση του μοντέλου είναι η εύρεση διαφόρων χαρακτηριστικών, στοιχείων και γεγονότων για κάθε ποιοτικό χαρακτηριστικό, τα οποία να μπορούν να υποδηλώσουν και να προσδιορίσουν τη γνώμη των χρηστών για αυτό το ποιοτικό χαρακτηριστικό. Αυτά τα χαρακτηριστικά-γεγονότα είναι προτάσεις που προσδιορίζουν του πιθανούς κόσμους του μοντέλου.

Το δεύτερο βήμα είναι η αναπαράσταση των πιθανών κόσμων στο μοντέλο. Το μοντέλο θα πρέπει για κάθε απλό γεγονός E να χωρίζεται με μία καμπύλη σε δύο τμήματα. Το ένα τμήμα είναι αυτό στο οποίο ισχύει το γεγονός (το τμήμα E), ενώ το άλλο είναι αυτό στο οποίο δεν ισχύει (το τμήμα $\neg E$). Αν σε έναν πιθανό κόσμο S_x ο χρήστης είναι σίγουρος ότι το γεγονός E είναι αληθές, τότε το S_x πρέπει να βρίσκεται στο τμήμα E και αντιστρόφως. Αν όμως ο χρήστης δεν είναι σίγουρος για το γεγονός E (δηλαδή δεν ξέρει αν ισχύει ή όχι) τότε το S_x πρέπει να βρίσκεται και στα δύο τμήματα. Αφού χωριστεί το μοντέλο με καμπύλες για κάθε ένα γεγονός ξεχωριστά, τότε γνωρίζοντας τη γνώμη του χρήστη για τα γεγονότα αυτά μπορούμε να αναπαραστήσουμε τον πιθανό κόσμο του στο μοντέλο.

Το τρίτο βήμα είναι ο προσδιορισμός του τρόπου με τον οποίο ο πιθανός κόσμος ενός χρήστη μπορεί να αλλάξει στο μοντέλο όταν ο χρήστης αναθεωρεί την άποψή του για ένα γεγονός. Ο νέος πιθανός κόσμος θα πρέπει να οριοθετείται από τα τμήματα του μοντέλου στα οποία αυτό το γεγονός ισχύει ή όχι. Επιπλέον, θα πρέπει να προσδιοριστεί κατά την αναθεώρηση αυτή με ποιο τρόπο η άποψη του χρήστη και για τα υπόλοιπα γεγονότα μπορεί να αλλάξει. Αν λοιπόν η αναθεώρηση επιφέρει επιπλέον αλλαγές, τότε θα έχουμε άμεσες και ενίοτε ριζικές μεταβολές στη γνωστική κατάσταση του χρήστη. Όλες οι αναθεωρήσεις θα πρέπει να έχουν ως αποτέλεσμα τις ελάχιστες δυνατές αλλαγές στη γνωστική κατάσταση και αφού περατωθούν όλες, ο

νέος πιθανός κόσμος που θα προκύψει θα αναπαριστά τη νέα γνωστική κατάσταση του χρήστη για την ποιότητα του προϊόντος λογισμικού.

6.4.3 Μοντελοποίηση των διαφόρων κατηγοριών των χρηστών

Όπως αναφέρθηκε στις προηγούμενες ενότητες, οι χρήστες πρέπει να διαχωρίζονται σε διάφορες κατηγορίες σύμφωνα με το επίπεδο εμπειρίας τους, αφού η γνώμη τους για ένα προϊόν λογισμικού μεταβάλλεται με διαφορετικό τρόπο. Επομένως, το προτεινόμενο μοντέλο θα πρέπει να διαφέρει για κάθε κατηγορία χρηστών, επειδή η αναθεώρηση στην άποψη για ένα κριτήριο ποιότητας μπορεί να έχει διαφορετικά αποτελέσματα στις γνωστικές καταστάσεις της κάθε κατηγορίας. Με άλλα λόγια, όταν οι άπειροι χρήστες ανακαλύπτουν ότι ένα γεγονός ισχύει ή όχι, τότε η γνώμη τους για την ποιότητα του προϊόντος θα μεταβληθεί σε μεγαλύτερο βαθμό από ό,τι θα μεταβληθεί η γνώμη των έμπειρων χρηστών.

Επειδή οι άπειροι χρήστες θεωρούν ότι υπάρχει κάποια συσχέτιση ανάμεσα στους διάφορους παράγοντες που απαρτίζουν την ποιότητα λογισμικού (ποιοτικά χαρακτηριστικά), τότε συνήθως όταν μένουν ικανοποιημένοι ή όχι με έναν παράγοντα, η γνώμη τους για τους υπόλοιπους θα είναι ανάλογη. Άρα, μία αναθεώρηση της άποψής τους για έναν παράγοντα συχνά οδηγεί και σε αναθεώρηση της άποψής τους και για τους υπόλοιπους. Αντιθέτως, σύμφωνα με τη γνώμη των έμπειρων χρηστών για την ποιότητα, οι διάφοροι παράγοντες που την απαρτίζουν δεν είναι αλληλοεξαρτώμενοι. Σε αυτή την περίπτωση μία αναθεώρηση της άποψης για έναν παράγοντα μπορεί να επηρεάσει την άποψη και για τους υπόλοιπους μόνο αν πρόκειται για μία ριζική αναθεώρηση που επιφέρει μεγάλες μεταβολές στη γνώμη του χρήστη.

Σχήμα 6.5: Μοντέλο της κατηγορίας των έμπειρων χρηστών

Το διαφορετικό επίπεδο αλληλεξάρτησης ανάμεσα στους παράγοντες της ποιότητας λογισμικού σύμφωνα με τη γνώμη των χρηστών οδηγεί στη διαφορετική σχεδίαση του προτεινόμενου μοντέλου ανάλογα με την κατηγορία των χρηστών. Στην περίπτωση των έμπειρων χρηστών, τα όρια που δηλώνουν αν ένα γεγονός ισχύει ή όχι απεικονίζονται με τέτοιο τρόπο, ώστε η αναθεώρηση στην άποψη για ένα γεγονός να έχει ως αποτέλεσμα τις ελάχιστες αλλαγές στη γνωστική κατάσταση των χρηστών. Με άλλα λόγια, τα όρια αυτά είναι ανεξάρτητα. Επομένως, δεν πρέπει να

παρατηρούνται περιοχές με μεγάλη πυκνότητα σε καμπύλες-όρια γεγονότων. Διαφορετικά, αν η γνωστική κατάσταση ενός χρήστη απεικονιζόταν ως μία σφαίρα μέσα στο σύστημα, η αναθεώρηση στη γνώμη για ένα γεγονός θα οδηγούσε συχνά σε μία ριζική αναθεώρηση, κάτι που δεν παρατηρείται στην περίπτωση των έμπειρων χρηστών. Το μοντέλο για την περίπτωση αυτή θα έχει μία μορφή όπως αυτή που παρουσιάζεται στο Σχήμα 6.5.

Αντιθέτως, στην περίπτωση των άπειρων χρηστών, η ανακάλυψη κάθε νέου χαρακτηριστικού του προϊόντος λογισμικού μπορεί να μεταβάλλει την άποψή τους ακόμα και για όλους του ποιοτικούς παράγοντες. Επομένως, το μοντέλο σε αυτή την περίπτωση θα σχεδιαστεί με τέτοιο τρόπο, όπου η αναθεώρηση στη γνώμη των χρηστών για έναν παράγοντα της ποιότητας θα οδηγήσει σε ριζικές αλλαγές στη γνώμη τους. Ο πιθανός κόσμος των άπειρων χρηστών δεν παρουσιάζει την ίδια σταθερότητα με αυτόν που έχουν οι έμπειροι χρήστες. Αφού λοιπόν οι άπειροι χρήστες έχουν συνήθως την αντίληψη ότι τα γεγονότα του μοντέλου αλληλεξαρτώνται, τα όρια αυτών των γεγονότων θα πρέπει να σχεδιάζονται σχετικά κοντά το ένα στο άλλο. Άρα, στο μοντέλο θα εμφανίζονται περιοχές με μεγάλη πυκνότητα σε καμπύλες-όρια γεγονότων, όπως φαίνεται στο Σχήμα 6.6.

Τα Σχήματα 6.5 και 6.6 παρουσιάζουν επίσης τη διαφορά ανάμεσα στους έμπειρους και τους άπειρους χρήστες μετά από μία αναθεώρηση άποψης. Για παράδειγμα, στο νέο κόσμο $C(A)$, που προέκυψε από την αναθεώρηση για το γεγονός A , οι άπειροι χρήστες σχηματίζουν μία άποψη για όλα τα γεγονότα του σχήματος που είναι εντελώς διαφορετική από την αρχική τους, δηλαδή από τον κόσμο $[K]$. Αντιθέτως, οι έμπειροι χρήστες αναθεωρούν την άποψή τους μόνο για ένα επιπλέον γεγονός του σχήματος.

Σχήμα 6.6: Μοντέλο της κατηγορίας των άπειρων χρηστών

6.4.4 Αναπαράσταση περιορισμών στο Σύστημα Σφαιρών

Η αναπαράσταση των περιορισμών, όπως παρουσιάστηκε στην ενότητα 6.3.2, μπορεί να κριθεί ως πλήρης και ορθή σε σχέση με το Σύστημα Σφαιρών του Grove. Με άλλα λόγια, μπορεί να αποδειχτεί ότι κάθε σύνολο από περιορισμούς αντιστοιχεί σε ένα και μόνο ένα Σύστημα Σφαιρών και αντίστροφα κάθε Σύστημα Σφαιρών αντιστοιχεί σε ένα και μόνο ένα σύνολο από περιορισμούς. Η πλήρης απόδειξη

παρουσιάζεται αναλυτικά στο κεφάλαιο 7. Όταν σχεδιάζεται το Σύστημα των Σφαιρών, οι διαφορετικοί πιθανοί κόσμοι, που αναπαριστούν τις διαφορετικές γνωστικές καταστάσεις του χρήστη, πρέπει να τοποθετηθούν με τέτοιο τρόπο ώστε να ικανοποιείται ο κανόνας της θεωρίας Αναθεώρησης Άποψης για ελάχιστες μεταβολές κατά τη διάρκεια μίας αναθεώρησης. Σύμφωνα με αυτόν τον κανόνα, αν η γνωστική κατάσταση ενός χρήστη είναι το $\{x_1, x_2, x_3, \dots, x_n\}$, τότε οι πλησιέστερες γνωστικές καταστάσεις σε αυτή είναι όλα τα $\{y_1, y_2, y_3, \dots, y_n\}$, όπου $\sum |x_i - y_i| = 1$, δεδομένου ότι όλες οι παραπάνω γνωστικές καταστάσεις είναι αποδεκτές σύμφωνα με το σύνολο των περιορισμών. Αυτές οι γνωστικές καταστάσεις θα τοποθετηθούν στην πρώτη σφαίρα γύρω από το $\{x_1, x_2, x_3, \dots, x_n\}$. Με παρόμοιο τρόπο, οι επόμενες πλησιέστερες γνωστικές καταστάσεις είναι όλα τα $\{z_1, z_2, z_3, \dots, z_n\}$, όπου $\sum |x_i - z_i| = 2$, δεδομένου ότι όλες οι παραπάνω γνωστικές καταστάσεις είναι αποδεκτές σύμφωνα με το σύνολο των περιορισμών. Αυτές οι γνωστικές καταστάσεις θα τοποθετηθούν στη δεύτερη σφαίρα γύρω από το $\{x_1, x_2, x_3, \dots, x_n\}$. Αν μία γνωστική κατάσταση είναι μη αποδεκτή, τότε ο πιθανός κόσμος της στο μοντέλο του Συστήματος Σφαιρών πρέπει να τοποθετηθεί πολύ μακριά από τις αποδεκτές καταστάσεις, δηλαδή στην τελευταία σφαίρα γύρω από το $\{x_1, x_2, x_3, \dots, x_n\}$. Με αυτό τον τρόπο, ο κόσμος που αντιστοιχεί στη μη αποδεκτή γνωστική κατάσταση θα θεωρείται ως απίθανος.

Όταν ένας χρήστης αναθεωρεί την άποψή του για τη βαθμολογία δύο ή περισσότερων κριτηρίων ταυτόχρονα, θα πρέπει να ακολουθηθούν αντίστοιχα βήματα για την εκτίμηση της νέας γνωστικής του κατάστασης. Αν όμως οι δύο ταυτόχρονες αναθεωρήσεις οδηγούν σε μη αποδεκτή γνωστική κατάσταση, σύμφωνα με το σύνολο των περιορισμών που έχει καθοριστεί για το επίπεδο εμπειρίας που ανήκει ο χρήστης, τότε θα πρέπει να προσδιοριστεί για αυτόν ένα σύνολο περιορισμών υψηλότερου επιπέδου εμπειρίας. Αυτό συμβαίνει όταν με την πάροδο του χρόνου ο χρήστης γίνεται πιο έμπειρος, οπότε θα πρέπει να μετατεθεί σε υψηλότερη κατηγορία εμπειρίας και επομένως να ανατεθεί σε αυτόν διαφορετικό σύνολο περιορισμών.

Όπως αναφέρθηκε ήδη, η αναπαράσταση των περιορισμών μπορεί να κριθεί ως πλήρης και ορθή σε σχέση με το Σύστημα Σφαιρών του Grove. Έχει γίνει προφανές πώς μπορεί να προκύψει ένα Σύστημα Σφαιρών από ένα σύνολο περιορισμών. Με παρόμοιο τρόπο, από ένα Σύστημα Σφαιρών που γνωρίζουμε εκ των προτέρων μπορεί να προκύψει ένα σύνολο από περιορισμούς. Αυτό μπορεί να επιτευχθεί εξετάζοντας τις θέσεις των πιθανών γνωστικών καταστάσεων στο Σύστημα Σφαιρών και τον τρόπο με τον οποίο έχουν σχεδιαστεί οι διάφορες σφαίρες γύρω από αυτές. Όταν για παράδειγμα η γνωστική κατάσταση $BS2 = \{A_1=y_1, A_2=x_2, A_3=x_3, \dots\}$ δεν είναι τοποθετημένη μέσα στην πρώτη σφαίρα γύρω από τη γνωστική κατάσταση $BS1 = \{A_1=x_1, A_2=x_2, A_3=x_3, \dots\}$, όπου $|x_1 - y_1| = 1$, αυτό σημαίνει ότι υπάρχει ένας περιορισμός που βασίζεται στο κριτήριο A_1 όταν αλλάζει η τιμή του από x_1 σε y_1 . Οι ακριβείς περιορισμοί μπορούν να βρεθούν με το ταίριασμα όλων αυτών των παραδειγμάτων.

6.5 Συμπεράσματα

Χρησιμοποιώντας το προτεινόμενο μοντέλο μία εταιρεία παραγωγής λογισμικού μπορεί να εκτιμήσει τη γνώμη των χρηστών για την ποιότητα των προϊόντων της, καθώς και τις αναθεωρήσεις στη γνώμη τους με την πάροδο του χρόνου,

αποφεύγοντας έτσι τη διεξαγωγή επαναληπτικών μετρήσεων που έχουν μεγάλο κόστος. Παρόλα αυτά, μετά από ένα εύλογο χρονικό διάστημα, το οποίο εξαρτάται από το μέγεθος του λογισμικού, το πλήθος των λειτουργιών που υποστηρίζει, το επίπεδο εμπειρίας των χρηστών και τη συχνότητα χρήσης, προτείνεται να διεξαχθεί μία νέα μέτρηση ώστε να ελεγχθεί η ορθότητα του μοντέλου, ενίοτε να ρυθμιστεί ο τρόπος που εφαρμόζεται και κυρίως να ελεγχθεί και πιθανώς να προσαρμοστεί το επίπεδο εμπειρίας στο οποίο ανήκουν οι χρήστες.

Οι εταιρείες παραγωγής λογισμικού που θα χρησιμοποιήσουν το προτεινόμενο μοντέλο θα πρέπει να το προσαρμόσουν στις δικές τους ιδιαίτερες ανάγκες. Τα όρια των γεγονότων που θα χρησιμοποιηθούν στο μοντέλο θα πρέπει να σχεδιαστούν σύμφωνα με το βάρος που θα δοθεί σε κάθε ποιοτικό χαρακτηριστικό του λογισμικού. Με άλλα λόγια, διαφορετικά βάρη μπορούν να δοθούν στα διάφορα ποιοτικά κριτήρια του λογισμικού, ανάλογα με τα συγκεκριμένα ποιοτικά χαρακτηριστικά στα οποία μία εταιρεία παραγωγής λογισμικού μπορεί να εστιάσει περισσότερο ή όχι. Σε αυτή την περίπτωση, τόσο η ταξινόμηση των πιθανών γνωστικών καταστάσεων όσο και ο καθορισμός των αποστάσεων ανάμεσά τους θα πρέπει να λαμβάνουν υπόψη τα βάρη αυτά.

Επεκτείνοντας το μοντέλο ώστε να περιέχει γεγονότα για κάθε χαρακτηριστικό και υπο-χαρακτηριστικό της ποιότητας λογισμικού, οι εταιρείες παραγωγής λογισμικού μπορούν να εκτιμήσουν όχι μόνο την άποψη των χρηστών συνολικά για την ποιότητα των προϊόντων τους, αλλά και την άποψή τους για κάθε ποιοτικό χαρακτηριστικό ξεχωριστά.

Κεφάλαιο**7****Χρήση Αρχών από τη θεωρία Αναθεώρησης
Αποψης**

Στο κεφάλαιο αυτό γίνεται χρήση και προσαρμογή των κανόνων από τη θεωρία Αναθεώρησης Αποψης για να διατυπωθούν με περισσότερο φορμαλιστικό τρόπο οι αναθεωρήσεις των απόψεων των χρηστών για την ποιότητα ενός προϊόντος λογισμικού με την πάροδο του χρόνου. Αρχικά, γίνεται μία εισαγωγή στη θεωρία Αναθεώρησης Αποψης και τα αξιώματα που τη διέπουν, σύμφωνα με το AGM παράδειγμα. Επίσης, παρουσιάζεται το Σύστημα Σφαιρών του Grove, το οποίο χρησιμοποιείται για τις ανάγκες της παρούσας διατριβής. Στη συνέχεια, επαναδιατυπώνεται το πλαίσιο αναθεωρήσεων της γνώμης των χρηστών για την ποιότητα λογισμικού χρησιμοποιώντας κατάλληλη σημειογραφία, ενώ παράλληλα προσδιορίζονται μέσω μαθηματικών εκφράσεων οι περιορισμοί ανάμεσα στους διάφορους παράγοντες της ποιότητας. Αποδεικνύεται βάσει αυτής της σημειογραφίας ότι η αναθεώρηση της άποψης των χρηστών για την ποιότητα λογισμικού αποτελεί μία AGM συνάρτηση αναθεώρησης άποψης. Τέλος, γίνονται μελέτες σχετικά με την υπολογιστική πολυπλοκότητα του προτεινόμενου φορμαλισμού και αναφέρονται τα αποτελέσματά τους.

7.1 Εισαγωγή στην AGM θεωρία Αναθεώρησης Αποψης

Στην ενότητα αυτή γίνεται μία σύντομη παρουσίαση των βασικών ιδεών και αποτελεσμάτων από την περιοχή της Αναθεώρησης Αποψης και ιδιαίτερα αυτών που σχετίζονται με το AGM παράδειγμα [ALC85]. Αρχικά θα πρέπει να γίνει μία εισαγωγή στην παρουσίαση των εννοιών και της ορολογίας που χρησιμοποιείται.

Το AGM παράδειγμα βασίζεται σε μία τυπική λογική γλώσσα L , η οποία για τους λόγους αυτής της διδακτορικής διατριβής θα θεωρείται ότι είναι προτασιακή. Αυτό δεν είναι απαραίτητο από το AGM παράδειγμα, αλλά διευκολύνει την παρουσίασή του στα πλαίσια της παρούσας διατριβής. Μία πλήρης περιγραφή των ελάχιστων απαιτήσεων για τη γλώσσα L δίνει ο Gardenfors [GAR88a]. Για ένα σύνολο προτάσεων Γ της L , ορίζουμε ως $Cn(\Gamma)$ το σύνολο όλων των λογικών συμπερασμάτων του Γ , δηλαδή $Cn(\Gamma) = \{\phi \in L : \Gamma \vdash \phi\}$. Μία θεωρία K της L είναι κάθε σύνολο προτάσεων της L που είναι κλειστό ως προς τη σχέση $\vdash$, δηλαδή ισχύει

$K = Cn(K)$. Ορίζουμε τα σύνολα όλων των θεωριών της L ως T_L . Μία θεωρία K της L είναι πλήρης αν και μόνο αν για όλες τις προτάσεις $\phi \in L$ ισχύει είτε $\phi \in K$ είτε $\neg\phi \in K$. Επίσης, ορίζουμε τα σύνολα όλων των θεωριών πλήρους συνέπειας (consistent complete) της L ως M_L . Στη θεωρία της Αναθεώρησης Άποψης, οι θεωρίες πλήρους συνέπειας έχουν συχνά το ρόλο των *πιθανών κόσμων*. Ακολουθώντας αυτή τη σύμβαση, στη συνέχεια θα χρησιμοποιούμε εναλλακτικά τους όρους “πιθανός κόσμος” (ή απλούστερα “κόσμος”) και “θεωρία πλήρους συνέπειας” (“consistent complete theory”). Για ένα σύνολο προτάσεων Γ της L , ως $[\Gamma]$ δηλώνεται το σύνολο όλων των θεωριών πλήρους συνέπειας της L που περιλαμβάνουν το Γ . Συχνά χρησιμοποιούμε τον όρο $[\phi]$ για μία πρόταση $\phi \in L$, ως συντομογραφία του $\{\phi\}$. Για μία θεωρία K και ένα σύνολο προτάσεων Γ της L , ορίζουμε ως $K + \Gamma$ το κλείσιμο κάτω από τον τελεστή $\vdash$ του $K \cup \Gamma$, δηλαδή $K + \Gamma = Cn(K \cup \Gamma)$. Για μία πρόταση $\phi \in L$ θα χρησιμοποιούμε συνήθως το $K + \phi$ ως συντομογραφία του $K + \{\phi\}$.

