

B. Αξιολόγηση Λογισμικού

Για την αξιολόγηση της πορείας του λογισμικού θα χρησιμοποιήσουμε μετρικές επιπέδου κλάσης, οι οποίες μας δίνουν πληροφορίες για:

- **το μέγεθος του κώδικα**

Η μέτρηση του μεγέθους του προγράμματος σε γραμμές κώδικα, μας δίνει πληροφορίες για το χρόνο και την προσπάθεια που έχει γίνει, προκειμένου το έργο να φτάσει στο εκάστοτε σημείο, στο οποίο λαμβάνεται η μέτρηση. Ενώ συνιστά τον πιο εύκολο τρόπο κοστολόγησης, δεν ανταποκρίνεται (πάντα) στην πραγματικότητα, κυρίως όσον αφορά στην ποιότητα, αλλά και τις συνθήκες μέσα από τις οποίες προέκυψε η τρέχουσα κατάσταση του κώδικα. Σίγουρα, οι γραμμές κώδικα, όντως αποτυπώνουν τον βαθμό προσπάθειας, όπως σε περιπτώσεις ελέγχου και εντοπισμού επίμονων σφαλμάτων. Από την άλλη μεριά, ένα σύνολο εντολών με δομές ελέγχου, σίγουρα χρειάζεται περισσότερο χρόνο υλοποίησης (σκέψη και συγγραφή) από ό,τι οι απλές εντολές εκτύπωσης της πορείας της εκτέλεσης του κώδικα. Οπότε κατά καιρούς χρησιμοποιήθηκαν εναλλακτικοί τρόποι υπολογισμού του μεγέθους του προγράμματος (συνδυασμός συναρτήσεων, τελεστών, μεταβλητών κλπ.). Αυτό που φαίνεται να ισχύει, τελικά, είναι το γεγονός ότι οι ίδιες πρακτικές δυσκολίες που συνηγορούν στην αποφυγή χρήσης του πλήθους γραμμών κώδικα, υπάρχουν και στις περιπτώσεις εναλλακτικών μονάδων μέτρησης.

Εμείς, στο πλαίσιο της εργασίας, θα χρησιμοποιήσουμε τη μονάδα μέτρησης **LOC (Lines of Code)**.

- **την συνεκτικότητα**

Η συνεκτικότητα μιας ενότητας (module), π.χ. κλάσης, του προγράμματός μας αντικατοπτρίζει τον βαθμό στον οποίο τα στοιχεία που βρίσκονται μέσα σε αυτή την ενότητα σχετίζονται μεταξύ τους. Κατά κανόνα, ενότητες με υψηλή συνεκτικότητα, είναι περισσότερο ισχυρές, πιο κατανοητές, πιο εύκολα συντηρήσιμες και επαναχρησιμοποιήσιμες. Αντίθετα, σε ενότητες που παρατηρείται χαμηλή συνεκτικότητα η κατανόηση και συντήρησή τους είναι δύσκολη, όπως και η επαναχρησιμοποίησή τους. Γενικά, κλάσεις με χαμηλή συνεκτικότητα, μπορούν να διαιρεθούν σε περισσότερες από μία κλάσεις, οι οποίες θα έχουν υψηλότερη συνεκτικότητα. Ως μονάδα μέτρησης, θα χρησιμοποιήσουμε την **LCOM (Lack of Cohesion of Methods)**, η οποία μας δείχνει αν έχω μεγάλη ή μικρή έλλειψη συνεκτικότητας μεταξύ των μεθόδων μιας κλάσης (Δύο μέθοδοι είναι συνεκτικές, εάν τα σύνολα των μελών δεδομένων που χρησιμοποιούν έχουν κοινά στοιχεία).