Έχοντας καθορίσει λοιπόν κάποια σημειογραφία, μπορούμε πλέον να παρουσιάσουμε εν συντομία τους βασικούς προσδιορισμούς και αποτελέσματα από την προσέγγιση του AGM στην αναθεώρηση άποψης. Στο περιβάλλον που προτείνουν οι Alchourron, Gardenfors και Makinson [ALC85] αναπαριστούν τις γνωστικές καταστάσεις ως θεωρίες της γλώσσας L και η διαδικασία της αναθεώρησης άποψης μοντελοποιείται ως μία ειδική συνάρτηση $*$ πάνω στις θεωρίες, η οποία ονομάζεται *συνάρτηση αναθεώρησης*. Για την ακρίβεια, η συνάρτηση αναθεώρησης $*$ ορίζεται ως συνάρτηση από το $T_L \times L$ στο T_L , αντιστοιχώντας το $\langle K, \phi \rangle$ στο $K * \phi$, η οποία ικανοποιεί τα παρακάτω αξιώματα:

- (K.1) Το $K * \phi$ είναι μία θεωρία της L
- (K.2) $\phi \in K * \phi$
- (K.3) $K * \phi \subseteq K + \phi$
- (K.4) Αν $\neg\phi \notin K$ τότε $K + \phi \subseteq K * \phi$
- (K.5) $K * \phi = L$ αν και μόνο αν $\vdash \neg\phi$
- (K.6) Αν $\vdash \phi \leftrightarrow \psi$ τότε $K * \phi = K * \psi$
- (K.7) $K * (\phi \wedge \psi) \subseteq (K * \phi) + \psi$
- (K.8) Αν $\neg\psi \notin K * \phi$ τότε $(K * \phi) + \psi \subseteq K * (\phi \wedge \psi)$

Το παραπάνω σύνολο από αξιώματα ακολουθεί και καθορίζεται κυρίως από την *αρχή των ελάχιστων αλλαγών*, σύμφωνα με την οποία θα πρέπει να συμβούν όσο το δυνατόν λιγότερες αλλαγές σε μία γνωστική κατάσταση K όταν αποδέχεται και εμπλουτίζεται με μία νέα πληροφορία ϕ . Ο Gardenfors παρουσιάζει μία λεπτομερή ανάλυση σχετικά με την αρχή των ελάχιστων αλλαγών και τη σχέση της με τα αξιώματα (K.1) έως (K.8) [GAR88a].

Εκτός από την παραπάνω προσέγγιση μέσω των αξιωμάτων στη θεωρία Αναθεώρησης Άποψης, ένας αριθμός από σαφείς και ολοκληρωμένες προτάσεις και μοντέλα έχει προταθεί για αυτή τη διαδικασία [ALC85], [GAR88b], [PEP95]. Στην παρούσα διατριβή γίνεται εξέταση και χρήση μίας από αυτές και συγκεκριμένα το Σύστημα Σφαιρών του Grove [GRO88].

Έστω ότι το A είναι ένα υποσύνολο του M_L (δηλαδή το A είναι ένα σύνολο από θεωρίες πλήρους συνοχής). Ένα σύστημα σφαιρών S με κέντρο το A είναι μία συλλογή από υποσύνολα του M_L , τα στοιχεία του οποίου καλούνται *σφαίρες*, που ικανοποιούν τις παρακάτω συνθήκες:

- (S.1) Το S είναι πλήρως διατεταγμένο σε σχέση με το συνυπολογισμό των συνόλων, δηλαδή αν $U, U' \in S$ τότε $U \subseteq U'$ ή $U' \subseteq U$
- (S.2) Η μικρότερη σφαίρα στο S είναι το A , δηλαδή ισχύει $A \in S$ και αν $U' \in S$ τότε $A \subseteq U'$
- (S.3) $M_L \in S$ και επομένως το M_L είναι η μεγαλύτερη σφαίρα στο S
- (S.4) Για κάθε $\varphi \in L$ αν υπάρχει σφαίρα στο S που τέμνεται με το $[\varphi]$, τότε υπάρχει επίσης μία ελάχιστη σφαίρα στο S η οποία τέμνεται με το $[\varphi]$

Για ένα σύστημα σφαιρών S και μία συνεπή (consistent) πρόταση $\varphi \in L$ η μικρότερη σφαίρα στο S που τέμνεται με το $[\varphi]$ δηλώνεται ως $C_S(\varphi)$. Όταν η φ είναι ασυνεπής (inconsistent), το $C_S(\varphi)$ είναι το ίδιο το M_L . Με κάθε σύστημα σφαιρών S ο Grove συνδέει μία συνάρτηση $f_S : L \mapsto 2^{M_L}$ που δηλώνεται ως εξής:

$$f_S(\varphi) = [\varphi] \cap C_S(\varphi), \text{ για κάθε } \varphi \in L.$$

Ας θεωρήσουμε τώρα μία θεωρία K του L και έστω S ότι είναι το σύστημα σφαιρών με κέντρο το $[K]$. Ο Grove χρησιμοποιεί το S για να ορίσει με κατασκευαστικό τρόπο τη διαδικασία αναθεώρησης του K με τη βοήθεια της παρακάτω συνθήκης:

$$(S^*) \quad K * \varphi = \begin{cases} \cap f_S(\varphi) & \text{if } [\varphi] \neq \emptyset \\ L & \text{otherwise} \end{cases}$$

Ας μελετήσουμε εν συντομία τη διαίσθηση πίσω από ένα Σύστημα Σφαιρών θεωρώντας το ως βάση για την κατασκευή συναρτήσεων αναθεώρησης. Έστω ότι το K είναι μία συνεπής θεωρία του L που λαμβάνεται ως μία γνωστική κατάσταση. Μία θεωρία πλήρους συνέπειας του L θα θεωρείται ως ένας *πιθανός κόσμος* και ένα σύστημα σφαιρών S με κέντρο το $[K]$ θα θεωρείται ως μία διάταξη από πιθανούς κόσμους, που αναπαριστά τη συγκριτική τους αληθοφάνεια, δεδομένου ότι το K είναι η τρέχουσα γνωστική κατάσταση. Όσο πιο κοντά βρίσκεται ένας πιθανός κόσμος στο κέντρο του S , τόσο πιο αληθοφανής είναι. Επομένως, η συνθήκη (S^*) ορίζει ότι η αναθεώρηση του K μέσω της φ είναι η θεωρία που προσδιορίζεται από τους πιο αληθοφανείς κόσμους που ικανοποιούν την φ .

Ο Grove παρουσιάζει επίσης ότι η κλάση των συναρτήσεων που παράγονται από ένα σύστημα σφαιρών με τη βοήθεια του (S^*) είναι ακριβώς η ομάδα των AGM συναρτήσεων αναθεώρησης, δηλαδή οι συναρτήσεις που ικανοποιούν τα AGM αξιώματα. Η κατάληξη αυτή του Grove αποτελεί το επίκεντρο της ανάλυσης που επακολουθεί και μέσω αυτής της κατάληξης θα αποδειχτεί στην ενότητα 7.3 η σύνδεση ανάμεσα στην AGM θεωρία Αναθεώρησης Αποψης και τη δυναμική και τις μεταβολές της γνώμης των χρηστών για την ποιότητα λογισμικού.

7.2 Αναθεωρήσεις της γνώμης των χρηστών για την ποιότητα λογισμικού

Στην προηγούμενη ενότητα παρουσιάστηκε το AGM παράδειγμα ως ένα γενικό περιβάλλον για την εισαγωγή των εννοιών και τη μελέτη της διαδικασίας της αναθεώρησης άποψης. Ωστόσο, σε αυτή τη διδακτορική διατριβή αυτό που μας ενδιαφέρει είναι η συγκεκριμένη διαδικασία των μεταβολών των απόψεων για την ποιότητα των προϊόντων λογισμικού. Για το λόγο αυτό σε αυτή την ενότητα γίνεται η παρουσίαση ενός νέου σχηματισμού που μορφοποιήθηκε ειδικά για τις αναθεωρήσεις των παραπάνω απόψεων. Στην ενότητα 7.3 θα γίνει μία σύγκριση ανάμεσα σε αυτή την προσέγγιση και την AGM θεωρία Αναθεώρησης Άποψης.

Έστω ότι $A_1, A_2, \dots, A_n$ είναι οι παράγοντες που αντιπροσωπεύουν τη γνώμη του χρήστη για ένα προϊόν λογισμικού (για παράδειγμα, το A_1 μπορεί να αντιπροσωπεύει τη λειτουργικότητα, το A_2 την αξιοπιστία, κτλ.). Επιπλέον, ας υποθέσουμε ότι οι τιμές που παίρνουν αυτοί οι παράγοντες είναι ακέραιοι αριθμοί που ανήκουν στην κλίμακα $[0,10]$.

Ορίζουμε ως *βαθμολογία λογισμικού* (ή απλούστερα *βαθμολογία*) το διάνυσμα $\bar{u} = \langle r_1, r_2, \dots, r_n \rangle$, όπου οι τιμές των r_i ανήκουν στο διάστημα $[0,10]$. Ο σκοπός του διανύσματος $\bar{u} = \langle r_1, r_2, \dots, r_n \rangle$ είναι ότι αναθέτει μία τιμή σε καθένα από τους παράγοντες $A_1, A_2, \dots, A_n$ (δηλαδή, $A_1=r_1, A_2=r_2$, κτλ.). Θα δηλώνουμε την τιμή που το διάνυσμα $\bar{u}$ αναθέτει στον παράγοντα A_j ως $\bar{u}(j)$ (δηλαδή $\bar{u}(j)=r_j$). Για την ακρίβεια, υπάρχουν 11ⁿ διαφορετικές βαθμολογίες, αφού ο βαθμός για τον κάθε παράγοντα παίρνει τιμές από 0 έως 10. Επίσης, ορίζουμε ως S το σύνολο όλων αυτών των βαθμολογιών.

Υποθέτουμε τώρα ότι η αρχική γνώμη ενός χρήστη για την ποιότητα ενός συγκεκριμένου προϊόντος λογισμικού δίνεται από τη βαθμολογία $\bar{u} = \langle r_1, r_2, \dots, r_n \rangle$. Επίσης, υποθέτουμε ότι, μετά από επιπλέον χρήση και αλληλεπίδραση με το λογισμικό, ο χρήστης αυτός αλλάζει τη γνώμη του για την τιμή του παράγοντα A_j και, αντί για r_j , τώρα εκτιμά ότι ο παράγοντας A_j πρέπει να πάρει την τιμή p_j . Οπότε, η αρχή των ελάχιστων αλλαγών ορίζει ότι η νέα γνώμη του χρήστη για την ποιότητα του προϊόντος λογισμικού θα πρέπει να αναπαρίσταται από τη βαθμολογία $\bar{u}' = \langle r_1, \dots, r_{j-1}, p_j, r_{j+1}, \dots, r_n \rangle$, η οποία διαφέρει από την αρχική βαθμολογία $\bar{u}$ μόνο στην τιμή του A_j .

Πιο επίσημα, θα καλούμε ένα διατεταγμένο ζεύγος $\langle A_j, p_j \rangle$, το οποίο αποτελείται από τον παράγοντα A_j και μία (νέα) τιμή p_j στο διάστημα $[0,10]$, ως ένα *ζεύγος ανάθεσης* (assignment pair). Δοθέντος ενός ζεύγους ανάθεσης $\langle A_j, p_j \rangle$ και μία αρχικής βαθμολογίας $\bar{u}$, ως $\bar{u} \circ \langle A_j, p_j \rangle$ θα δηλώνουμε την *αναθεώρηση της βαθμολογίας* του $\bar{u}$ μέσω του $\langle A_j, p_j \rangle$. Επομένως, σύμφωνα με τη σημειογραφία αυτή, στο παραπάνω παράδειγμα θα ισχύει $\bar{u} \circ \langle A_j, p_j \rangle = \langle r_1, \dots, r_{j-1}, p_j, r_{j+1}, \dots, r_n \rangle$.

Ο προσδιορισμός της αναθεώρησης βαθμολογίας για περιπτώσεις όπως η προηγούμενη είναι άμεσος και απλός. Τα πράγματα όμως γίνονται περισσότερο πολύπλοκα όταν εμφανίζονται *εξαρτήσεις* ανάμεσα στους διάφορους παράγοντες της ποιότητας. Έστω μάλιστα ότι οι εξαρτήσεις αυτές έχουν την εξής μορφή

$A_i \in [h_1, h_2] \Rightarrow A_j \in [c_1, c_2]$, όπου τα $[h_1, h_2]$ και $[c_1, c_2]$ είναι υποδιαστήματα του $[0, 10]$. Θα λέμε λοιπόν ότι μία βαθμολογία $\bar{u}$ ικανοποιεί την παραπάνω φόρμουλα, η οποία από εδώ και στο εξής θα καλείται *περιορισμός*, αν και μόνο αν το $h_1 \leq \bar{u}(i) \leq h_2$ συνεπάγεται ότι $c_1 \leq \bar{u}(j) \leq c_2$. Δηλαδή, αν η τιμή του A_i ανήκει στο διάστημα $[h_1, h_2]$, τότε η τιμή του A_j θα ανήκει στο διάστημα $[c_1, c_2]$.

Υποθέτουμε ότι ένα σύνολο C από περιορισμούς αυτού του είδους δίνεται εκ των προτέρων και ότι εξαρτάται από διάφορους παράγοντες που έχουν ήδη αναφερθεί στα προηγούμενα κεφάλαια, όπως τα ιδιαίτερα χαρακτηριστικά του προϊόντος λογισμικού που εξετάζεται, τα χαρακτηριστικά και η εμπειρία του χρήστη ή της ομάδας χρηστών στην οποία αναφερόμαστε κ.ά. Επομένως, μία βαθμολογία $\bar{u}$ θα είναι έγκυρη αν και μόνο αν ικανοποιεί όλους τους περιορισμούς του C . Θα δηλώνουμε ως S_C το σύνολο όλων των έγκυρων βαθμολογιών.

Ας επανεξετάσουμε λοιπόν τη διαδικασία της αναθεώρησης της βαθμολογίας λαμβάνοντας υπόψη την ύπαρξη περιορισμών. Υποθέτουμε, χωρίς βλάβη της γενικότητας, ότι η αρχική βαθμολογία ενός χρήστη είναι $\bar{u} = \langle 0, 0, 0, \dots, 0 \rangle$. Επίσης, υποθέτουμε ότι το σύνολο C αποτελείται από τους ακόλουθους περιορισμούς:

$$(C.1) \quad A_1 \in [2, 5] \Rightarrow A_2 \in [3, 6]$$

$$(C.2) \quad A_2 \in [2, 4] \Rightarrow A_3 \in [4, 7]$$

Ας θεωρήσουμε τώρα ότι η γνώμη του χρήστη για τον παράγοντα A_1 αλλάζει τιμή σε 3. Μεταβάλλοντας μόνο την τιμή του A_1 προκύπτει η βαθμολογία $\bar{u}' = \langle 3, 0, 0, \dots, 0 \rangle$. Το πρόβλημα με την $\bar{u}'$ όμως είναι ότι παραβιάζει τους περιορισμούς του C και επομένως δεν μπορεί να επιλεγεί ως η νέα βαθμολογία του χρήστη. Για την ακρίβεια η $\bar{u}'$ παραβιάζει τον περιορισμό (C.1). Για να ικανοποιηθεί ο (C.1), θα πρέπει να δοθεί στον παράγοντα A_2 μία τιμή από το διάστημα $[3, 6]$. Λόγω της προσπάθειας να ελαχιστοποιούνται οι αλλαγές, θα πρέπει να δώσουμε στον A_2 την τιμή 3. Επομένως προκύπτει η βαθμολογία $\bar{u}'' = \langle 3, 3, 0, \dots, 0 \rangle$. Δυστυχώς, η νέα βαθμολογία $\bar{u}''$ είναι επίσης άκυρη, αφού αυτή τη φορά παραβιάζεται ο περιορισμός (C.2). Για να διορθωθεί αυτό, ενώ θα πρέπει παράλληλα να συντηρούμε τις διάφορες αλλαγές στο ελάχιστο, θα πρέπει να δώσουμε στον παράγοντα A_3 την τιμή 4. Οπότε, προκύπτει η βαθμολογία $\bar{u} = \langle 3, 3, 4, \dots, 0 \rangle$. Αυτή η νέα βαθμολογία $\bar{u}$ είναι έγκυρη (δηλαδή ικανοποιεί όλους τους περιορισμούς) και συνεπώς είναι η αποδεκτή επιλογή βαθμολογίας για τη νέα γνώμη του χρήστη για την ποιότητα του προϊόντος λογισμικού.

Ας εξετάσουμε όμως το ακόλουθο εναλλακτικό σενάριο. Ξεκινώντας από τη μη αποδεκτή βαθμολογία $\bar{u}' = \langle 3, 0, 0, \dots, 0 \rangle$, αυτή τη φορά αλλάζουμε την τιμή του A_2 σε 5, οπότε παράγεται η βαθμολογία $\bar{u}' = \langle 3, 5, 0, \dots, 0 \rangle$. Παρατηρούμε ότι δεν χρειάζονται περαιτέρω αλλαγές στην $\bar{u}'$ αφού για τον περιορισμό (C.2) δεν υφίσταται κάποιο πρόβλημα. Για το λόγο αυτό, η $\bar{u}'$ θα είναι μία έγκυρη βαθμολογία όταν παίρνει τη σωστή τιμή για τον παράγοντα A_1 και επιπλέον, ακολουθώντας μία προκαθορισμένη λογική, θα τηρεί την αρχή των ελάχιστων αλλαγών κατά την αναθεώρηση της αρχικής βαθμολογίας $\bar{u}$.

Εξετάζοντας καλύτερα τις βαθμολογίες $\bar{u}$ και $\bar{u}'$ παρατηρούμε ότι συγκριτικά με την πρώτη, στη δεύτερη έγινε μία αύξηση στον παράγοντα A_2 κατά 2 και μία μείωση στον παράγοντα A_3 κατά 4. Το ερώτημα λοιπόν που προκύπτει είναι αν με αυτές τις μεταβολές μπορεί να χαρακτηριστεί η $\bar{u}'$ ως μικρότερη αλλαγή της $\bar{u}$ από ό,τι η $\bar{u}$. Η απάντηση στο παραπάνω ερώτημα εξαρτάται από το προϊόν λογισμικού που αναφερόμαστε. Δηλαδή, σχετίζεται με τη συγκριτική σπουδαιότητα των παραγόντων A_2 και A_3 σύμφωνα με το χρήστη και το υπό μελέτη λογισμικό. Για την αντιμετώπιση αυτών των αμφιβολιών, υποθέτουμε ότι οι διάφοροι παράγοντες της ποιότητας είναι ταξινομημένοι κατά σειρά σπουδαιότητας. Επομένως, ο A_1 είναι πιο σημαντικός από τον A_2 , ο οποίος είναι πιο σημαντικός από τον A_3 , κ.ο.κ. Αξίζει να σημειωθεί ότι η υπόθεση αυτή δεν είναι τόσο περιοριστική όσο παρουσιάζεται αρχικά, αφού η ονομασία του κάθε A_j δεν είναι σταθερή, αλλά αντιθέτως μπορεί να αλλάζει ανάλογα με το υπό μελέτη προϊόν λογισμικού. Με άλλα λόγια, η σειρά σπουδαιότητας μπορεί να διαφέρει από προϊόν σε προϊόν. Δεδομένης αυτής της υπόθεσης, η βαθμολογία $\bar{u}$ αποτελεί μικρότερη αλλαγή της βαθμολογίας $\bar{u}$ από ό,τι η βαθμολογία $\bar{u}'$ και επομένως λαμβάνεται ως η νέα βαθμολογία του χρήστη.

Γενικεύοντας, έστω ότι $\bar{u}$, $\bar{u}$ και $\bar{z}$ είναι τρεις βαθμολογίες, όπου $\bar{u} \neq \bar{z}$. Επιπλέον, έστω ότι ο A_j είναι ο πρώτος παράγοντας στον οποίο οι βαθμολογίες $\bar{u}$ και $\bar{z}$ έχουν διαφορετικές τιμές. Θα λέμε ότι η $\bar{u}$ είναι πλησιέστερα στην $\bar{u}$ από την $\bar{z}$, το οποίο θα δίνεται από τη σχέση $\bar{u} <_{\bar{v}} \bar{z}$, αν και μόνο αν είτε ισχύει ότι $|\bar{u}(j) - \bar{u}(j)| < |\bar{u}(j) - \bar{z}(j)|$ είτε ισχύουν τα $|\bar{u}(j) - \bar{u}(j)| = |\bar{u}(j) - \bar{z}(j)|$ και $\bar{u}(j) < \bar{z}(j)$. Θα χρησιμοποιούμε τη σχέση $\bar{u} \leq_{\bar{v}} \bar{z}$ για να δηλώσουμε το γεγονός ότι είτε $\bar{u} <_{\bar{v}} \bar{z}$ είτε $\bar{u} = \bar{z}$.

Παρατηρούμε ότι η διμελής σχέση $\leq_{\bar{v}}$ εξαρτάται από την αρχική βαθμολογία $\bar{u}$. Στην πραγματικότητα, υπάρχει μία διαφορετική σχέση $\leq_{\bar{v}}$ για κάθε $\bar{u} \in S$.

Λήμμα 7.2.1 Για κάθε βαθμολογία $\bar{u} \in S$, η διμελής σχέση $\leq_{\bar{v}}$ είναι μία σχέση πλήρους διάταξης.

Απόδειξη. Η ανακλαστικότητα είναι προφανής από τον ίδιο τον ορισμό της σχέσης $\leq_{\bar{v}}$.

Για τη μεταβατικότητα, θεωρούμε τρεις βαθμολογίες $\bar{u}, \bar{z}, \bar{w} \in S$, όπου ισχύει $\bar{u} \leq_{\bar{v}} \bar{z} \leq_{\bar{v}} \bar{w}$. Αν οποιεσδήποτε δύο από αυτές τις τρεις βαθμολογίες είναι μεταξύ τους ίδιες, τότε εύκολα προκύπτει ότι $\bar{u} \leq_{\bar{v}} \bar{w}$. Έστω τώρα ότι έχουμε $\bar{u} \neq \bar{z} \neq \bar{w}$ και επομένως ισχύει $\bar{u} <_{\bar{v}} \bar{z} <_{\bar{v}} \bar{w}$. Έστω ότι ο A_j είναι ο πρώτος παράγοντας στον οποίο οι βαθμολογίες $\bar{u}$ και $\bar{z}$ διαφέρουν. Παρόμοια, έστω ότι ο A_k είναι ο πρώτος παράγοντας στον οποίο οι βαθμολογίες $\bar{z}$ και $\bar{w}$ διαφέρουν. Λόγω του $\bar{u} <_{\bar{v}} \bar{z}$ προκύπτει ότι έχουμε είτε $|\bar{u}(j) - \bar{u}(j)| < |\bar{u}(j) - \bar{z}(j)|$ είτε $|\bar{u}(j) - \bar{u}(j)| = |\bar{u}(j) - \bar{z}(j)|$ και $\bar{u}(j) < \bar{z}(j)$. Επίσης, λόγω του $\bar{z} <_{\bar{v}} \bar{w}$ προκύπτει ότι είτε $|\bar{u}(k) - \bar{z}(k)| < |\bar{u}(k) - \bar{w}(k)|$ είτε $|\bar{u}(k) - \bar{z}(k)| = |\bar{u}(k) - \bar{w}(k)|$ και $\bar{z}(k) < \bar{w}(k)$. Διακρίνουμε τις παρακάτω τρεις περιπτώσεις: $j < k$, $j = k$ και $j > k$. Ξεκινώντας από την πρώτη περίπτωση, δηλαδή ότι $j < k$, προκύπτει ότι ο πρώτος παράγοντας στον οποίο διαφέρουν οι βαθμολογίες $\bar{u}$ και $\bar{w}$ είναι ο A_j . Επιπρόσθετα, ισχύει ότι $\bar{w}(j) = \bar{z}(j)$. Συνδυάζοντας τα παραπάνω,

προκύπτει ότι ισχύει είτε $|\bar{u}(j) - \bar{u}(j)| < |\bar{u}(j) - \bar{w}(j)|$ είτε $|\bar{u}(j) - \bar{u}(j)| = |\bar{u}(j) - \bar{w}(j)|$ και $\bar{u}(j) < \bar{w}(j)$. Επομένως, οδηγούμαστε και στο $\bar{u} \leq_{\bar{v}} \bar{w}$ όπως επιθυμούσαμε. Οι αποδείξεις για τις άλλες δύο περιπτώσεις ($j=k$ και $j>k$) είναι ακριβώς συμμετρικές. Άρα η σχέση $\leq_{\bar{v}}$ είναι μεταβατική.

Για την αντισυμμετρικότητα, έστω ότι έχουμε δύο βαθμολογίες $\bar{u}, \bar{z} \in S$, όπου $\bar{u} \leq_{\bar{v}} \bar{z}$ και $\bar{u} \neq \bar{z}$. Έστω ότι ο A_j είναι ο πρώτος παράγοντας στον οποίο οι βαθμολογίες $\bar{u}$ και $\bar{z}$ διαφέρουν. Τότε ισχύει ότι είτε $|\bar{u}(j) - \bar{u}(j)| < |\bar{u}(j) - \bar{z}(j)|$ είτε $|\bar{u}(j) - \bar{u}(j)| = |\bar{u}(j) - \bar{z}(j)|$ και $\bar{u}(j) < \bar{z}(j)$. Και για τις δύο αυτές περιπτώσεις προκύπτει ότι (i) $|\bar{u}(j) - \bar{z}(j)| \neq |\bar{u}(j) - \bar{u}(j)|$ και (ii) αν $|\bar{u}(j) - \bar{z}(j)| = |\bar{u}(j) - \bar{u}(j)|$ τότε $\bar{z}(j) \neq \bar{u}(j)$. Από αυτό συνεπάγεται ότι $\bar{z} \not\leq_{\bar{v}} \bar{u}$ όπως επιθυμούσαμε.

Τέλος για την πληρότητα, έστω ότι οι $\bar{u}$ και $\bar{z}$ είναι δύο οποιεσδήποτε βαθμολογίες του S . Προφανώς, αν $\bar{u} = \bar{z}$ τότε και $\bar{u} \leq_{\bar{v}} \bar{z}$. Έστω τώρα ότι $\bar{u} \neq \bar{z}$ και έστω ότι ο A_j είναι ο πρώτος παράγοντας στον οποίον οι δύο αυτές βαθμολογίες διαφέρουν. Υποθέτουμε επιπλέον ότι ισχύει $\bar{z} \not\leq_{\bar{v}} \bar{u}$. Από αυτό προκύπτει ότι $|\bar{u}(j) - \bar{z}(j)| \neq |\bar{u}(j) - \bar{u}(j)|$ και αν $|\bar{u}(j) - \bar{z}(j)| = |\bar{u}(j) - \bar{u}(j)|$ τότε $\bar{z}(j) \neq \bar{u}(j)$. Από αυτό ξανά προκύπτει ότι είτε $|\bar{u}(j) - \bar{u}(j)| < |\bar{u}(j) - \bar{z}(j)|$ είτε αν $|\bar{u}(j) - \bar{u}(j)| = |\bar{u}(j) - \bar{z}(j)|$ τότε $\bar{u}(j) < \bar{z}(j)$. Συνεπώς, ισχύει ότι $\bar{u} \leq_{\bar{v}} \bar{z}$ όπως επιθυμούσαμε. ■

Από το Λήμμα 7.2.1 προκύπτει ότι κάθε μη κενό υποσύνολο του S έχει ένα μοναδικό ελάχιστο στοιχείο όσον αφορά τη σχέση $\leq_{\bar{v}}$. Δηλαδή, για κάθε υποσύνολο S' του S , θα λέμε ότι η βαθμολογία $\bar{u}$ είναι ελάχιστη στο S' αν και μόνο αν $\bar{u} \in S'$ και για όλα τα $\bar{z} \in S'$ αν $\bar{z} \leq_{\bar{v}} \bar{u}$, τότε και $\bar{u} \leq_{\bar{v}} \bar{z}$. Θα δηλώνουμε ως $\min(S', \leq_{\bar{v}})$ το σύνολο των ελάχιστων στοιχείων του S' . Το παρακάτω αποτέλεσμα προκύπτει με προφανή τρόπο από το Λήμμα 7.2.1.

Πόρισμα 7.2.1 Για κάθε βαθμολογία $\bar{u}$ και κάθε μη κενό υποσύνολο S' του S , το σύνολο $\min(S', \leq_{\bar{v}})$ είναι μονομελές.

Εξαιτίας του Πορίσματος 7.2.1, μπορούμε πλέον, αντιβαίνοντας την ισχύουσα σημειογραφία, να χρησιμοποιούμε το $\min(S', \leq_{\bar{v}})$ για να δηλώνουμε το ελάχιστο στοιχείο του S' αντί του συνόλου που περιλαμβάνει αυτό το στοιχείο.

Θεωρούμε τώρα ένα ζεύγος ανάθεσης $\langle A_j, p_j \rangle$. Ως $\|\langle A_j, p_j \rangle\|$ θα δηλώνουμε το σύνολο όλων των έγκυρων βαθμολογιών στις οποίες ο παράγοντας A_j έχει την τιμή p_j , δηλαδή ισχύει $\|\langle A_j, p_j \rangle\| = \{\bar{u} \in S_C : \bar{u}(j) = p_j\}$. Όταν ισχύει $\|\langle A_j, p_j \rangle\| \neq \emptyset$, θα λέμε ότι το ζεύγος ανάθεσης $\langle A_j, p_j \rangle$ είναι συνεπές. Στη συνέχεια του κεφαλαίου θα εστιάζουμε μόνο σε συνεπή ζεύγη ανάθεσης.

Ορίζουμε ότι η αναθεώρηση μίας βαθμολογίας $\bar{u}$ που προκύπτει από ένα συνεπές ζεύγος ανάθεσης $\langle A_j, p_j \rangle$ οδηγεί στη βαθμολογία $\bar{u} \circ \langle A_j, p_j \rangle$ η οποία υπολογίζεται ως εξής:

$$\bar{v} \circ \langle A_j, p_j \rangle = \min(\| \langle A_j, p_j \rangle \|, \leq_{\bar{v}})$$

Το σκεπτικό πίσω από τον παραπάνω ορισμό πρέπει να είναι προφανές. Δεδομένης μίας αρχικής βαθμολογίας $\bar{v}$ και ενός ζεύγους ανάθεσης $\langle A_j, p_j \rangle$, η αναθεώρηση της $\bar{v}$ μέσω του $\langle A_j, p_j \rangle$ ορίζεται ως η βαθμολογία η οποία διαφέρει το ελάχιστο από την $\bar{v}$, ανάμεσα σε όλες τις έγκυρες βαθμολογίες όπου ο παράγοντας A_j έχει την επιθυμητή τιμή p_j .

Όπως ήδη αναφέρθηκε, ενδιαφερόμαστε μόνο στις αναθεωρήσεις βαθμολογίας που προκύπτουν από συνεπή ζεύγη ανάθεσης. Παρόλα αυτά, για λόγους σωστής μέτρησης ορίζουμε ότι η αναθεώρηση της βαθμολογίας $\bar{v}$ που προκύπτει από ένα μη συνεπές ζεύγος ανάθεσης $\langle A_j, p_j \rangle$ οδηγεί στη βαθμολογία που δίνεται από τη σχέση $\bar{v} \circ \langle A_j, p_j \rangle = \langle \underbrace{0, \dots, 0}_{j-1}, p_j, \underbrace{0, \dots, 0}_{n-j} \rangle$. Παρατηρούμε ότι αφού ισχύει $\| \langle A_j, p_j \rangle \| \neq \emptyset$, τότε η βαθμολογία $\langle \underbrace{0, \dots, 0}_{j-1}, p_j, \underbrace{0, \dots, 0}_{n-j} \rangle$ θα είναι πάντοτε άκυρη.

7.3 Σχέση της αναθεώρησης βαθμολογίας με την AGM αναθεώρηση άποψης

Ο επόμενος στόχος μας είναι να σχετίσουμε την αναθεώρηση βαθμολογίας με την AGM αναθεώρηση άποψης, όπως αυτή διατυπώθηκε στις προηγούμενες ενότητες. Για να συγκριθούν οι δύο αυτές διαδικασίες θα πρέπει πρώτα να επαναδιατυπώσουμε την αναθεώρηση βαθμολογίας μέσα από ένα λογικό σχήμα και στη συνέχεια να την εξετάσουμε σε σχέση με τα AGM αξιώματα.

Αρχικά, ορίζουμε μία προτασιακή γλώσσα L_S που περιλαμβάνει $11n$ προτασιακές μεταβλητές οι οποίες δηλώνονται ως $q[i, j]$, όπου τα i, j είναι ακέραιοι με $1 \leq i \leq n$ και $0 \leq j \leq 10$. Η πραγματική έννοια του $q[i, j]$ είναι ότι ο παράγοντας A_i έχει την τιμή j .

Για να ενισχύσουμε την έννοια αυτή, θα πρέπει να εισάγουμε τους ακόλουθους ισχυρισμούς:

$$(M.1) \quad q[i, 0] \vee q[i, 1] \vee q[i, 2] \vee \dots \vee q[i, 10], \text{ όπου } 1 \leq i \leq n.$$

$$(M.2) \quad q[i, j] \rightarrow \neg q[i, k], \text{ όπου } 1 \leq i \leq n \text{ και } 0 \leq j \neq k \leq 10.$$

Η ακόλουθη ομάδα ισχυρισμών στην πραγματικότητα μεταγλωττίζει τους περιορισμούς του συνόλου C σε ένα λογικό σχήμα. Συγκεκριμένα, έστω ότι το $A_i \in [h_1, h_2] \Rightarrow A_j \in [c_1, c_2]$ είναι ένας περιορισμός στο C . Τότε εισάγεται ο ακόλουθος ισχυρισμός:

$$(M.3) \quad q[i, h_1] \vee q[i, h_1 + 1] \vee \dots \vee q[i, h_2] \rightarrow q[j, c_1] \vee q[j, c_1 + 1] \vee \dots \vee q[j, c_2].$$

Τέλος, σε κάθε βαθμολογία $\bar{v} = \langle r_1, r_2, \dots, r_n \rangle$ αναθέτουμε τον ισχυρισμό:

$$(M.4) \quad q[1, r_1] \wedge q[2, r_2] \wedge \dots \wedge p[n, r_n].$$

Για μία βαθμολογία $\bar{v} = \langle r_1, r_2, \dots, r_n \rangle$, ως $M(\bar{v})$ θα δηλώνουμε το λογικό κλείσιμο των προτάσεων (M.1) έως (M.4). Δηλαδή, $M(\bar{v}) = \text{Cn}\{(M.1), (M.2), (M.3), (M.4)\}$.

Επιπλέον, για ένα ζεύγος ανάθεσης $\langle A_j, p_j \rangle$, ως $M(\langle A_j, p_j \rangle)$ θα δηλώνουμε την προτασιακή μεταβλητή $q[j, p_j]$. Δηλαδή, $M(\langle A_j, p_j \rangle) = q[j, p_j]$.

Χρησιμοποιώντας τους παραπάνω ορισμούς, μπορούμε πλέον να αποδείξουμε ότι η αναθεώρηση βαθμολογίας είναι στην πραγματικότητα μία AGM συνάρτηση αναθεώρησης άποψης.

Θεώρημα 5.3.1 Υπάρχει μία συνάρτηση αναθεώρησης $*$, που ικανοποιεί τα AGM αξιώματα (K.1) έως (K.8), έτσι ώστε, για κάθε βαθμολογία $\bar{u}$ και οποιοδήποτε συνεπές ζεύγος ανάθεσης $\langle A_j, p_j \rangle$, να ισχύει $M(\bar{u}) * M(\langle A_j, p_j \rangle) = M(\bar{u} \circ \langle A_j, p_j \rangle)$.

Απόδειξη. Αποδεικνύουμε την ύπαρξη της $*$ κατασκευάζοντας για κάθε βαθμολογία $\bar{u}$ ένα ανάλογο σύστημα σφαιρών S με κέντρο το $[M(\bar{u})]$.

Θεωρούμε ότι η βαθμολογία $\bar{u}$ αναπαριστά την αρχική γνωστική κατάσταση ενός χρήστη. Αρχικά, θα πρέπει να παρατηρήσουμε ότι η θεωρία που αντιστοιχεί στην $\bar{u}$ μέσω της αντιστοίχισης M είναι μία θεωρία πλήρους συνέπειας. Βασίζόμενοι σε αυτή την παρατήρηση, κατασκευάζουμε για κάθε έγκυρη βαθμολογία $\bar{w}$ μία σφαίρα $U_{\bar{w}}$ που ορίζεται ως εξής:

$$U_{\bar{w}} = \{M(\bar{z}) : \bar{z} \in S_C \text{ και } \bar{z} \leq_{\bar{u}} \bar{w}\}$$

Στη συνέχεια, ορίζουμε ότι η $U^\#$ είναι η σφαίρα που περιλαμβάνει όλες τις θεωρίες πλήρους συνέπειας της γλώσσας L_S .

Τέλος, ορίζουμε ότι το S είναι το σύνολο $S = \{U_{\bar{w}}\}_{\bar{w} \in S_C} \cup \{U^\#\}$. Δεν είναι δύσκολο να επιβεβαιώσουμε ότι το S είναι πράγματι ένα σύστημα σφαιρών με κέντρο το $M(\bar{u})$ (δηλαδή, ικανοποιεί τις συνθήκες (S.1) έως (S.4)) και επιπλέον, για κάθε έγκυρο ζεύγος ανάθεσης $\langle A_j, p_j \rangle$ ισχύει ότι $f_s(M(\langle A_j, p_j \rangle)) = M(\min(\|\langle A_j, p_j \rangle\|, <_{\bar{u}}))$. Συνεπώς, βάσει της κατάληξης του Grove, όπως αυτή διατυπώθηκε στην ενότητα 7.1, η συνάρτηση αναθεώρησης $*$ στο σύστημα σφαιρών S ικανοποιεί τα AGM αξιώματα (K.1) έως (K.8), ενώ για κάθε έγκυρο ζεύγος ανάθεσης $\langle A_j, p_j \rangle$ ισχύει ότι $M(\bar{u}) * M(\langle A_j, p_j \rangle) = M(\bar{u} \circ \langle A_j, p_j \rangle)$. ■

7.4 Μελέτη υπολογιστικής πολυπλοκότητας

Ας εξετάσουμε τώρα τη διαδικασία σχεδιασμού ενός αλγορίθμου για τον υπολογισμό της αναθεώρησης μίας βαθμολογίας $\bar{u}$ μέσω ενός ζεύγους ανάθεσης $\langle A_j, p_j \rangle$ όπως περιγράφηκε στις προηγούμενες ενότητες. Όπως έχει δείξει ο Nebel [NEB91], [NEB94] και οι Eiter και Gottlob [EIT92], οι περισσότεροι από τους γνωστούς τελεστές αναθεώρησης άποψης είναι δισεπίλυτοι και συγκεκριμένα ανήκουν γενικώς στην NP-hard και co-NP-hard κατηγορία προβλημάτων. Επομένως, δεν θα είναι παράδοξο αν και η αναθεώρηση βαθμολογίας, όπως θα αποδειχτεί σε αυτήν την ενότητα, αποτελεί ένα NP-hard πρόβλημα.

Πριν προβούμε στην απόδειξη της παραπάνω πρότασης, θα πρέπει να επαναδιατυπώσουμε την αναθεώρηση βαθμολογίας ως ένα πρόβλημα απόφασης:

Αναθεώρηση Βαθμολογίας:

Περιστατικό: Έστω ότι έχουμε ένα διάνυσμα $\bar{v} = \langle r_1, r_2, \dots, r_n \rangle$, όπου κάθε r_j ανήκει στο διάστημα $[0,10]$, ένα διατεταγμένο ζεύγος $\langle A_j, p_j \rangle$, όπου $1 \leq j \leq n$ και $p_j \in [0,10]$, και ένα σύνολο C από περιορισμούς. Επιπλέον, έχουμε δύο ακέραιους m και k , όπου $0 \leq m \leq n$ και $0 \leq k \leq 10$.

Ερώτηση: Είναι το $\bar{v} \circ \langle A_j, p_j \rangle$ μία έγκυρη βαθμολογία όπου ισχύει $\bar{v} \circ \langle A_j, p_j \rangle(m) = k$;

Το βασικό αποτέλεσμα αυτής της ενότητας παρουσιάζει ότι η αναθεώρηση βαθμολογίας δεν είναι υπολογιστικά ένα εύκολο πρόβλημα:

Θεώρημα 5.4.1 Η αναθεώρηση βαθμολογίας είναι ένα NP-hard πρόβλημα.

Απόδειξη. Η απόδειξη γίνεται με την αναγωγή του 3SAT στην αναθεώρηση βαθμολογίας. Έστω ότι τα $B_1, B_2, \dots, B_t$ είναι προτασιακές μεταβλητές και ότι το $S = \{H_0, H_1, \dots, H_m\}$ είναι ένα τυχαίο περιστατικό του 3SAT πάνω σε αυτές τις προτασιακές μεταβλητές, όπου $H_j = \{\neg B_{j1}, \neg B_{j2}, \neg B_{j3}\}$. Το σύμβολο $\neg$ δηλώνει ότι ο τελεστής άρνησης $\neg$ μπορεί να ισχύει ή όχι.

Εφόσον υπάρχουν t προτασιακές μεταβλητές, θα υπάρχουν $t(t+1)/2$ ζεύγη $\langle B_i, B_j \rangle$ από προτασιακά στοιχεία, όπου $1 \leq i \leq j \leq t$. Για κάθε ζεύγος $\langle B_i, B_j \rangle$ ορίζουμε έναν παράγοντα, τον οποίο θα τον δηλώνουμε ως $A[i,j]$. Ορίζουμε επίσης έναν παράγοντα $E[j]$ για κάθε προτασιακή μεταβλητή B_j . Τέλος, ορίζουμε έναν παράγοντα “ρύθμισης” X . Υποθέτουμε ότι ο X εμφανίζεται πρώτος στη λίστα των παραγόντων και επομένως θα έχει ως δείκτη το 1.

Στη συνέχεια, ορίζουμε ένα σύνολο C από περιορισμούς, που αποτελείται από τρεις ξεχωριστές ομάδες, όπου κάθε μία εξυπηρετεί διαφορετικό σκοπό. Η πρώτη ομάδα περιορίζει τις τιμές όλων των παραγόντων $A[i,j]$ στο διάστημα $[0,3]$ και τις τιμές των παραγόντων $E[i]$ στο διάστημα $[0,1]$:

$$(I.1) \quad A[i,j] \in [0,10] \Rightarrow A[i,j] \in [0,3], \text{ όπου } 1 \leq i \leq j \leq t.$$

$$(I.2) \quad E[i] \in [0,10] \Rightarrow E[i] \in [0,1] \text{ όπου } 1 \leq i \leq t.$$

Όπως έχει ήδη αναφερθεί, ο παράγοντας $E[j]$ αντιστοιχεί στην προτασιακή μεταβλητή B_j και ο παράγοντας $A[i,j]$ αντιστοιχεί στο διατεταγμένο ζεύγος από προτασιακές μεταβλητές $\langle B_i, B_j \rangle$. Επιπλέον, υποθέτουμε ότι η τιμή του παράγοντα $A[i,j]$ καθορίζει τις τιμές των παραγόντων $E[i]$ και $E[j]$ ως εξής: (i) αν $A[i,j]=0$ ισχύει ότι $E[i]=E[j]=0$, (ii) αν $A[i,j]=1$ ισχύει ότι $E[i]=0$ και $E[j]=1$, (iii) αν $A[i,j]=2$ ισχύει ότι $E[i]=1$ και $E[j]=0$ και (iv) αν $A[i,j]=3$ ισχύει ότι $E[i]=E[j]=1$.

Για να ενισχύσουμε την παραπάνω σχέση ανάμεσα στον $A[i,j]$ και τους $E[i], E[j]$, εισάγουμε τη δεύτερη ομάδα από περιορισμούς ως εξής:

$$(II.1) \quad A[i,j] \in [0,1] \Rightarrow E[i] \in [0,0], \text{ όπου } 1 \leq i \leq j \leq t.$$

$$(II.2) \quad A[i,j] \in [2,3] \Rightarrow E[i] \in [1,1], \text{ όπου } 1 \leq i \leq j \leq t.$$

$$(II.3) \quad A[i,j] \in [0,0] \Rightarrow E[j] \in [0,0], \text{ όπου } 1 \leq i \leq j \leq t.$$

$$(II.4) \quad A[i,j] \in [1,1] \Rightarrow E[j] \in [1,1], \text{ όπου } 1 \leq i \leq j \leq t.$$

(II.5) $A[i, j] \in [2, 2] \Rightarrow E[j] \in [0, 0]$, όπου $1 \leq i \leq j \leq t$.

(II.6) $A[i, j] \in [3, 3] \Rightarrow E[j] \in [1, 1]$, όπου $1 \leq i \leq j \leq t$.

Η τρίτη ομάδα από περιορισμούς εγγυάται ότι το περιστατικό S ικανοποιείται αν και μόνο αν υπάρχει τουλάχιστον μία έγκυρη βαθμολογία. Συγκεκριμένα, για κάθε πρόταση $H_j = \{\neg B_{j_1}, \neg B_{j_2}, \neg B_{j_3}\}$ στο S (όπου χωρίς βλάβη της γενικότητας μπορούμε να υποθέσουμε ότι οι δείκτες εμφανίζονται σε μη φθίνουσα διάταξη, δηλαδή ισχύει ότι $j_1 \leq j_2 \leq j_3$) εισάγουμε ένα νέο περιορισμό στο C , ο οποίος εξαρτάται από την παρουσία ή την απουσία του τελεστή άρνησης στις προτάσεις, ως εξής:

(III.1) $A[j_1, j_2] \in [0, 0] \Rightarrow E[j_3] \in [1, 1]$, για κάθε πρόταση $H_j = \{B_{j_1}, B_{j_2}, B_{j_3}\}$.

(III.2) $A[j_1, j_2] \in [0, 0] \Rightarrow E[j_3] \in [0, 0]$, για κάθε πρόταση $H_j = \{B_{j_1}, B_{j_2}, \neg B_{j_3}\}$.

(III.3) $A[j_1, j_2] \in [1, 1] \Rightarrow E[j_3] \in [1, 1]$, για κάθε πρόταση $H_j = \{B_{j_1}, \neg B_{j_2}, B_{j_3}\}$.

(III.4) $A[j_1, j_2] \in [1, 1] \Rightarrow E[j_3] \in [0, 0]$, για κάθε πρόταση $H_j = \{B_{j_1}, \neg B_{j_2}, \neg B_{j_3}\}$.

(III.5) $A[j_1, j_2] \in [2, 2] \Rightarrow E[j_3] \in [1, 1]$, για κάθε πρόταση $H_j = \{\neg B_{j_1}, B_{j_2}, B_{j_3}\}$.

(III.6) $A[j_1, j_2] \in [2, 2] \Rightarrow E[j_3] \in [0, 0]$, για κάθε πρόταση $H_j = \{\neg B_{j_1}, B_{j_2}, \neg B_{j_3}\}$.

(III.7) $A[j_1, j_2] \in [3, 3] \Rightarrow E[j_3] \in [1, 1]$, για κάθε πρόταση $H_j = \{\neg B_{j_1}, \neg B_{j_2}, B_{j_3}\}$.

(III.8) $A[j_1, j_2] \in [3, 3] \Rightarrow E[j_3] \in [0, 0]$, για κάθε πρόταση $H_j = \{\neg B_{j_1}, \neg B_{j_2}, \neg B_{j_3}\}$.

Δεν είναι δύσκολο να επιβεβαιώσουμε ότι ο αριθμός των περιορισμών που δηλώνονται από τις τρεις παραπάνω ομάδες είναι πολυωνυμικός με βάση το μέγεθος του S .

Έστω τώρα ότι έχουμε $\bar{v} = \bar{0}$ και θεωρούμε την αναθεώρηση της βαθμολογίας $\bar{v}$ μέσω του ζεύγους ανάθεσης $\langle X, 1 \rangle$. Όπως αναφέρθηκε, ο X είναι ο παράγοντας ρύθμισης ο οποίος παρουσιάστηκε παραπάνω μαζί με τους παράγοντες $A[i, j]$ και $E[i]$. Δεν είναι δύσκολο να επαληθευτεί ότι το αρχικό περιστατικό S στο 3SAT ικανοποιείται αν και μόνο αν υπάρχει τουλάχιστον μία έγκυρη βαθμολογία. Επιπλέον, αν υπάρχει κάποια έγκυρη βαθμολογία, τότε θα υπάρχει και μία στην οποία ο παράγοντας X θα έχει την τιμή 1. Συνεπώς, το αρχικό περιστατικό S στο 3SAT ικανοποιείται αν και μόνο αν η βαθμολογία $\bar{v} \circ \langle X, 1 \rangle$ είναι έγκυρη και $\bar{v} \circ \langle X, 1 \rangle(1) = 1$ (υπενθυμίζουμε ότι ο X έχει τεθεί να είναι ο πρώτος παράγοντας στη λίστα των παραγόντων). Οπότε, θέτοντας $m=1$, $k=1$, $A_j=X$ και $p_j=1$, έχουμε κατασκευάσει την επιθυμητή αναγωγή του 3SAT στην αναθεώρηση βαθμολογίας. ■

Κεφάλαιο**8**

Ανάλυση της προτεινόμενης μεθοδολογίας

Στο κεφάλαιο αυτό γίνεται ανάλυση της μεθοδολογίας της διδακτορικής διατριβής. Η μεθοδολογία αυτή βασίζεται στις μεθόδους που περιγράφηκαν στα προηγούμενα κεφάλαια κάνοντας χρήση των αποτελεσμάτων και των συμπερασμάτων που προέκυψαν. Με άλλα λόγια, γίνεται μία σύνθεση των μεθόδων αυτών σε ακολουθιακά στάδια από την οποία προκύπτει η προτεινόμενη μεθοδολογία. Αναφέρονται οι στόχοι, ο τρόπος χρήσης και τα πλεονεκτήματα της μεθοδολογίας και παρουσιάζονται αναλυτικά οι διάφορες φάσεις που την αποτελούν. Τέλος, περιγράφονται τα συμπεράσματα που προέκυψαν κατά την ανάπτυξη της μεθοδολογίας.

8.1 Σύντομη περιγραφή της μεθοδολογίας

8.1.1 Σκοπός της μεθοδολογίας

Σε μία μεθοδολογία εξασφάλισης ποιότητας λογισμικού βασικό και κύριο αντικείμενο του ενδιαφέροντος θα πρέπει να είναι οι πελάτες και η ικανοποίηση των προσδοκιών τους για τα υπό ανάπτυξη προϊόντα λογισμικού. Καθώς η ποιότητα είναι άμεσα συνδεδεμένη με τις μετρήσεις, σε ένα πρόγραμμα ποιότητας πρωταρχικό ρόλο θα πρέπει να έχει η διεξαγωγή μετρήσεων για την εκτίμηση της γνώμης των τελικών χρηστών για τα προϊόντα. Μάλιστα, η εκτίμηση αυτή είναι επιθυμητό από μία εταιρεία παραγωγής λογισμικού να γίνεται όσο το δυνατόν νωρίτερα, ώστε να προλαμβάνονται πιθανές ανεπιθύμητες καταστάσεις και να διορθώνονται εγκαίρως πιθανές ατέλειες. Μία έγκαιρη εκτίμηση λοιπόν της γνώμης των χρηστών για την ποιότητα μπορεί να οδηγήσει σε βελτιώσεις από τις πρώιμες φάσεις του κύκλου ζωής λογισμικού.

Ο παραπάνω είναι και ο βασικός στόχος της προτεινόμενης μεθοδολογίας, η οποία εστιάζει στην ενεργή συμμετοχή του χρήστη καθόλη τη διαδικασία παραγωγής λογισμικού. Όπως αναφέρθηκε στα προηγούμενα κεφάλαια, χρήστης με την ευρεία έννοια του όρου μπορεί να θεωρηθεί οποιοσδήποτε αποδέκτης ενός παραγόμενου σε όλες τις φάσεις του κύκλου ζωής λογισμικού. Για το λόγο αυτό, η μεθοδολογία της παρούσας διατριβής δεν περιορίζεται μόνο στην τελική παράδοση ενός προϊόντος και στους τελικούς χρήστες, αλλά αντιθέτως λαμβάνει υπόψη της και τις υπόλοιπες κατηγορίες χρηστών, προτείνοντας τη διεξαγωγή μετρήσεων στα ενδιάμεσα στάδια

παραγωγής λογισμικού. Στη μεθοδολογία αυτή δίνεται ιδιαίτερη έμφαση στη μείωση της προσπάθειας που θα απαιτηθεί στη φάση της συντήρησης, καθώς και στη μείωση του κόστους για την εύρεση της γνώμης του χρήστη για την ποιότητα ενός προϊόντος λογισμικού.

Είναι κοινώς αποδεκτό άλλωστε ότι όσο νωρίτερα μία εταιρεία παραγωγής λογισμικού λαμβάνει μία εκτίμηση για την αντίληψη των χρηστών για την ποιότητα λογισμικού τόσο ευκολότερα μπορεί να προβεί σε διορθωτικές κινήσεις και να καθορίσει τη στρατηγική της απέναντι στην αγορά. Στις περισσότερες περιπτώσεις, ανάλογες εκτιμήσεις μπορούν να ληφθούν με τη διεξαγωγή μετρήσεων στους χρήστες.

Η αξία των μετρήσεων είναι ευρέως αναγνωρισμένη για τον υπολογισμό των ποιοτικών χαρακτηριστικών ενός προϊόντος όπως αυτά αντιλαμβάνονται από τους χρήστες. Όπως αναφέρει ο Kaplan [KAP95], οι μετρήσεις είναι μέθοδοι που μπορούν να ποσοτικοποιηθούν και επομένως όχι μόνο μπορούν να χρησιμοποιηθούν απλά ως ενδείξεις, αλλά επίσης επιτρέπουν την εφαρμογή πιο πολύπλοκων και ουσιωδών τεχνικών ανάλυσης. οι οποίες είναι κατάλληλες για οργανισμούς με υψηλότερα επίπεδα ωριμότητας όσον αφορά τη διασφάλιση ποιότητας. Από την άλλη πλευρά όμως, η διεξαγωγή μετρήσεων είναι μία διαδικασία που μπορεί να αποβεί αρκετά δαπανηρή. Επίσης, τα αποτελέσματά των μετρήσεων μπορεί να αλλάξουν ακόμα και κατά τη διάρκεια της φάσης ανάλυσής τους, αφού, όπως αναφέρθηκε στα προηγούμενα κεφάλαια, η γνώμη των χρηστών για την ποιότητα συνήθως αναθεωρείται με την πάροδο του χρόνου.

Όπως αποκάλυψε η διεξαγωγή επαναληπτικών μετρήσεων που έγιναν κατά τη διάρκεια της διδακτορικής διατριβής, η βαρύτητα της γνώμης των πελατών για την ποιότητα ενός προϊόντος λογισμικού αυξάνει με την πάροδο του χρόνου. Επομένως, θα πρέπει να λαμβάνεται υπόψη κάθε νέο χαρακτηριστικό του προϊόντος που ανακαλύπτει ο πελάτης, το οποίο θα οδηγήσει αναμφισβήτητα σε αναθεώρηση της άποψής τους για το προϊόν. Επίσης, με κάθε νέο χαρακτηριστικό που ανακαλύπτεται, οι χρήστες αποκτούν μεγαλύτερη αυτοπεποίθηση για την εγκυρότητα της γνώμης τους. Η προτεινόμενη μεθοδολογία λοιπόν παρέχει έναν απλούστερο, ταχύτερο και αυτοματοποιήσιμο τρόπο για την εκτίμηση της αναθεώρησης της γνώμης των διαφόρων ομάδων χρηστών, χωρίς να χρειάζεται η διεξαγωγή επαναληπτικών μετρήσεων, οι οποίες έχουν υψηλό κόστος.

8.1.2 Παρουσίαση των φάσεων της μεθοδολογίας

Η μεθοδολογία που προτείνεται βασίζεται στις μεθόδους που περιγράφηκαν στα προηγούμενα κεφάλαια της διατριβής και κάνει χρήση των αποτελεσμάτων και των συμπερασμάτων που προέκυψαν σε αυτά. Με άλλα λόγια, γίνεται μία σύνθεση των μεθόδων αυτών σε ακολουθιακά στάδια από την οποία προκύπτει η προτεινόμενη μεθοδολογία. Εδώ θα πρέπει να σημειωθεί ότι για την εφαρμογή της μεθοδολογίας σε μία εταιρεία παραγωγής λογισμικού, πέρα από τους χρήστες που συμμετέχουν στις εξωτερικές μετρήσεις και το ίδιο το λογισμικό που τους παραδίδεται, απαραίτητο στοιχείο είναι και ο ίδιος ο κώδικας, στον οποίον θα γίνουν οι εσωτερικές μετρήσεις. Προφανώς, σε μία εταιρεία παραγωγής λογισμικού τα παραδοτέα δεν είναι μόνο ο κώδικας και το λογισμικό που τρέχει, αλλά όλα όσα πρέπει να συνοδεύουν την παραγωγή λογισμικού, όπως για παράδειγμα τα εγχειρίδια εγκατάστασης και χρήσης, οι αναφορές που προκύπτουν στις διάφορες φάσεις του κύκλου ζωής λογισμικού, οι περιπτώσεις ελέγχου και τα αποτελέσματά που προκύπτουν κτλ. Η προτεινόμενη

μεθοδολογία όμως εστιάζει μόνο στον κώδικα και το τελικό λογισμικό, θεωρώντας ότι στα πλαίσια του προγράμματος ποιότητας της εταιρείας θα έχουν προβλεφθεί μέθοδοι και έλεγχοι για τη διασφάλιση της ποιότητας και των υπολοίπων. Εξάλλου, ο σκοπός της μεθοδολογίας δεν είναι να αντικαταστήσει το πρόγραμμα ποιότητας της εταιρείας, αλλά να το συμπληρώσει με τις μεθόδους που περιγράφει.

Στο Σχήμα 8.1 παρουσιάζονται γραφικά οι διάφορες φάσεις της μεθοδολογίας. Στην απεικόνιση αυτή έγινε χρήση της σημειολογίας από τα Διαγράμματα Ροής Δεδομένων, όπου στις φυσαλίδες τους σχήματος αναφέρονται οι διάφορες διαδικασίες της μεθοδολογίας, ενώ στις ακμές που τις συνδέουν αναφέρονται οι πληροφορίες και τα δεδομένα που μετακινούνται από τη μία στην άλλη.

Η μεθοδολογία θα μπορούσε να θεωρηθεί ως ένας συνεχής κύκλος παραγωγής κώδικα, μετρήσεων (εσωτερικών και εξωτερικών) και βελτιστοποιήσεων, μέχρις ότου τα αποτελέσματα για την ποιότητα του παραγόμενου λογισμικού αποβούν ικανοποιητικά για τις αρχικές προδιαγραφές που έχουν θεσπιστεί. Επομένως, οι αρχικές ενέργειες μίας εταιρείας σύμφωνα με τη μεθοδολογία θα πρέπει να είναι ο *Καθορισμός επιθυμητών ορίων στις εσωτερικές μετρικές*, ο *Καθορισμός βαρύτητας των ποιοτικών χαρακτηριστικών* και ο *Καθορισμός των προαπαιτούμενων για την εφαρμογή της θεωρίας Αναθεώρησης Άποψης*. Στην πρώτη ενέργεια καθορίζονται τα κατώτερα αποδεκτά όρια των μετρικών που θα χρησιμοποιηθούν κατά τη διεξαγωγή των εσωτερικών μετρήσεων και αποθηκεύονται στη *Βάση Εσωτερικών Μετρικών*. Στη δεύτερη ενέργεια καθορίζονται τα βάρη όλων των ποιοτικών χαρακτηριστικών του προτύπου ή μοντέλου που χρησιμοποιεί η εταιρεία και αποθηκεύονται στη *Βάση Εξωτερικών Μετρήσεων*. Θα πρέπει να σημειωθεί ότι τα βάρη αυτά είναι διαφορετικά για κάθε έργο, αφού, όπως είναι προφανές, σε κάθε προϊόν λογισμικού δίνεται και διαφορετική έμφαση στους παράγοντες που απαρτίζουν την ποιότητα. Στην τρίτη ενέργεια καθορίζονται τα διάφορα σύνολα περιορισμών, όπως αυτά αναφέρθηκαν στα κεφάλαια 6 και 7, για όλες τις ομάδες χρηστών, δηλαδή για κάθε επίπεδο εμπειρίας των χρηστών που θα συμμετέχουν στις εξωτερικές μετρήσεις. Τα σύνολα αυτά αποθηκεύονται επίσης στη *Βάση Εξωτερικών Μετρήσεων*.

Το επόμενο βήμα μετά το πέρας των παραπάνω αρχικοποιήσεων είναι η διεξαγωγή μετρήσεων. Αρχικά γίνονται οι *Εσωτερικές Μετρήσεις*, οι οποίες παίρνουν ως είσοδο τον κώδικα από τη διαδικασία *Παραγωγής Κώδικα* και αποθηκεύουν τα αποτελέσματα στη *Βάση Εσωτερικών Μετρήσεων*. Στη συνέχεια, κάνοντας χρήση τόσο των παραπάνω αποτελεσμάτων όσο και των αποδεκτών ορίων στις χρησιμοποιούμενες μετρικές, γίνεται ο *Έλεγχος των Ποιοτικών Χαρακτηριστικών* και ο *Έλεγχος της Συντηρησιμότητας* του κώδικα και συγγράφονται οι αντίστοιχες *Αναφορές*. Όπως αναφέρθηκε και στο κεφάλαιο 5 και απεικονίζεται κατάλληλα στο Σχήμα 8.1, κατά τον έλεγχο της συντηρησιμότητας γίνεται χρήση και του ίδιου του κώδικα, ειδικά στις περιπτώσεις όπου τα χαμηλά αποτελέσματα των εσωτερικών μετρικών είναι αναπόφευκτα. Οι παραπάνω έλεγχοι θα οδηγήσουν στη διαδικασία *Απόφασης για βελτιστοποιήσεις*, κατά την οποία είτε θα αναφερθούν τα προβλήματα και οι διορθωτικές κινήσεις που σημαίνει επιστροφή στη διαδικασία *Παραγωγής Κώδικα* για βελτιώσεις είτε, αν η εσωτερική ποιότητα του λογισμικού κριθεί ικανοποιητική, θα ακολουθήσουν οι επόμενες ενέργειες της μεθοδολογίας, δηλαδή ο *Έλεγχος και η Ολοκλήρωση για την παραγωγή λογισμικού*. Από την τελευταία αυτή διαδικασία προκύπτει το λογισμικό ως τελικό προϊόν, το οποίο πλέον μπορεί να χρησιμοποιηθεί στη διεξαγωγή *Εξωτερικών Μετρήσεων*.

Σχήμα 8.1: Παρουσίαση των φάσεων της μεθοδολογίας

Για τον καθορισμό της εξωτερικής ποιότητας του λογισμικού γίνεται χρήση κατάλληλων ερωτηματολογίων, τα οποία προκύπτουν από τη διαδικασία *Προετοιμασίας Εξωτερικών Μετρήσεων*. Παράλληλα, σε αυτή τη διαδικασία καθορίζεται το δείγμα των χρηστών που θα συμμετέχει, καθώς και οι χρονικές στιγμές που θα διεξαχθούν οι μετρήσεις. Πέρα από τις *Εξωτερικές Μετρήσεις*, διεξάγονται παράλληλα και *Μετρήσεις Εμπειρίας των Χρηστών*, οι οποίες καθορίζουν και τη βαρύτητα της γνώμης τους. Τα αποτελέσματα της γνώμης των χρηστών και τα αποτελέσματα της εμπειρίας τους αποθηκεύονται στη *Βάση Εξωτερικών Μετρήσεων*. Έτσι, ανά πάσα στιγμή και κάνοντας χρήση των μεθόδων ανάλυσης αποτελεσμάτων που παρουσιάστηκαν στο κεφάλαιο 3, μπορεί να γίνει ο *Υπολογισμός της Άποψης των Χρηστών για την Ποιότητα του Λογισμικού*. Το αποτέλεσμα της παραπάνω διαδικασίας υπολογισμού είναι η παραγωγή των αντίστοιχων *Αναφορών*. Παράλληλα, οι απόψεις των χρηστών για την ποιότητα μεταφέρονται στη διαδικασία *Απόφασης για βελτιστοποιήσεις*, η οποία με τη σειρά της μπορεί να οδηγήσει σε προτάσεις για διορθωτικές κινήσεις του κώδικα, δηλαδή του ίδιου του προϊόντος λογισμικού.

Όμως, καθώς με την πάροδο του χρόνου είναι πιθανόν να μεταβληθεί η άποψη του κάθε χρήστη για την ποιότητα λογισμικού, στη μεθοδολογία έχουν ενσωματωθεί αρχές από τη θεωρία Αναθεώρησης Άποψης. Για το λόγο αυτό, όπως απεικονίζεται και στο Σχήμα 8.1, έχει προβλεφθεί η διαδικασία *Αναθεώρησης Άποψης του Χρήστη*, όπου με κάθε νέο γνωστικό χαρακτηριστικό που ανακαλύπτει και παίρνοντας ως είσοδο από τη *Βάση Εξωτερικών Μετρήσεων* την αρχική του γνώμη και το σύνολο των περιορισμών της ομάδας που ανήκει μπορεί να προκύψει η νέα του γνώμη. Ο υπολογισμός της νέας γνώμης γίνεται αυτόματα, όπως παρουσιάστηκε στο κεφάλαιο 7, δηλαδή χωρίς την ανάγκη διεξαγωγής επαναληπτικών μετρήσεων. Τέλος, γίνεται ενημέρωση στη Βάση, όπου αποθηκεύεται η τελική γνώμη του χρήστη για την ποιότητα του προϊόντος λογισμικού. Παράλληλα, έχει προβλεφθεί μία διαδικασία *Αναβάθμισης της Εμπειρίας του Χρήστη*, η οποία είναι λογικό να συμβαίνει, καθώς με την πάροδο του χρόνου ο χρήστης γίνεται ολοένα και πιο έμπειρος για το συγκεκριμένο λογισμικό. Η στιγμή που θα τελεστεί η εν λόγω αναβάθμιση εξαρτάται από το πλήθος και κυρίως το είδος των νέων γνωστικών στοιχείων που θα ανακαλύψει ο χρήστης για το λογισμικό. Η συγκεκριμένη διαδικασία λοιπόν θα οδηγήσει στην αποθήκευση της ενημερωμένης εμπειρίας του χρήστη στη *Βάση Εξωτερικών Μετρήσεων*.

Η μεθοδολογία που παρουσιάζεται στην παρούσα διατριβή προτείνει τη συνεχή βελτιστοποίησή της μέσα από την εφαρμογή της στα πλαίσια του προγράμματος ποιότητας μίας εταιρείας. Για το λόγο αυτό, έχει προβλεφθεί η διαδικασία *Διορθωτικές Κινήσεις και Επαναρύθμιση της Μεθοδολογίας*, η οποία θα δεχτεί προτάσεις από τη διαδικασία της *Απόφασης για βελτιστοποιήσεις* και με τη σειρά της θα επαναπροσδιορίσει τις τρεις αρχικές διαδικασίες της μεθοδολογίας με τους καθορισμούς. Με άλλα λόγια, η ίδια η μεθοδολογία μπορεί μετά από χρήση όλων των παραπάνω διαδικασιών να δεχτεί βελτιστοποιήσεις και επανακαθορισμούς α) στα επιθυμητά όρια των εσωτερικών μετρικών, β) στις βαρύτητες των διάφορων ποιοτικών χαρακτηριστικών που απαρτίζουν την ποιότητα λογισμικού και γ) στα σύνολα των περιορισμών για την αναθεώρηση άποψης.

8.2 Ανάλυση των βημάτων και χρήση της μεθοδολογίας

Όπως έχει ήδη αναφερθεί, η παρούσα μεθοδολογία απαρτίζεται από διάφορες μεθόδους οι οποίες παρουσιάστηκαν εκτενώς στα προηγούμενα κεφάλαια. Στη

συνέχεια, γίνεται μία αναλυτική περιγραφή των διαφόρων βημάτων της μεθοδολογίας και της ένταξης των χρησιμοποιούμενων μεθόδων σε αυτήν. Παράλληλα, παρουσιάζονται ο τρόπος χρήσης της μεθοδολογίας από μία εταιρεία παραγωγής λογισμικού, καθώς και οι επιδιωκόμενοι στόχοι.

8.2.1 Εσωτερικές μετρήσεις

Όσον αφορά τις εσωτερικές μετρήσεις, η κάθε εταιρεία παραγωγής λογισμικού θα πρέπει αρχικά να επιλέξει ένα σύνολο από εσωτερικές μετρικές. Οι μετρικές αυτές θα πρέπει να αναφέρονται ρητά στο εγχειρίδιο ποιότητας του προγράμματος εξασφάλισης ποιότητας της εταιρείας, όπως παρουσιάστηκε στην ενότητα 1.6. Φυσικά, η επιλογή των κατάλληλων μετρικών για τη διεξαγωγή εσωτερικών μετρήσεων εξαρτάται κυρίως από τη φύση των προγραμμάτων που υλοποιούνται, καθώς και τις γλώσσες προγραμματισμού που χρησιμοποιούν οι κατασκευαστές λογισμικού της εταιρείας. Στη συνέχεια, για το κάθε έργο λογισμικού θα πρέπει να επιλέγονται οι κατάλληλες μετρικές από το παραπάνω σύνολο και να αναφέρονται στο πλάνο ποιότητας του έργου.

Παράλληλα με την επιλογή των παραπάνω μερικών θα πρέπει να γίνει και ο καθορισμός των επιθυμητών ορίων τους. Τα όρια αυτά είναι οι κατώτερες αποδεκτές τιμές στα αποτελέσματα των μετρήσεων. Επομένως, αν ένα τμήμα του παραγόμενου κώδικα εμφανίζει χαμηλότερες τιμές στις χρησιμοποιούμενες μετρικές από τα προβλεπόμενα όρια, τότε θα πρέπει να βελτιωθεί ή ακόμα και να ξαναγραφτεί. Οι αριθμητικές τιμές των ορίων εξαρτώνται από το επίπεδο ποιότητας που επιθυμεί η εταιρεία και από τη βαρύτητα που δίνει σε ορισμένα χαρακτηριστικά του κώδικα. Έτσι για παράδειγμα, αν χρησιμοποιείται η μετρική του McCabe και στο πλάνο ποιότητας ενός έργου δίνεται μεγάλη βαρύτητα στη χαμηλή πολυπλοκότητα του κώδικα, θα πρέπει να τεθεί και αντίστοιχο μικρό όριο στην ανώτερη αποδεκτή τιμή της κυκλωματικής πολυπλοκότητας.

Οι εσωτερικές μετρήσεις δεν είναι απαραίτητο να διεξαχθούν αφού έχει τελειώσει η ολοκλήρωση του λογισμικού. Αντιθέτως, σε οποιαδήποτε χρονική στιγμή της φάσης υλοποίησης μπορούν να γίνουν μετρήσεις του παραγόμενου κώδικα. Συνίσταται μάλιστα με την ολοκλήρωση κάθε ξεχωριστού τμήματος του λογισμικού και μάλιστα σε κάθε έκδοσή του το τμήμα αυτό να μετράται με βάση τις επιλεγμένες μετρικές λογισμικού. Με αυτό τον τρόπο, η ομάδα εργασίας θα μπορεί να αποφανθεί από πολύ νωρίς για την εσωτερική ποιότητα του κώδικα. Επομένως, συλλέγοντας τα αποτελέσματα από αυτές τις μετρήσεις, θα πρέπει να γίνει ο απαραίτητος έλεγχος ικανοποίησης ή μη των ορίων που έχουν τεθεί.

Όπως είδαμε στο κεφάλαιο 4, τα χαμηλά αποτελέσματα στις εσωτερικές μετρικές οδηγούν σε χαμηλής εξωτερικής ποιότητας λογισμικό. Συνεπώς, τα τμήματα κώδικα που εμφανίζουν χαμηλότερες τιμές στις μετρικές από τις επιθυμητές θα πρέπει να βελτιωθούν ή ακόμα και να ξαναγραφτούν. Πολλές φορές μάλιστα, είναι προτιμότερο να δοθεί μεγαλύτερη προσπάθεια κατά τη φάση της υλοποίησης με επαναλαμβανόμενους κύκλους ανάπτυξης κώδικα μέχρι να έχουμε ικανοποιητικά αποτελέσματα, παρά να εμφανιστούν προβλήματα χαμηλής ποιότητας κατά τη φάση της συντήρησης όπου το κόστος μίας αλλαγής είναι σαφέστερα πολύ μεγαλύτερο. Στις περιπτώσιολογικές μελέτες που παρουσιάστηκαν φάνηκε ότι η συσχέτιση των εσωτερικών μετρικών και της εξωτερικής ποιότητας είναι πολύ μεγάλη. Συνήθως, όσο υψηλότερα είναι τα αποτελέσματα των μετρικών τόσο υψηλότερη και η ποιότητα του λογισμικού όπως την αντιλαμβάνονται οι χρήστες. Βέβαια, όπως είδαμε, επειδή η

παραπάνω συνθήκη είναι αναγκαία αλλά όχι ικανή, ακόμα και αν τα αποτελέσματα των εσωτερικών μετρήσεων είναι ικανοποιητικά, θα πρέπει η εταιρεία παραγωγής λογισμικού να προχωρήσει στη διεξαγωγή εξωτερικών μετρήσεων για να αποφανθεί με σιγουριά για την ποιότητα των προϊόντων της.

Κατά τη μέτρηση της εσωτερικής ποιότητας του παραγόμενου κώδικα θα πρέπει να γίνεται διαφορετικός έλεγχος όσον αφορά τη συντηρησιμότητα και τα ποιοτικά υπο-χαρακτηριστικά που την αποτελούν. Στο κεφάλαιο 5 παρουσιάστηκε η μεγάλη συσχέτιση ανάμεσα στις μετρικές και στη συντηρησιμότητα. Μάλιστα, όσο μεγαλύτερα είναι τα αποτελέσματα των εσωτερικών μετρικών τόσο μεγαλύτερη είναι και η συντηρησιμότητα του κώδικα. Ειδικά για αυτό το ποιοτικό χαρακτηριστικό η ικανοποίηση των μερικών είναι συνθήκη ικανή. Όμως ορισμένες φορές εμφανίζονται και περιπτώσεις τμημάτων κώδικα με εκ των πραγμάτων χαμηλά αποτελέσματα στις εσωτερικές μετρικές, κυρίως λόγω της ιδιαίτερης φύσης των λειτουργιών του παραγόμενου προϊόντος που υλοποιούν ή ακόμα και της ίδιας της γλώσσας προγραμματισμού όπου έχουν υλοποιηθεί. Όπως είδαμε στις περιπτώσιολογικές μελέτες που παρουσιάστηκαν, τα τμήματα αυτά ενδέχεται να παρουσιάζουν ικανοποιητική συντηρησιμότητα σύμφωνα με τη γνώμη των προγραμματιστών. Επομένως, αν κατά τις εσωτερικές μετρήσεις ένα τμήμα κώδικα εμφανίσει χαμηλά αποτελέσματα στις μετρικές, δεν θα πρέπει να απορρίπτεται αυτόματα όσον αφορά τη συντηρησιμότητά του. Αντιθέτως, θα πρέπει να διερευνώνται οι λειτουργίες που υλοποιεί και να εξετάζεται η δυνατότητα ή όχι για την περαιτέρω βελτίωσή του.

8.2.2 Εξωτερικές μετρήσεις

Όταν μία έκδοση του παραγόμενου λογισμικού είναι διαθέσιμη, μετά τον έλεγχο και την ολοκλήρωσή του το τμήμα εξασφάλισης ποιότητας της εταιρείας μπορεί να ενεργοποιήσει τη διαδικασία των εξωτερικών μετρήσεων. Με άλλα λόγια, το τμήμα εξασφάλισης ποιότητας βασισμένο στις αρχικές προδιαγραφές του έργου εξάγει ένα πρόγραμμα έρευνας για τη μέτρηση της άποψης των πελατών για τα εξωτερικά ποιοτικά χαρακτηριστικά του λογισμικού. Ασφαλώς, το πρόγραμμα έρευνας αυτό μπορεί να βασιστεί σε μία ή περισσότερες από τις τεχνικές μέτρησης που περιγράφηκαν στην ενότητα 3.3. Στην παρούσα μεθοδολογία δίνεται μεγαλύτερη έμφαση στην έρευνα με τη χρήση των ερωτηματολογίων, χωρίς αυτό βέβαια να σημαίνει ότι δεν μπορούν να χρησιμοποιηθούν παράλληλα και άλλες τεχνικές μέτρησης. Η επιλογή μίας τεχνικής ως καταλληλότερης για τη διεξαγωγή εξωτερικών μετρήσεων βασίζεται στις ιδιαιτερότητες του κάθε έργου. Παρόλα αυτά, η επιλογή των ερωτηματολογίων μπορεί να εφαρμοστεί στην πλειοψηφία των περιπτώσεων, αν όχι σε όλες τις περιπτώσεις.

Ένα βασικό στοιχείο στις εξωτερικές μετρήσεις είναι ο καθορισμός του δείγματος χρηστών που θα συμμετέχει. Το τμήμα εξασφάλισης ποιότητας θα πρέπει, ανάλογα με το έργο, να αποφασίσει για τον αριθμό των χρηστών, το βαθμό εμπειρίας τους, καθώς και τη βαρύτητα της γνώμης τους ανάλογα με την εμπειρία τους. Επιπλέον σημαντικά στοιχεία αποτελούν ο τόπος και η χρονική στιγμή όπου θα διεξαχθεί η έρευνα. Είναι ασφαλώς προτιμότερο το δείγμα μας να αποτελείται από έμπειρους χρήστες, αλλά υπάρχουν και ορισμένες περιπτώσεις, όπως για παράδειγμα σε προϊόντα εκπαιδευτικού λογισμικού, όπου είναι προτιμότερος ένας μεγάλος αριθμός από άπειρους. Σε γενικές περιπτώσεις και σύμφωνα με τα αποτελέσματα της ενότητας 6.2, το δείγμα μας μπορεί να αποτελείται από λίγους έμπειρους χρήστες, καθώς αυτοί σχηματίζουν από πολύ νωρίς μία άποψη για την ποιότητα του λογισμικού που είναι πολύ κοντά στην τελική τους άποψη. Διαφορετικά, αν δεν είναι διαθέσιμος ένας

ικανοποιητικός αριθμός από έμπειρους χρήστες, το δείγμα θα πρέπει να απαρτίζεται και από έναν σαφέστερα μεγαλύτερο αριθμό άπειρων, αφού η διακύμανση στις αρχικές τους απόψεις είναι πολύ μεγάλη, με την ελπίδα ότι ο μέσος όρος των αρχικών αυτών απόψεων θα απέχει πολύ λίγο από την τελική τους άποψη.

Ασφαλώς, με την πάροδο του χρόνου και με τη συνεχή χρήση του λογισμικού θα αυξηθεί η εμπειρία των χρηστών. Υπάρχουν όμως και περιπτώσεις λογισμικού που θα χρησιμοποιηθεί για πολύ λίγο καιρό. Ένα τέτοιο παράδειγμα ήταν η εφαρμογή σύγχρονης τηλεσυνεργασίας στο Πρόγραμμα Σπουδών Πληροφορικής του Ελληνικού Ανοικτού Πανεπιστημίου, όπου το περιβάλλον σύγχρονης συνεργασίας χρησιμοποιήθηκε από τους φοιτητές για λιγότερο από 2 μήνες [XEN04a], [XEN04b], [XEN04c]. Σε αυτές τις περιπτώσεις θα πρέπει να δοθεί ιδιαίτερη προσοχή στα εγχειρίδια χρήσης και την εκπαίδευση των χρηστών, αφού δεν θα υπάρχει αρκετό χρονικό διάστημα για να γίνουν έμπειροι.

Σύμφωνα με την προτεινόμενη μεθοδολογία, θα πρέπει να προετοιμαστούν δύο ειδών ερωτηματολόγια για τη διεξαγωγή των εξωτερικών μετρήσεων. Το πρώτο αφορά τη μέτρηση της γνώμης των χρηστών για τα διάφορα ποιοτικά χαρακτηριστικά του λογισμικού και κυρίως αποτελείται από ερωτήσεις πολλαπλής επιλογής. Προτείνεται μάλιστα οι τιμές στις πιθανές απαντήσεις να παίρνουν τιμές από το διάστημα $[0,10]$ και να ισαπέχουν μεταξύ τους. Παράλληλα, το ερωτηματολόγιο θα πρέπει να είναι δομημένο με τέτοιο τρόπο ώστε οι ερωτήσεις να μπορούν να ομαδοποιηθούν σύμφωνα με το ποιοτικό χαρακτηριστικό στο οποίο αναφέρονται. Με αυτόν τον τρόπο θα μπορούμε να καθορίσουμε τη γνώμη του κάθε χρήστη τόσο για τη συνολική ποιότητα ενός προϊόντος λογισμικού όσο και για κάθε ποιοτικό χαρακτηριστικό ξεχωριστά. Το δεύτερο ερωτηματολόγιο αφορά τη μέτρηση της εμπειρίας των χρηστών που θα συμμετάσχουν στην έρευνα. Σύμφωνα με αυτό, οι χρήστες θα ομαδοποιηθούν σε διάφορες κατηγορίες εμπειρίας, όπου στην καθεμία θα αντιστοιχεί και διαφορετικό βάρος γνώμης. Στο κεφάλαιο 3 γίνεται αναλυτικότερη περιγραφή των δύο παραπάνω ερωτηματολογίων.

Για τη στατιστική επεξεργασία των αποτελεσμάτων των εξωτερικών μετρήσεων γίνεται χρήση ορισμένων μαθηματικών σχέσεων. Συγκεκριμένα, η φόρμουλα $C_j O_i$ που δίνεται από τη σχέση 8.2.1 υπολογίζει τη γνώμη του χρήστη i για την ποιότητα του προϊόντος σύμφωνα με το ποιοτικό χαρακτηριστικό j . Σε αυτή τη σχέση το m είναι ο αριθμός των ερωτήσεων του ερωτηματολογίου για αυτό το ποιοτικό χαρακτηριστικό, ο παράγοντας Q_k είναι το βάρος που δίνεται στην ερώτηση k και ο παράγοντας V_k είναι η τιμή που αντιστοιχεί στην απάντηση που επέλεξε ο χρήστης σε αυτή την ερώτηση.

$$C_j O_i = \frac{\sum_{k=1}^m (Q_k \cdot V_k)}{\sum_{k=1}^m Q_k} \quad (\text{Σχέση 8.2.1})$$

Η φόρμουλα O_i που δίνεται από τη σχέση 8.2.2 υπολογίζει τη γνώμη του χρήστη i για την ποιότητα του προϊόντος σύμφωνα με το σύνολο των ποιοτικών χαρακτηριστικών στα οποία αναφέρεται το ερωτηματολόγιο. Σε αυτή τη σχέση το n είναι ο αριθμός των διάφορων ποιοτικών χαρακτηριστικών, ο παράγοντας C_j είναι το βάρος που δίνεται στο ποιοτικό χαρακτηριστικό j (από τον σχεδιαστή του ερωτηματολογίου) και ο παράγοντας $C_j O_i$ είναι η γνώμη του χρήστη για αυτό το ποιοτικό χαρακτηριστικό, όπως αυτή υπολογίστηκε από τη σχέση 8.2.1.

$$O_i = \frac{\sum_{j=1}^n (C_j \cdot C_j O_i)}{\sum_{j=1}^n C_j} \quad (\text{Σχέση 8.2.2})$$

Τέλος, για τον υπολογισμό της μέσης γνώμης των χρηστών για την ποιότητα του προϊόντος λογισμικού με βαρύτητα εμπειρίας μπορεί να χρησιμοποιηθεί είτε η μέθοδος QWCO (Qualifications Weighted Customer Opinion), η οποία δίνεται από τη σχέση 8.2.3, είτε η μέθοδος QWCO_{DS} (Qualifications Weighted Customer Opinion with Double Safeguards), η οποία δίνεται από τη σχέση 8.2.4.

$$QWCO = \frac{\sum_{i=1}^x (O_i \cdot E_i)}{\sum_{i=1}^x E_i} \quad (\text{Σχέση 8.2.3})$$

$$QWCO_{DS} = \frac{\sum_{i=1}^x \left(O_i \cdot E_i \cdot \frac{S_i}{S_T} \cdot P_i \right)}{\sum_{i=1}^x \left(E_i \cdot \frac{S_i}{S_T} \cdot P_i \right)} \quad (\text{Σχέση 8.2.4})$$

Ο σκοπός αυτών των δύο μεθόδων είναι να δώσουν στη γνώμη του κάθε χρήστη μία βαρύτητα ανάλογα με τα προσόντα του. Για το λόγο αυτό, ο παράγοντας O_i είναι το κανονικοποιημένο αποτέλεσμα της γνώμης του χρήστη i και μπορεί να υπολογιστεί από τη σχέση 8.2.2, ο παράγοντας E_i απεικονίζει τα προσόντα, δηλαδή το βαθμό εμπειρίας, του χρήστη i , ενώ το x είναι ο αριθμός των χρηστών που συμμετέχουν στην έρευνα. Για να εντοπιστούν πιθανά λάθη κατά τη συμπλήρωση των ερωτηματολογίων καθώς και η ορθότητά του, μπορεί να χρησιμοποιηθεί ένας αριθμός από ερωτήσεις ασφαλείας μέσα σε αυτό. Έτσι, στη σχέση 8.2.4 ο παράγοντας S_i είναι ο αριθμός των ερωτήσεων ασφαλείας στις οποίες ο χρήστης i απάντησε σωστά, ενώ ο παράγοντας S_T είναι ο συνολικός αριθμός ερωτήσεων ασφαλείας του ερωτηματολογίου. Ο επιπρόσθετος παράγοντας P_i παίρνει τιμές 0 ή 1. Συγκεκριμένα η τιμή του P_i είναι ίση με 0 στην περίπτωση κατά την οποία ο πελάτης i έχει απαντήσει έστω και μία ερώτηση ασφαλείας λάθος από αυτές τις ερωτήσεις ασφαλείας που έχουν ενσωματωθεί στο τμήμα μέτρησης των προσόντων του πελάτη. Αντίθετα, η τιμή του P_i είναι ίση με 1 μόνο στην περίπτωση όπου ο πελάτης i έχει απαντήσει σωστά σε όλες αυτές τις ερωτήσεις ασφαλείας. Όπως προκύπτει από την παραπάνω σχέση, οι πελάτες που έχουν δώσει λάθος απάντηση στο τμήμα του ερωτηματολογίου που σχετίζεται με τη μέτρηση της εμπειρίας τους αποκλείονται πρακτικά και δεν συμμετέχουν στην έρευνα, αφού οι απαντήσεις τους δεν λαμβάνονται υπόψη στην ανάλυση.

Θα πρέπει να σημειωθεί ότι η έρευνα μπορεί να εκτελεστεί παράλληλα με παρόμοιες λειτουργίες της εταιρείας, όπως είναι η συλλογή των αναφορών προβλημάτων (defect reports) ή οι έρευνες ικανοποίησης του χρήστη. Τα αποτελέσματα των εξωτερικών μετρήσεων διατίθενται στο τμήμα εξασφάλισης ποιότητας. Στην περίπτωση κατά την οποία οι χρήστες εντοπίσουν έντονα προβλήματα για ορισμένα ποιοτικά χαρακτηριστικά ενός έργου, ενδέχεται να αποφασιστούν αλλαγές σε αυτό και ετοιμασία νέας έκδοσης στην οποία θα δοθεί έμφαση στα ποιοτικά χαρακτηριστικά όπου εντοπίστηκε το πρόβλημα. Αυτό

συνεπάγεται αλλαγές στο πρόγραμμα εξασφάλισης ποιότητας του έργου, είτε με τη μορφή αυστηρότερων ορίων είτε με την εισαγωγή νέων μετρικών με στόχο την επικέντρωση του ενδιαφέροντος στα ποιοτικά χαρακτηριστικά τα οποία οι χρήστες εντόπισαν ως μη ικανοποιητικά.

Οι εξωτερικές μετρήσεις εκτός από τη δυνατότητα άμεσης μέτρησης των ποιοτικών χαρακτηριστικών του λογισμικού όπως τα αντιλαμβάνονται οι χρήστες, προσφέρουν και επιμέρους δυνατότητες, όπως η δυνατότητα ελέγχου της ακρίβειας των εσωτερικών μετρήσεων που βασίζονται σε νέες μετρικές που εντάχθηκαν στο εγχειρίδιο ποιότητας και η δυναμική παρακολούθηση των ποιοτικών προδιαγραφών για επιμέρους χαρακτηριστικά όπως τις αντιλαμβάνονται οι χρήστες του λογισμικού.

8.2.3 Αναθεώρηση άποψης χρήστη

Με βάση την παραπάνω έρευνα κατά τις εξωτερικές μετρήσεις μπορεί να καθοριστεί η αρχική άποψη των χρηστών για την ποιότητα ενός προϊόντος λογισμικού. Όπως αναφέρθηκε όμως ήδη στο κεφάλαιο 6, με την πάροδο του χρόνου οι χρήστες μπορεί να παρουσιάσουν διάφορες διακυμάνσεις στην άποψή τους μέχρι να καταλήξουν στην τελική τους γνώμη για την ποιότητα, γεγονός που μπορεί να συμβεί μετά από ένα εύλογο χρονικό διάστημα. Μάλιστα, όσο χαμηλότερη είναι η εμπειρία ενός χρήστη τόσο μεγαλύτερες αυτές οι διακυμάνσεις και τόσο μεγαλύτερο το χρονικό διάστημα που απαιτείται για να κατασταλάξει ο χρήστης στην τελική του άποψη για την ποιότητα ενός προϊόντος λογισμικού.

Επειδή όμως η διεξαγωγή εξωτερικών μετρήσεων απαιτεί μεγάλο κόστος, τόσο σε χρόνο όσο και σε ανθρώπινο δυναμικό, η μεθοδολογία της παρούσας διατριβής προτείνει την ενσωμάτωση κανόνων από τη θεωρία Αναθεώρησης Άποψης. Με τον τρόπο αυτό μπορούν να εκτιμηθούν οι μεταβολές της γνώμης ενός χρήστη για την ποιότητα χωρίς την ανάγκη διεξαγωγής επαναληπτικών μετρήσεων. Για να είναι όμως αυτό εφικτό, θα πρέπει το τμήμα εξασφάλισης ποιότητας να έχει εκ των προτέρων καθορίσει ένα σύνολο περιορισμών για κάθε ομάδα χρηστών. Όπως ήδη αναφέρθηκε, οι χρήστες ενός προϊόντος λογισμικού έχουν την αντίληψη πως υπάρχει μία συσχέτιση ανάμεσα στα ποιοτικά χαρακτηριστικά του λογισμικού. Η αναπαράσταση της συσχέτισης αυτής που ακολουθήθηκε γίνεται με τον καθορισμό ενός συνόλου από περιορισμούς ανάμεσα στις βαθμολογίες του κάθε χαρακτηριστικού ποιότητας που αφορούν τον τελικό χρήστη. Κάθε περιορισμός καθορίζει την πιθανή σχέση ανάμεσα σε δύο χαρακτηριστικά και ακολουθεί το φορμαλισμό: $A_i \in [h_1, h_2] \Rightarrow A_j \in [c_1, c_2]$, όπου τα A_i και A_j είναι οι βαθμολογίες για τα δύο διαφορετικά ποιοτικά χαρακτηριστικά i και j , ενώ τα $[h_1, h_2]$ και $[c_1, c_2]$ είναι υποσύνολα του συνόλου $[0, 10]$, το οποίο σημαίνει ότι αν η βαθμολογία για το χαρακτηριστικό i παίρνει τιμές από h_1 έως h_2 , τότε η βαθμολογία για το χαρακτηριστικό j θα παίρνει τιμές από c_1 έως c_2 .

Για να γίνει ένας διαχωρισμός στις απόψεις των χρηστών για τις σχέσεις ανάμεσα στα ποιοτικά χαρακτηριστικά, θα πρέπει να καθοριστούν διαφορετικά σύνολα από περιορισμούς ανάλογα με το επίπεδο εμπειρίας των χρηστών. Εφόσον, όπως παρουσιάστηκε παραπάνω, οι άπειροι χρήστες θεωρούν σε μεγαλύτερο βαθμό από τους έμπειρους ότι υπάρχει συσχέτιση ανάμεσα σε αυτά τα χαρακτηριστικά, τότε θα πρέπει όσο πιο έμπειρος είναι ο χρήστης τόσο λιγότερο αυστηροί να είναι οι περιορισμοί. Επομένως, οι χρήστες θα πρέπει αρχικά να κατηγοριοποιούνται σε διάφορα επίπεδα εμπειρίας, δημιουργώντας αντίστοιχες ομάδες χρηστών και στην καθεμία να αντιστοιχεί διαφορετικό σύνολο από περιορισμούς.

Για την εφαρμογή των κανόνων της θεωρίας Αναθεώρησης Άποψης είναι απαραίτητος ο καθορισμός της αρχικής άποψης των τελικών χρηστών για την ποιότητα ενός προϊόντος λογισμικού. Αυτό μπορεί προφανώς να επιτευχθεί με τη διεξαγωγή εξωτερικών μετρήσεων στα πολύ αρχικά στάδια χρήσης του προϊόντος. Στη συνέχεια, θα πρέπει η εταιρεία παραγωγής λογισμικού να βρίσκεται σε συνεχή επαφή με τους χρήστες ώστε να ενημερώνεται και να καταγράφει τα νέα γνωστικά στοιχεία που ανακαλύπτουν αυτοί για το προϊόν, τα οποία δυνητικά μπορούν να μεταβάλουν την άποψή τους για την ποιότητά του. Αυτή η επαφή μπορεί να γίνει με διάφορους από τους κλασικούς τρόπους επικοινωνίας, όπως για παράδειγμα οι σύντομες συζητήσεις σε τακτά χρονικά διαστήματα με τον κάθε χρήστη στον χώρο εργασίας του, οι τηλεφωνικές επικοινωνίες, η ανταλλαγή κειμένων ή μηνυμάτων μέσω ηλεκτρονικού ταχυδρομείου ή άλλων online υπηρεσιών, η συζήτηση μέσω υπηρεσιών δημοσίων συζητήσεων (forum) κ.ά.

Πρέπει να σημειωθεί στον κάθε χρήστη που θα συμμετέχει στην έρευνα να αναφέρει όλα τα νέα γνωστικά στοιχεία που ανακαλύπτει για το προϊόν λογισμικού, είτε αυτά είναι θετικά είτε αρνητικά. Θα πρέπει μάλιστα να προσεχθεί ιδιαίτερος αυτή η επικοινωνία να μην εκφυλιστεί σε μία απλή αναφορά λαθών του λογισμικού που ανακαλύπτει ο χρήστης. Αντιθέτως, θα πρέπει να αναφέρει επίσης και κάθε νέα λειτουργία που ανακαλύπτει, τις διάφορες προσθήκες ή βελτιώσεις που θα ήθελε σε μία πιθανή επόμενη έκδοση του λογισμικού, ακόμα και την πιθανή μεταβολή της γνώμης του για κάποιο ποιοτικό κριτήριο σχετικά με το λογισμικό. Καθώς η εμπειρία ενός χρήστη αυξάνει με την πάροδο του χρόνου, θα πρέπει μετά από ένα εύλογο χρονικό διάστημα ο χρήστης να αντιστοιχιστεί σε σύνολο περιορισμών μεγαλύτερης εμπειρίας.

Γνωρίζοντας την αρχική άποψη ενός χρήστη για την ποιότητα ενός προϊόντος λογισμικού και έχοντας καθορίσει ποιο σύνολο περιορισμών αντιστοιχεί για αυτόν, το τμήμα εξασφάλισης ποιότητας της εταιρείας μπορεί με κάθε νέο γνωστικό στοιχείο που αυτός ανακαλύπτει να υπολογίζει και τη νέα αναθεωρημένη άποψή του για αυτό το προϊόν. Θα πρέπει πρώτα αυτό το νέο γνωστικό στοιχείο να «μεταφραστεί» σε πιθανές νέες τιμές στα ποιοτικά χαρακτηριστικά και υπο-χαρακτηριστικά που αντιστοιχεί. Οι νέες αυτές τιμές μπορεί με τη σειρά τους και με βάση το σύνολο των περιορισμών που ισχύει σε αυτή την περίπτωση να οδηγήσουν σε αναθεωρήσεις και στις τιμές άλλων ποιοτικών χαρακτηριστικών. Έτσι, σύμφωνα με όσα αναφέρθηκαν στο κεφάλαιο 7, είναι δυνατόν να υπολογιστεί εξολοκλήρου η νέα αναθεωρημένη άποψη του χρήστη. Τέλος, είναι δυνατόν ένας χρήστης να αναφέρει ταυτόχρονα δύο ή περισσότερα νέα γνωστικά στοιχεία που ανακάλυψε για το προϊόν λογισμικού. Σε αυτήν την περίπτωση, αν οι αναθεωρήσεις ενός γνωστικού στοιχείου έρχονται σε αντίθεση με τις αναθεωρήσεις ενός άλλου, θα πρέπει να μελετάται πιθανή αναβάθμιση της εκτίμησης της εμπειρίας του χρήστη. Με άλλα λόγια, ο χρήστης θα πρέπει πλέον να τοποθετηθεί σε ομάδα υψηλότερης εμπειρίας, όπου το σύνολο των περιορισμών είναι πιο χαλαρό, έτσι ώστε να ικανοποιούνται όλες οι αναθεωρήσεις που προέκυψαν από τα νέα γνωστικά στοιχεία που ανακάλυψε.

8.2.4 Συνεχής εξέλιξη

Πρέπει να τονιστεί ότι μία βασική λειτουργία του τμήματος εξασφάλισης ποιότητας σύμφωνα με την προτεινόμενη μεθοδολογία είναι η συνεχής εξέλιξη. Τα αποτελέσματα των εσωτερικών μετρικών πρέπει να αντιπαραβάλλονται με τα αποτελέσματα των εξωτερικών και να γίνονται διορθωτικές αλλαγές στο πρόγραμμα των εσωτερικών μετρήσεων όπου εντοπίζονται αδυναμίες του. Αυτές οι αλλαγές

καταγράφονται στο πλάνο εξασφάλισης ποιότητας της εταιρείας. Νέες μετρικές για συγκεκριμένα ποιοτικά χαρακτηριστικά, οι οποίες παρουσιάζονται, ελέγχονται και επιβεβαιώνονται από τις εξωτερικές μετρήσεις, μπορούν να ενσωματωθούν στο πλάνο εξασφάλισης ποιότητας.

Παράλληλα, όπως φαίνεται και από το Σχήμα 8.1, θα πρέπει να προβλεφθεί μία διαδικασία για διορθωτικές κινήσεις και επαναρύθμιση της ίδιας της μεθοδολογίας. Ο ρόλος της διαδικασίας αυτής είναι ο επαναπροσδιορισμός των προαπαιτούμενων ορίων και τιμών για τις εσωτερικές και τις εξωτερικές ρυθμίσεις. Συγκεκριμένα, το τμήμα εξασφάλισης ποιότητας έχει τη δυνατότητα να επανακαθορίζει τα επιθυμητά όρια στις εσωτερικές μετρικές που χρησιμοποιούνται στο πλάνο εξασφάλισης ποιότητας της εταιρείας. Επίσης, μπορεί να επανακαθορίζει τη βαρύτητα που πρέπει να δίνεται στα διάφορα χαρακτηριστικά και υπο-χαρακτηριστικά στα οποία απαρτίζεται η ποιότητα λογισμικού. Με αυτό τον τρόπο δηλώνονται τα ποιοτικά κριτήρια στα οποία η εταιρεία παραγωγής λογισμικού θα πρέπει να δώσει μεγαλύτερη έμφαση. Παράλληλα, επαναπροσδιορίζονται τα βάρη των διαφόρων ερωτήσεων του ερωτηματολογίου μέτρησης της γνώμης των χρηστών για την ποιότητα ενός προϊόντος λογισμικού.

Μέσω αυτής της διαδικασίας επαναρύθμισης της μεθοδολογίας είναι δυνατόν να γίνουν βελτιστοποιήσεις και στα προαπαιτούμενα στοιχεία για την εφαρμογή της θεωρίας Αναθεώρησης Άποψης. Δηλαδή, μπορούν να επανακαθοριστούν τα σύνολα περιορισμών που θα πρέπει να ισχύουν για κάθε ομάδα χρηστών. Επιπλέον, το τμήμα εξασφάλισης ποιότητας έχει τη δυνατότητα να προτείνει για κάποιο έργο και τη διεξαγωγή επαναληπτικών εξωτερικών μετρήσεων, ώστε είτε να επιβεβαιωθούν τα σύνολα των περιορισμών και γενικότερα οι ισχύοντες κανόνες που ενσωματώθηκαν από τη θεωρία Αναθεώρησης Άποψης είτε να διαφανούν οι ανάγκες για τις απαιτούμενες διορθωτικές κινήσεις.

Τέλος, το τμήμα εξασφάλισης ποιότητας έχει ως στόχο του να μαθαίνει από κάθε έργο και την εμπειρία αυτή να τη μετουσιώνει σε βελτιώσεις στο πλάνο εξασφάλισης ποιότητας. Εξάλλου, η συνεχής εξέλιξη είναι βασικός παράγοντας επιτυχίας του τμήματος εξασφάλισης ποιότητας σε κάθε εταιρεία παραγωγής λογισμικού.

8.2.5 Πειραματική εφαρμογή

Η μεθοδολογία έγκαιρης εκτίμησης της γνώμης των χρηστών για την ποιότητα λογισμικού που παρουσιάζεται σε αυτή τη διατριβή αποτελείται από διάφορες μεθόδους, οι οποίες μπορούν να εφαρμοστούν και ξεχωριστά η μία από την άλλη. Συγκεκριμένα, οι μέθοδοι που αναφέρθηκαν και βασίζονται στις εσωτερικές μετρικές λογισμικού έχουν εφαρμοστεί σε μεγάλο βαθμό, ενώ τα αποτελέσματά τους έχουν επιβεβαιωθεί από διάφορες περιπτώσιολογικές μελέτες. Η διεξαγωγή μίας αντίστοιχης περιπτώσιολογικής μελέτης σε μεγάλη κλίμακα σχετικά με την εφαρμογή κανόνων από τη θεωρία Αναθεώρησης Άποψης, όπως αυτοί αναφέρθηκαν στα προηγούμενα κεφάλαια, αποτελεί ένα μελλοντικό στόχο για έρευνα.

Η μεθοδολογία της διατριβής έχει εφαρμοστεί πειραματικά σε μικρή κλίμακα. Τα συμπεράσματα που προέκυψαν βοήθησαν όχι μόνο στον υπολογισμό της ακρίβειας των αποτελεσμάτων που προκύπτουν, αλλά κυρίως στον επανακαθορισμό ορισμένων βημάτων της μεθοδολογίας. Είναι προφανές ότι για μία ουσιαστική διερεύνηση στη χρήση της μεθοδολογίας θα πρέπει να μελετηθεί ένας ικανοποιητικός αριθμός από έργα λογισμικού, τα οποία μάλιστα δεν θα πρέπει να έχουν υλοποιηθεί όλα από την ίδια εταιρεία. Για την παραπάνω μελέτη θα πρέπει να ληφθούν υπόψη διάφοροι

παράγοντες, όπως το πλήθος των χρηστών που χρησιμοποιεί το κάθε έργο, οι συνθήκες κάτω από τις οποίες γίνεται η χρήση, η διαφορετικότητα στην εμπειρία των χρηστών, ο πραγματικός χρόνος χρήσης που απαιτείται για το κάθε έργο ώστε να μπορούμε να αποφανθούμε με σαφήνεια για την εφαρμογή της μεθοδολογίας κ.ά.

Είναι προφανές ότι το κόστος μίας μελέτης σε αυτή την κλίμακα, όπου θα πρέπει να συμμετέχουν διάφορες εταιρείες παραγωγής λογισμικού και ικανοποιητικός αριθμός χρηστών για κάθε έργο, είναι πολύ μεγάλο κυρίως από άποψης χρόνου. Με άλλα λόγια είναι αδύνατον να συμπεριληφθεί και αυτή στα χρονικά πλαίσια της παρούσας διδακτορικής διατριβής, αφού θα απαιτούσε αρκετά χρόνια επιπλέον έρευνας. Αυτό φυσικά δεν σημαίνει ότι δεν αποτελεί έναν μελλοντικό μας ερευνητικό στόχο.

Για τη διεξαγωγή μίας τέτοιας μελέτης είναι απαραίτητο πρώτα να υλοποιηθεί μία αντίστοιχη πλατφόρμα, ώστε να υπολογίζονται αυτόματα με μαθηματικό τρόπο όλες οι αναθεωρήσεις στη βαθμολογία του κάθε χρήστη για την ποιότητα λογισμικού. Κατά την πειραματική εφαρμογή της μεθοδολογίας δεν υπήρχε διαθέσιμο ένα κατάλληλο εργαλείο, οπότε οι παραπάνω υπολογισμοί έγιναν αναγκαστικά με το χέρι, γεγονός που απαιτούσε αρκετές ανθρωποώρες. Παρόλα αυτά, έγιναν μερικές αξιοσημείωτες παρατηρήσεις. Καταρχήν, χρησιμοποιώντας κανόνες από τη θεωρία Αναθεώρησης Άποψης μπορούμε πράγματι να υπολογίσουμε ορθότερα την τελική γνώμη του χρήστη μετά τις διάφορες αναθεωρήσεις στην άποψή του. Αυτό διαπιστώθηκε όχι μόνο για τα κριτήρια της ποιότητας λογισμικού στα οποία άλλαξε η βαθμολογία του άμεσα, αλλά και για τα υπόλοιπα κριτήρια τα οποία σχετίζονται με τα πρώτα σύμφωνα με το σύνολο των περιορισμών που ισχύει ανάμεσά τους.

Επίσης, βρέθηκε ότι όσο μεγαλύτερος είναι ο αριθμός των ομάδων στις οποίες χωρίζονται οι χρήστες βάσει της εμπειρίας τους τόσο πιο ακριβή και τα αποτελέσματα της μεθοδολογίας. Φυσικά, ένας μεγάλος αριθμός από ομάδες αυξάνει κατά πολύ την πολυπλοκότητα της μεθοδολογίας και την κάνει δύσχρηστη. Έτσι, αυτός ο αριθμός προτείνεται να μην είναι μεγαλύτερος του 5. Τέλος, διαπιστώθηκε ότι προκύπτουν εγκυρότερα αποτελέσματα όταν χρησιμοποιούμε στη διατύπωση των περιορισμών όλα τα υπο-χαρακτηριστικά του ISO9126 αντί μόνο των έξι βασικών χαρακτηριστικών του. Αυτό είναι λογικό, γιατί αναπαριστώντας με ένα μεγαλύτερο διάνυσμα τη βαθμολογία του χρήστη για την ποιότητα, μπορούμε να συσχετίσουμε καλύτερα και αναλυτικότερα τους διάφορους παράγοντες που απαρτίζουν την ποιότητα λογισμικού.

Για να εφαρμοστεί σωστά η προτεινόμενη μεθοδολογία σε μία εταιρεία παραγωγής λογισμικού είναι απαραίτητο να συμμετέχουν ενεργά οι τελικοί χρήστες. Με άλλα λόγια, θα πρέπει αυτοί να πειστούν να αναφέρουν κάθε νέο στοιχείο που ανακαλύπτουν για το λογισμικό, είτε θετικό είτε αρνητικό, καθώς επίσης να αναφέρουν αν η γνώμη τους για κάποιο κριτήριο της ποιότητας έχει αλλάξει. Επίσης, το τμήμα ποιότητας της εταιρείας θα πρέπει να κατασκευάσει με τέτοιο τρόπο τα σύνολα των περιορισμών που απαιτούνται για την εφαρμογή της μεθοδολογίας, ώστε να διαφαίνεται σε ποια χαρακτηριστικά δίνεται μεγαλύτερη βαρύτητα. Δηλαδή όσο μεγαλύτερης βαρύτητας είναι το χαρακτηριστικό στο οποίο ενδέχεται να προκύψει μία αναθεώρηση τόσο περισσότερες αναθεωρήσεις στα υπόλοιπα χαρακτηριστικά της ποιότητας πρέπει να ακολουθήσουν.

8.3 Συμπεράσματα και επίλογος

Στην παρούσα διδακτορική διατριβή παρουσιάζεται μία μεθοδολογία για την έγκαιρη εκτίμηση της γνώμης των χρηστών για την ποιότητα λογισμικού, η οποία βασίζεται σε εσωτερικές και εξωτερικές μετρήσεις των ποιοτικών χαρακτηριστικών του λογισμικού, καθώς και σε αρχές από τη θεωρία Αναθεώρησης Άποψης. Η μεθοδολογία αυτή εντάσσεται σε κάθε πρόγραμμα εξασφάλισης ποιότητας μίας εταιρείας, το οποίο έχει οργανωθεί με βάση τα διεθνή πρότυπα. Είναι σαφώς γρηγορότερη και χαμηλότερου κόστους από την επαναληπτική διεξαγωγή μετρήσεων στους τελικούς χρήστες του λογισμικού.

Όπως είναι γνωστό, η έννοια της ποιότητας λογισμικού είναι άρρηκτα συνδεδεμένη με τα ποιοτικά χαρακτηριστικά, τις απαιτήσεις των πελατών-χρηστών και τις προδιαγραφές που θα πρέπει να ικανοποιεί το λογισμικό. Βασικοί στόχοι της μεθοδολογίας είναι α) η έγκαιρη εκτίμηση της γνώμης των χρηστών για την ποιότητα λογισμικού, ώστε να διορθωθούν νωρίς τα πιθανά λάθη και να γίνουν βελτιώσεις σε πρώιμες φάσεις του κύκλου ζωής λογισμικού, β) η ενεργή συμμετοχή του τελικού χρήστη σε όλες τις φάσεις του κύκλου ζωής λογισμικού, αφού κάθε διαδικασία κατά την παραγωγή λογισμικού θα πρέπει να ακολουθεί τους κανόνες εξασφάλισης ποιότητας που έχουν θεσπιστεί στο πλάνο ποιότητας ενός έργου, γ) η μείωση της προσπάθειας που θα απαιτηθεί στη φάση της συντήρησης, αφού πιθανά προβλήματα θα έχουν διαφανεί σε προηγούμενες φάσεις, δ) η μείωση του κόστους για την εύρεση της γνώμης του χρήστη για την ποιότητα ενός προϊόντος λογισμικού, αφού δεν είναι απαραίτητη η διεξαγωγή επαναληπτικών εξωτερικών μετρήσεων, ε) ο καθορισμός μετρήσιμων στόχων κατά την παραγωγή και η παρακολούθηση των βελτιώσεων από έκδοση σε έκδοσή του και στ) η δυνατότητα εστίασης σε συγκεκριμένα χαρακτηριστικά της ποιότητας και ο καθορισμός απαιτούμενων ενεργειών. Είναι μάλιστα προφανές ότι όσο νωρίτερα ένας κατασκευαστής λογισμικού μπορεί να εκτιμήσει την άποψη των χρηστών για την ποιότητα λογισμικού τόσο ευκολότερα μπορεί να προχωρήσει σε διορθωτικές ενέργειες αλλά και να καθορίσει ή να επαναπροσδιορίσει τις τάσεις που επικρατούν στην αγορά προϊόντων λογισμικού. Στις περισσότερες περιπτώσεις, αυτού του είδους οι εκτιμήσεις μπορούν να επιτευχθούν διεξάγοντας έρευνες στους ίδιους τους χρήστες.

Παρόλο που μπορεί αρχικά να φαίνεται ως μία πολύπλοκη μεθοδολογία, μπορεί εύκολα να αυτοματοποιηθεί και να προσαρμοστεί σύμφωνα με τη μέθοδο που θα επιλέξει το τμήμα ποιότητας της εταιρείας για τη συλλογή της γνώμης των χρηστών. Με άλλα λόγια, οι μεταβολές της γνώμης των χρηστών με την πάροδο του χρόνου δεν είναι ανάγκη να υπολογίζονται χειροκίνητα. Αντιθέτως, σύμφωνα με τους κανόνες που αναπτύχθηκαν στο κεφάλαιο 7, μπορεί να υλοποιηθεί μία κατάλληλη πλατφόρμα η οποία θα μπορεί να υπολογίζει αυτόματα την άποψη των χρηστών για την ποιότητα λογισμικού ανά πάσα χρονική στιγμή. Στην πλατφόρμα αυτή θα πρέπει να υποστηρίζονται τα προαναφερθέντα σύνολα περιορισμών και γενικότερα οι κανόνες Αναθεώρησης Άποψης. Οι λειτουργίες που θα εμπεριέχονται σε αυτή θα πρέπει σε γενικές γραμμές να ακολουθούν τη φιλοσοφία που παρουσιάζεται στο Σχήμα 6.3. Τα δεδομένα εισόδου θα αποτελούνται από την αρχική γνωστική κατάσταση ενός χρήστη, η οποία θα υπολογίζεται με τη βοήθεια μίας έρευνας βασισμένης σε ερωτηματολόγιο, και από κάθε νέο γνωστικό στοιχείο ή νέα πληροφορία που ανακαλύπτει ο χρήστης για το λογισμικό. Τα δεδομένα εξόδου θα δίνουν τη νέα γνωστική κατάσταση του χρήστη, μετά τις απαραίτητες αναθεωρήσεις οι οποίες θα υπολογίζονται αυτόματα.

Ιδιαίτερη βαρύτητα πρέπει να δίνεται στον καθορισμό των περιορισμών ανάμεσα στις τιμές των διαφόρων ποιοτικών χαρακτηριστικών, όπως αυτά τα αντιλαμβάνονται οι χρήστες και ιδιαίτερα οι χρήστες με χαμηλότερο βαθμό εμπειρίας. Επιπλέον, οι εταιρείες παραγωγής λογισμικού που θα εντάξουν την προτεινόμενη μεθοδολογία στο πρόγραμμα εξασφάλισης ποιότητάς τους έχουν τη δυνατότητα να την προσαρμόσουν σύμφωνα με τις δικές τους ανάγκες. Με άλλα λόγια, κατά τη φάση της ανάλυσης των αποτελεσμάτων που προκύπτουν από τη μεθοδολογία μπορούν να δοθούν διαφορετικά βάρη στα διάφορα ποιοτικά χαρακτηριστικά, ανάλογα με τους ποιοτικούς παράγοντες στους οποίους θέλει η κάθε εταιρεία να εστιάσει περισσότερο ή όχι. Σε αυτή την περίπτωση, τα νέα βάρη που θα προκύψουν θα πρέπει να ληφθούν υπόψη κατά την ταξινόμηση των πιθανών γνωστικών καταστάσεων των χρηστών και κατά τον καθορισμό των ελάχιστων δυνατών μεταβολών κατά τη μετάβαση από μία γνωστική κατάσταση σε μία άλλη.

Το μοντέλο που βασίζεται στις αρχές από τη θεωρία Αναθεώρησης Άποψης κατασκευάστηκε από μετρήσεις που έγιναν κατά τη διάρκεια της διατριβής, δηλαδή από ένα σύνολο από επαναλαμβανόμενες μετρήσεις για το κάθε υπό εξέταση προϊόν λογισμικού. Οι κανόνες του λοιπόν και η ορθότητά του επιβεβαιώθηκαν εκ των υστέρων, εφόσον ήταν ήδη γνωστά τα αποτελέσματα τα οποία έπρεπε να προκύψουν από τη χρήση του. Αυτό ήταν αναπόφευκτο στον αρχικό σχεδιασμό του, αφού με αυτό μόνο τον τρόπο θα μπορούσαν να καθοριστούν τα σύνολα περιορισμών ανάμεσα στις τιμές των ποιοτικών χαρακτηριστικών για την κάθε ομάδα χρηστών. Για τον υπολογισμό του ποσοστού επιτυχίας του μοντέλου αυτού και την πρακτική του επιβεβαίωση χρειάζεται πρώτα να υλοποιηθεί μία κατάλληλη πλατφόρμα που θα υπολογίζει αυτόματα τις αναθεωρήσεις της γνώμης των χρηστών, σύμφωνα με τους κανόνες που έχουν ήδη περιγραφεί. Επιπλέον χρειάζεται τόσο η εφαρμογή αυτής της πλατφόρμας για ένα εύλογο χρονικό διάστημα σε μία εταιρεία παραγωγής λογισμικού, καθώς και για ένα πλήθος από προϊόντα λογισμικού, όσο και η συνεχής και ενεργή συμμετοχή των χρηστών που θα συμμετέχουν στην έρευνα αυτή. Θα πρέπει να καθοριστούν οι ομάδες χρηστών με βάση την εμπειρία τους και τα σύνολα περιορισμών που θα ισχύουν στην κάθε ομάδα. Τέλος, θα πρέπει να λαμβάνεται υπόψη ότι με τον καιρό ένας χρήστης γίνεται όλο και πιο έμπειρος, δηλαδή ότι είναι δυνατή η μετακίνησή του από μία ομάδα σε μία άλλη, αλλά και ότι με την πάροδο του χρόνου οι απαιτήσεις του αυξάνουν.

Όσον αφορά τις εσωτερικές μετρήσεις, όσα αναφέρονται στη διατριβή σχετικά με τη συσχέτιση ανάμεσα στις εσωτερικές μετρικές και τα ποιοτικά χαρακτηριστικά όπως τα αντιλαμβάνεται ο χρήστης έχουν κατά καιρούς επιβεβαιωθεί από διάφορες περιπτωσιολογικές μελέτες, μερικές από τις οποίες παρουσιάστηκαν ήδη στα προηγούμενα κεφάλαια. Η κάθε εταιρεία παραγωγής λογισμικού πρέπει να επιλέγει τις κατάλληλες εσωτερικές μετρικές λογισμικού, ανάλογα με το έργο που πρόκειται να υλοποιηθεί, και να θέτει τα όρια που επιθυμεί στα αποτελέσματα αυτών των μετρικών. Το ύψος των αποδεκτών αυτών ορίων θα εξαρτάται προφανώς από το επίπεδο ωριμότητας της εταιρείας, τη φύση του εκάστοτε έργου, καθώς και τις εσωτερικές μετρικές που θα επιλεγθούν.

Η μεθοδολογία που παρουσιάζεται στη διατριβή μπορεί να συμπεριληφθεί και να εφαρμοστεί στο πρόγραμμα ποιότητας μίας εταιρείας παραγωγής λογισμικού. Εναλλακτικά, μπορεί να γίνει εστίαση σε συγκεκριμένα βήματα που ακολουθεί και να επιλεγούν οι αντίστοιχες από τις προτεινόμενες μεθόδους που την αποτελούν. Μάλιστα, η μεθοδολογία αυτή μπορεί να εφαρμοστεί και στη φάση της πρωτοτυποποίησης, όπου μπορεί να διεξαχθεί μία ανάλογη έρευνα για ένα καλό

πρωτότυπο και μετά να διερευνηθούν οι πιθανές μεταβολές που θα γίνουν στα επόμενα πρωτότυπα. Δηλαδή, δεν χρειάζεται να διεξαχθεί εκ νέου έρευνα για κάθε νέο πρωτότυπο που θα παραχθεί, αρκεί αυτό να μην έχει ριζικές και μεγάλου μεγέθους αλλαγές από το προηγούμενο από αυτό πρωτότυπο. Τέλος, παρόλο που για την αντιμετώπιση του προβλήματος της έγκαιρης εκτίμησης της γνώμης των χρηστών για την ποιότητα λογισμικού η παρούσα μεθοδολογία μπορεί να θεωρηθεί πλήρης και να εφαρμοστεί αυτούσια σε ένα πρόγραμμα ποιότητας, παρέχει επίσης τη δυνατότητα εμπλουτισμού της και με άλλες ερευνητικές μεθόδους. Αυτό φυσικά εξαρτάται και από τη φύση του λογισμικού που αναπτύσσεται. Για παράδειγμα, στην περίπτωση συστημάτων ηλεκτρονικού εμπορίου (e-commerce systems) έχει γίνει μελέτη για την ανάπτυξη ενός μοντέλου εκτίμησης της ποιότητας όπου θα περιορίζεται η υποκειμενικότητα της γνώμης των χρηστών. Το μοντέλο αυτό παρουσιάζεται στις δημοσιεύσεις [STF03], [STF04] και [STF05] και είναι βασισμένο στα Bayesian Networks. Συγκεκριμένα, σε αυτό η ποιότητα αναλύεται στα διάφορα ποιοτικά χαρακτηριστικά και υπο-χαρακτηριστικά σύμφωνα με το πρότυπο ISO 9126, ενώ στη συνέχεια σε αυτά αντιστοιχούνται τα διάφορα χαρακτηριστικά των συστημάτων ηλεκτρονικού εμπορίου.

Το μεγαλύτερο πλεονέκτημα της μεθοδολογίας που παρουσιάστηκε στην παρούσα διατριβή είναι η έγκαιρη εκτίμηση της αντίληψης των χρηστών για την ποιότητα λογισμικού χωρίς την ανάγκη διεξαγωγής επαναληπτικών μετρήσεων. Ως χρήστης με την ευρεία έννοια του όρου μπορεί να θεωρηθεί ο αποδέκτης των παραδοτέων οποιασδήποτε φάσης κατά την παραγωγή λογισμικού. Η μεθοδολογία αυτή λοιπόν όχι μόνο ενθαρρύνει τη συλλογή δεδομένων ανατροφοδότησης από τους εν δυνάμει χρήστες σε όλες τις φάσεις του κύκλου ζωής λογισμικού, αλλά επίσης υποστηρίζει και βελτιώνει τον τρόπο ανάλυσης και αξιοποίησης των δεδομένων αυτών.

Βιβλιογραφία

- [ABO00] Αβούρης Ν., “Εισαγωγή στην επικοινωνία ανθρώπου-υπολογιστή”, Δίαυλος, ISBN: 960-531-098-8, 2000.
- [ABR94a] Abreu F. and Carapuca R., “*Candidate Metrics for Object Oriented Software within a Taxonomy Framework*”, Journal of Systems and Software, Volume 26, No 1, 1994.
- [ABR94b] Abreu F., “*MOOD – Metrics for Object-Oriented Design*”, OOPSLA’94 Workshop, 1994.
- [AHE04] Ahern D., Clouse A. and Turner R., “*CMMI Distilled: A Practical Introduction to Integrated Process Improvement*”, 2nd Edition, Addison-Wesley, ISBN: 0-321-18613-3, 2004.
- [AIR91] U.S. Air Force, “*HQ AFSC/FMC. The Metrics Handbook*”, Andrews AFB, D.C., 1991.
- [ALB79] Albrecht A.J., “*Measuring application development productivity*”, Proc. Of IBM Applic. Dev, Joint SHARE/GUIDE Symposium, Monterey, CA, pp. 83-92, 1979.
- [ALC85] Alchourron C., Gardenfors P. and Makinson D. “*On the Logic of Theory Change: Partial Meet Contraction and Revision Functions*”, Journal of Symbolic Logic, 50(2), 510-530, 1985.
- [AME78] American Society for Quality Control, “*Standard A3*”, 1978.
- [AND96] Anderson I.P. and Ferneley E., “*User Centred Quality*”, 5th Software Quality Conference, University of Abertay, Dundee, 1996.
- [ARN95] Arnold K.L. and Holler M., “*Quality Assurance. Methods and Technologies*”, GLENCOE, McGraw-Hill, New York, 1995.
- [AUB85] Aubrey C., “*Quality Management in Financial Service*”, Wheaton, IL: Hitchcock Publishing, 1985.
- [BAC94] Bache R. and Bazzana G., “*Software Metrics for Product Assesment*”, McGraw-Hill, 1994.

- [BAN97] Bansiya J. and Davis C., “*Using QMOOD++ for object-oriented metrics*”, Dr. Dobb’s Journal, December, 1997.
- [BAP96] Βαρουφάκης Γ., “*Αρχαία Ελλάδα & Ποιότητα. Η ιστορία και ο έλεγχος των υλικών που σημάδεψαν τον ελληνικό πολιτισμό*”, Αίολος, ISBN: 960-7267-99-0, 1996.
- [BAS79] Basili V.R. and Reiter R.W., “*An Investigation of Human Factors in Software Development*”, IEEE Computer 12, 12, pp. 21-38, 1979.
- [BAS96] Basili V.R., Briand L.C. and Melo W.L., “*A Validation of Object-Oriented Design Metrics as Quality Indicators*”, IEEE Transactions on Software Engineering, SE-22, Volume 10, 1996.
- [BEL92] Bell D., Morrey I. and Pugh J., “*Software Engineering: A Programming Approach*”, 2nd ed., Prentice Hall, ISBN: 0-13-832536-7, 1992.
- [BEN99] Bellin D., Tyagi M. and Tyler M., “*Object Oriented Metrics: An Overview*”, Web Publication, 1999.
- [BEV97] Bevan N., “*Quality and usability: A new framework*”, van Veenendaal E. and McMullan J. (eds.), Achieving software product quality, Tutein Nolthenius, Netherlands, 1997.
- [BOE78] Boehm B.W., Brown J.R., Kaspar J.R., Lipow M., McCleod G.J. and Merrit M.J., “*Characteristics of Software Quality*”, North Holland, 1978.
- [BOY93] Boyd N., “*Building object-oriented frameworks*”, The Smalltalk report, Volume 3(1), pp. 1-16, 1993.
- [BRO91] Brown M.G., “*Baldrige Award Winning Quality: How to Interpret the Malcom Baldrige Award Criteria*”, Milwaukee, WI: ASQC Quality Press, 1991.
- [CAR83] Card S.K., Moran T.P. and Newell A., “*The Psychology of Human-Computer Interaction*”, Lawrence Erlbaum, Hillsdale, New Jersey, 1983.
- [CHI91] Chidamber R.S. and Kemerer C.F., “*Towards a metrics suite for object-oriented design*”, Proceedings of the OOPSLA ’91 Conference, pp. 197-211, 1991.
- [CHI94] Chidamber S.R. and Kemerer C.F., “*A Metrics Suite for Object Oriented Design*”, IEEE Transactions on Software Engineering, SE-20, Volume 6, pp. 476-493, 1994.
- [CON86] Conte S.D., Dunsmore H.E. and Shen V.Y., “*Software Engineering Metrics and Models*”, Benjamin Cummings, ISBN: 0-8053-2162-4, 1986.
- [CRO79] Crosby P., “*Quality is free*”, New York, McGraw-Hill, 1979.
- [DEM82] Deming W.E., “*Out of the Crisis*”, Cambridge University Press, 1982.

-
- [DMA82] DeMacro T., *“Controlling Software Projects”*, Prentice Hall, New York, 1982.
 - [DMA99] DeMacro T., *“Management Can Make Quality (Im)Possible”*, 6th European Conference on Software Quality, Vienna, 1999.
 - [DRE89] Dreger J.B., *“Function point analysis”*, Prentice-Hall, 1989.
 - [DRO95] Dromey R.G., *“A Model for Software Product Quality”*, IEEE Transactions on Software Eng., Vol. 21, 2, pp. 146-162, 1995.
 - [DRO96] Dromey R.G., *“Cornering the Chimera”*, IEEE Software, January 96, pp. 33-43, 1996.
 - [DUN79] Dunsmore H.E. and Cannon J.D., *“Data Referencing: An Empirical Investigation”*, IEEE Computer 12, 12, pp. 50-59, 1979.
 - [ΕΓΚ-ΠΑΠ] Εγκυκλοπαίδεια Πάπυρος-Λαρούς.
 - [EIT92] Eiter T. and Gottlob G., *“On the Complexity of Propositional Knowledge Base Revision, Updates and Counterfactuals”*, Artificial Intelligence, 59, pp. 227-270, 1992.
 - [FEI83] Feigenbaum V.A., *“Total Quality Control”*, 3rd ed., New York, McGraw-Hill, 1983.
 - [FEN92] Fenton N.E., *“Software Metrics. A Rigorous Approach”*, Chapman & Hall, ISBN: 0-442-31355-1, 1992.
 - [FEN94] Fenton N. and Pfleeger S., *“Software Metrics”*, PWS International Company, Thomson Computer Press, 1994.
 - [FEN97] Fenton N. and Pfleeger S., *“Software Metrics A Rigorous & Practical Approach”*, Thomson Computer Press, 1997.
 - [GAR88a] Gardenfors P., *“Knowledge in Flux – Modelling the Dynamics of Epistemic States”*, MIT Press, Cambridge, England, ISBN: 0-262-07109-6, 1988.
 - [GAR88b] Gardenfors P. and Makinson D., *“Revision of Knowledge Systems using Epistemic Entrenchment”*, Proceedings of the 2nd Conference on Theoretical Aspects of Reasoning about Knowledge, pp. 83-95, Morgan Kaufmann, 1988.
 - [GIL87] Gilb T., *“Principles of Software Engineering Management”*, Addison Wesley, 1987.
 - [GRA87] Grady R. and Caswell D., *“Software Metrics: Establishing a Company-Wide Program”*, Prentice Hall, 1987.
 - [GRO88] Grove A., *“Two modelings for theory change”*, Journal of Philosophical Logic, 17, pp. 157-170, 1988.
 - [HAL75] Halstead M.H., *“Elements of Software Science”*, Elsevier Publications, N-Holland, 1975.
 - [HAM96] Hambling B., *“Managing Software Quality”*, McGraw-Hill, London, 1996.
-

- [HAN97] Harrington J. and Mathers D., *“ISO 9000 and Beyond. From Compliance to Performance Improvement”*, McGraw-Hill, ISBN:0-07-026777-4, 1997.
- [HAR98] Harrison R., *“An Evaluation of the MOOD Set of Object-Oriented Software Metrics”*, IEEE Transactions on Software Engineering, SE-24, Volume 6, 1998.
- [HAS84] Harrison W.A., *“Software complexity metrics: a bibliography and category index”*, SIGPLAN Notices 19 (2), pp. 17-27, 1984.
- [HEN81] Henry S. And Kafura D., *“Software Structure Metrics based on Information Flow”*, IEEE Transactions on Software Engineering, SE-7, 5, pp. 510-518, 1981.
- [HUD94] Hudli R., Hoskins C. and Hudli A., *“Software Metrics for Object Oriented Designs”*, IEEE, 1994.
- [IEEE89] IEEE, *“Standard for a Software Quality Metrics Methodology”*, P-1061/D20, IEEE Press, New York, 1989.
- [ISO91] ISO, *“International technology - Evaluation of software - Quality characteristics and guides for their use”*, International Standard, ISO/IEC 9126, 1991.
- [ISO00] ISO/IEC 9001:2000, *“Quality management systems - Requirements”*, International Organization for Standardization, ISO/IEC 9001, 2000.
- [ISO01] ISO/IEC 9126, *“Software Product Evaluation – Quality Characteristics and Guidelines for the User”*, International Organization for Standardization, Geneva, 2001.
- [INC93] Ince D., *“An Introduction to Software Quality Assurance and Its Implementation”*, McGraw-Hill, Maidenhead, 1993.
- [INC94] Ince D., *“ISO 9001 and Software Quality Assurance”*, Quality Forum, McGraw-Hill, ISBN: 0-07-707885-3, 1994.
- [INC95] Ince D., *“Software Quality Assurance: A Student Introduction”*, McGraw-Hill, 1995.
- [JAV96] Javeau C., *“Η Έρευνα με Ερωτηματολόγιο. Το Εγχειρίδιο του Καλού Ερευνητή”*, μετάφραση Τζαννόνε-Τζωρτζή Κ., Τυπωθήτω, ISBN: 960-7643-33-X, 1996.
- [JON91] Jones C., *“Applied Software Measurement: Assuring Productivity and Quality”*, McGraw-Hill, ISBN: 0-07-032813-7, 1991.
- [JUR80] Juran J. and Gryna F., *“Quality Planning and Analysis”*, 2nd ed., New York, McGraw-Hill, 1980.
- [KAF87] Kafura D. and Reddy R., *“The Use of Software Complexity Metrics in Software Maintenance”*, IEEE Trans. Software Engineering, vol. SE-13, no. 3, pp. 335-343, 1987.
- [KAN96] Kan S., *“Metrics and Models in Software Quality Engineering”*, Addison-Wesley, 1996.

-
- [KAP95] Kaplan C., Clark R. and Tang V., *"Secrets of Software Quality"*, McGraw-Hill, ISBN: 0-07-911795-3, 1995.
 - [KER99] Kernighan B. and Pike P., *"The Practice of Programming"*, Addison Wesley, ISBN: 0-201-61586-X, 1999.
 - [KHO90] Khoshgoftaar T.M. and Munson J.C., *"Predicting software development errors using complexity metrics"*, IEEE Journal of Selected Areas in Communications, 8 (2), pp. 253-261, 1990.
 - [KIR86] Kirk J., Miller M. and Marc L., *"Reliability and Validity in Qualitative Research"*, Newbury Park, London, Sage Publications, 1986.
 - [KOL93] Kolewe R., *"Metrics in Object-Oriented Design and Programming"*, Software Development, October, 1993.
 - [KOP79] Kopetz H., *"Software Reliability"*, Springer-Verlag, p. 93, 1979.
 - [LIK32] Likert R., *"A Technique for the Measurement of Attitudes"*, Archives of Psychology, Vol. 140, p. 55, 1932.
 - [LOR91] Lorenz M., *"Real world reuse"*, Journal of object-oriented programming. Volume 6, 1991.
 - [LOR94] Lorenz M. and Kidd J., *"Object-Oriented Software Metrics"*, Prentice Hall, 1994.
 - [LYY88] Lyytinen K., *"Stakeholders, Information System Failures and Soft Systems Methodology: An Assessment"*, Journal of Applied Systems Analysis, pp. 61-81, 1988.
 - [MCB76] McCabe T.J., *"A complexity measure"*, IEEE Transactions in Software Engineering SE-2 (4), pp. 308-320, 1976.
 - [MCB94a] McCabe & Associates, *"McCabe Object-Oriented Tool User's Instructions"*, 1994.
 - [MCB94b] McCabe T.J., Dreyer L.A. et al., *"Testing an object-oriented application"*, Journal of the Quality Assurance Institute, pp. 21-27, October, 1994.
 - [MCC77] McCall J.A., Richards P.K. and Walters G.F., *"Factors in Software Quality, Vols I, II, III"*, US Rome Air Development Center Reports NTIS AD/A-049 014, 015, 055, 1977.
 - [MLS94] Miles M.B. and Huberman A.M., *"An Expanded Sourcebook Qualitative Data Analysis"*, London. Thousand Oaks, 1994.
 - [MOR88] Morris K.L., *"Metrics for Object-Oriented Software Development Environments"*, Master thesis, M.I.T., 1988.
 - [MUC68] Mucchielli R., *"Le questionnaire dans l' enquete psycho-sociale"*, Paris, Librairies Techniques/Edition Sociales Francaise, 1968.
 - [NEB91] Nebel B., *"Belief Revision and Default Reasoning: Syntax-Based Approaches"*, Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning, Morgan Kaufmann, 1991.
-

- [NEB94] Nebel B., *“Base Revision Operations and Schemes: Semantics, Representation and Complexity”*, Proceedings of the Eleventh European Conference on Artificial Intelligence, John Wiley & Sons, 1994.
- [NOR93] Northedge R., *“Plug Pulled on Taurus”*, Daily Telegraph 12th March 1993.
- [PAU93] Paulk M.C. et al., *“Capability Maturity Model, Version 1.1”*, IEEE Software, 10(4), pp. 18-27, 1993.
- [PAY51] Payne S.L., *“The Art of Asking Questions”*, New Jersey, Princeton University, 1951.
- [PEP95] Peppas P. and Williams M.A., *“Constructive Modellings for Theory Change”*, Notre Dame Journal of Formal Logic, vol. 36, Num. 1, pp. 120-133, 1995.
- [PEP96] Peppas P., *“Well Behaved and Multiple Belief Revision”*, European Conference on Artificial Intelligence, 1996.
- [PRE97] Pressman R.S., *“Software Engineering. A Practitioner’s Approach”*, 4th ed., McGraw Hill, ISBN: 0-07-709411-5, 1997.
- [ROM87] Rombach D., *“A Controlled Experiment on the Impact of Software Structure on Maintainability”*, IEEE Trans. Software Engineering, vol. SE-13, no. 3, pp. 344-354, 1987.
- [ROS97] Rosenberg L.H. and Hyatt L.E., *“Software Quality Metrics for Object-Oriented Environments”*, Crosstalk Journal, April, 1997.
- [ROY70] Royce W.W., *“Managing the Development of Large Software Systems”*, Proceedings WESTCON, USA, 1970.
- [SCH79] Schneidewind N.F. and Hoffman H., *“An Experiment in Software Error Data Collection and Analysis”*, IEEE Transactions on Software Engineering, SE-5, 3, pp. 276-286, 1979.
- [SHA93] Sharble R. and Cohen S., *“The Object-Oriented Brewery: A Comparison of Two Object-Oriented Development Methods”*, Software Engineering Notes, Volume 18, No 2, pp. 60-73, 1993.
- [SHE93] Shepperd M.J. and Ince D., *“Derivation and Validation of Software Metrics”*, Clarendon Press, Oxford, UK, 1993.
- [SHI97] Shih T.K., Chung C.M., Wang C.C. and Pai W.C., *“Decomposition of inheritance hierarchy DAGs for object-oriented software metrics”*, Workshop on ECBS, 1997.
- [SOM89] Sommerville I., *“Software Engineering”*, 3rd ed., Addison-Wesley, 1989.
- [SQM93] Software Quality Management, *“A Review of the Managing Software Quality in the 90’s”*, Conference by Elliott Marley, Issue 18, Spring, pp. 24-28, 1993.
- [STA95] Σταυρινούδης Δ., *“Πειραματική μελέτη του βαθμού συσχέτισης εσωτερικών και εξωτερικών μετρικών ποιότητας λογισμικού”*,

- Διπλωματική Εργασία, Τμήμα Μηχανικών Η/Υ και Πληροφορικής, Πανεπιστήμιο Πατρών, 1995.
- [STA99a] Stavrinoudis D., Xenos M., Peppas P. and Christodoulakis D., *“Measuring User’s Perception and Opinion of Software Quality”*, 6th European Conference on Software Quality, EOQ-SC, Vienna, 1999.
- [STA99b] Stavrinoudis D., Xenos M. and Christodoulakis D., *“Relation Between Software Metrics and Maintainability”*, FESMA 1999, Amsterdam, 1999.
- [STA00] Stavrinoudis D., Xenos M., Peppas P. and Christodoulakis D., *“Modelling Revisions of Users’ Perception of Software Quality”*, 2WCSQ, the 2nd World Congress on Software Quality, Yokohama, 2000.
- [STA05] Stavrinoudis D., Xenos M., Peppas P. and Christodoulakis D., *“Early estimation of users’ perception of software quality”*, Software Quality Journal, Springer Science + Business Media, Vol 13, Issue 2, pp. 155-175, 2005.
- [STE93] Steeples M.M., *“The Corporate Guide to the Malcom Baldrige National Quality Award”*, WI: ASQC Quality Press, 1993.
- [STF03] Stefani A., Xenos M. and Stavrinoudis D., *“Modelling E-Commerce Systems’ Quality with Belief Networks”*, IEEE International Conference on Virtual Environments, Human-Computer Interfaces and Measurement Systems, VECIMS, Lugano, 2003.
- [STF04] Stefani A., Stavrinoudis D. and Xenos M., *“Experimental Based Tool Calibration used for Assessing the Quality of E-Commerce Systems”*, 1st IEEE International Conference on E-Business and Telecommunication Networks, Setubal, Portugal, 2004.
- [STF05] Stefani A., Stavrinoudis D., Xenos M., *“Experimental Based Tool Calibration used for Assessing the Quality of E-Commerce Systems”*, In e-Business and Telecommunication Networks, Ascenso J., Vasiu L., Belo C., Saramago M., (Eds.), 2005, Selected (31 out of 166) papers from ICETE2004.
- [STR90] Strauss A., *“Basics of Qualitative Research Grounded Theory Procedures and Techniques”*, London, Newbury Park, Sage Publications, 1990.
- [STV47] Stevens S.S., *“On the theory of scales of measurement”*, Science, Volume 103, pp. 677-709, 1947.
- [TER96] Tervonen I., *“Support for Quality-Based Design and Inspection”*, IEEE Software, January 96, pp. 44-54, 1996.
- [THO94] Thomas B., *“The Human Dimension of Quality”*, McGraw Hill, ISBN: 0-07-709051-9, 1994.
- [TRO01] Trochim W., *“The Research Methods Knowledge Base”*, 2nd Edition, Atomic Dog Publishing, ISBN: 1-931442-48-7, 2001.

- [TSA86] Tsai W.T., Lopez M.A., Rodriguez V. and Volovik D., *“An Approach to Measuring Data Structure Complexity”*, Compsac86, pp. 240-246, 1986.
- [TSL90] Τσαλίδης Χ., *“Ολοκληρωμένη Μέτρηση Ποιοτικών Χαρακτηριστικών Λογικού Ανεξάρτητα από Γλώσσα Προγραμματισμού”*, Διδακτορική Διατριβή, Πανεπιστήμιο Πατρών, 1990.
- [TSL91] Tsalidis C., Christodoulakis D. and Maritsas D., *“Athena: A Software Measurement and Metrics Environment”*, Software Maintenance Research and Practice, 1991.
- [WES92] West M., *“An investigation of C++ metrics to improve C++ project estimation”*, IBM internal paper, 1992.
- [WEY88] Weyuker E., *“Evaluating Software Complexity Measures”*, IEEE Transactions on Software Engineering, SE-14 (9), pp. 1357-1365, 1988.
- [WOO79] Woodward M.R., Hennell M.A. and Hedley D., *“A measure of control flow complexity in program text”*, IEEE Transactions on Software Engineering, SE-5 (1), pp. 45-50, 1979.
- [XEN91] Ξένος Μ., *“Ερμηνεία Μετρικών”*, Διπλωματική Εργασία, Τμήμα Μηχανικών Η/Υ και Πληροφορικής, Πανεπιστήμιο Πατρών, 1991.
- [XEN94] Ξένος Μ. και Χριστοδουλάκης Δ., *“Τεχνολογία Λογισμικού. Αρχές και Μεθοδολογίες”*, Εκδόσεις Πανεπιστημίου Πατρών, 1994.
- [XEN95] Xenos M. and Christodoulakis D., *“Evaluating Software Quality by the Use of User Satisfaction Measurements”*, 4th Software Quality Conference, University of Abertay, Dundee, 1995.
- [XEN96a] Ξένος Μ., *“Μεθοδολογία Ελέγχου και Εξασφάλισης της Ποιότητας Λογισμικού Βασισμένη στις Μετρικές Προϊόντος και στα Εξωτερικά Ποιοτικά Χαρακτηριστικά του Λογισμικού”*, Διδακτορική Διατριβή, Πανεπιστήμιο Πατρών, 1996.
- [XEN96b] Xenos M., Stavrinooudis D. and Christodoulakis D., *“The Correlation Between Developer-oriented and User-oriented Software Quality Measurements (A Case Study)”*, 5th European Conference on Software Quality, EOQ-SC, Dublin, 1996.
- [XEN99] Xenos M., Stavrinooudis D. and Christodoulakis D., *“CESM: a Perceived Software Quality Assessment Tool”*, 4th International Conference on Software Process Improvement - Research into Education and Training, INSPIRE '99, Heraklion, Crete, Greece, 1999.
- [XEN00] Xenos M., Stavrinooudis D., Zikouli K. and Christodoulakis D., *“Object-Oriented Metrics – A Survey”*, FESMA 2000, Madrid, 2000.

- [XEN04a] Xenos M., Avouris N., Komis V., Stavrinoudis D. and Margaritis M., "*Synchronous collaboration in distance education: a case study in a Computer Science course*", 4th IEEE International Conference on Advance Learning Technologies, Joensuu, Finland, 2004.
- [XEN04b] Ξένος Μ., Αβούρης Ν., Κόμης Β., Σταυρινούδης Δ., Μαργαρίτης Μ., "*Εφαρμογή μεθόδων σύγχρονης συνεργασίας από απόσταση στο πρόγραμμα σπουδών του Ανοικτού Πανεπιστημίου*", Workshop "Ζητήματα Σχεδιασμού και Αξιολόγησης Σύγχρονων Συνεργατικών Συστημάτων Μάθησης με Υπολογιστές", στο πλαίσιο του 4ου Συνεδρίου της ΕΤΠΕ "Οι Τεχνολογίες της Πληροφορίας και της Επικοινωνίας στην Εκπαίδευση", Αθήνα, 2004.
- [XEN04c] Ξένος Μ., Αβούρης Ν., Κόμης Β., Σταυρινούδης Δ., Μαργαρίτης Μ., "*Εφαρμογή σύγχρονης τηλε-συνεργασίας στο Πρόγραμμα Σπουδών Πληροφορικής του ΕΑΠ*", Technical Report HOU-CS-TR-2004-03-GR, Ελληνικό Ανοικτό Πανεπιστήμιο, 2004.
- [YAU80] Yau S.S. and Collofello J.S., "*Some stability measures for software maintenance*", IEEE Transactions on Software Engineering, SE-6 (6), pp. 545-552, 1980.
- [YEH93] Yeh H.T., "*Software Process Quality*", McGraw Hill, ISBN: 0-07-072272-2, 1993.
- [ZAH95] Zahedi F., "*Quality Information Systems*", ITP, ISBN: 0-7895-0061-2, 1995.
- [ZIK00] Ζηκούλη Κ., "*Αντικειμενοστραφείς Μετρικές Λογισμικού*", Διπλωματική Εργασία, Τμήμα Μηχανικών Η/Υ και Πληροφορικής, Πανεπιστήμιο Πατρών, 2000.
- [ZUS90] Zuse H., "*Software Complexity: Measures and Methods*", de Gruyter, 1990.