- **τη σύζευξη**

Η σύζευξη εκφράζει τον βαθμό αλληλεξάρτησης μεταξύ των ενοτήτων (π.χ. κλάσεων) ενός προγράμματος. Όσο λιγότερες είναι οι αλληλεξαρτήσεις μιας κλάσης, τόσο πιο εύκολος είναι ο έλεγχος, η τροποποίηση και η επαναχρησιμοποίησή της. Η σύζευξη είναι αντιστρόφως ανάλογη προς την συνεκτικότητα. Υψηλή συνεκτικότητα συνεπάγεται χαμηλή σύζευξη και το αντίστροφο. Θα χρησιμοποιήσουμε τη μονάδα μέτρησης **CBO (Coupling Between Objects)**. Αρχικά, η CBO υπολογιζόταν ως το άθροισμα του πλήθους των κλάσεων στις οποίες αναφέρεται μια κλάση και των κλάσεων οι οποίες αναφέρονται σε αυτήν, ενώ τελευταία, η πιο συνήθης ερμηνεία της αναφέρεται στον αριθμό των κλάσεων που παρέχουν σε μία κλάση Α τις απαραίτητες πληροφορίες (δεδομένα ή λειτουργίες) ώστε να ολοκληρώνονται οι μέθοδοί της.

- **Πολυπλοκότητα**

Για τον υπολογισμό της πολυπλοκότητας θα χρησιμοποιήσουμε τη μονάδα μέτρησης **WMC (Weighted Method Complexity)**. Χρησιμοποιούμε τον αριθμό των μεθόδων και την πολυπλοκότητά τους ως δείκτης της προσπάθειας που απαιτείται για την υλοποίηση και τον έλεγχο μιας κλάσης. Όσο αυξάνει ο αριθμός των μεθόδων μιας κλάσης τόσο πιο περίπλοκο γίνεται το δένδρο κληρονομικότητας και τόσο αυξάνει η συσχέτιση της κλάσης με μία συγκεκριμένη εφαρμογή, με συνέπεια να περιορίζεται η δυνατότητα επαναχρησιμοποίησης της κλάσης. Για τους λόγους αυτούς, η τιμή της μετρικής WMC θα πρέπει να παραμένει κατά το δυνατόν χαμηλή.

Για τον υπολογισμό των μετρικών χρησιμοποιήθηκε το λογισμικό **CodeMR**, ως plugin στο eclipse IDE. Η ανάλυση έγινε και στα 3 στιγμιότυπα του project, κάθε ένα από τα οποία αντιστοιχεί σε ένα Sprint (GymSupportFree [github branch **Free**], GymSupportPro [github branch **Pro**] και GymSupportRefactored [github **master** branch]). Επίσης, στο repository του master, υπάρχει φάκελος **CodeMR Analysis**, μέσα στον οποίο υπάρχει διαδραστική ανάλυση με γραφήματα, που μπορεί να προβληθεί με τη χρήση ενός web browser (υλοποιημένα σε html/javascript).

Οι τιμές των μετρικών για κάθε στάδιο του project είναι:

GymSupportFree

QualifiedName	Element	LOC	Coupling	Complexity	Size	Lack of Cohesion	CBO	WMC	LCOM
GSFree									
<Package>		1102	low	low	low-medium	low		123	
CreateAccountUI	CreateAccountUI	235	low	medium-high	low-medium	low	3	13	0,859
CreateUserFromJson	CreateUserFromJson	14	low-medium	low	low	low	6	4	0
GymSupportUI	GymSupportUI	360	low	medium-high	medium-high	low	5	19	0,915
LoginScreen	LoginScreen	117	low	medium-high	low-medium	low	3	6	0,75
MyProfile	MyProfile	115	low	medium-high	low-medium	low	1	4	0,2
SaveUserToJson	SaveUserToJson	22	low	low	low	low	3	4	0
User	User	38	low	low	low	low	0	10	0,889
UserList	UserList	27	low	low	low	low	1	15	0,6
Workout	Workout	174	low-medium	medium-high	low-medium	low	6	19	0,778
		1102					3,11	10,44	0,55

GymSupportPro

QualifiedName	Element	LOC	Coupling	Complexity	Size	Lack of Cohesion	CBO	WMC	LCOM
GSPro									
<Package>		1552	low	low-medium	medium-high	low		212	
CreateAccountUI	CreateAccountUI	235	low	medium-high	low-medium	low	3	13	0,85
CreateUserFromJson	CreateUserFromJson	15	low-medium	low	low	low	6	4	0,00
Exercise	Exercise	22	low	low	low	low	0	6	0,80
FullWorkout	FullWorkout	189	low-medium	medium-high	low-medium	low-medium	6	23	0,88
GymSupportUI	GymSupportUI	411	low-medium	medium-high	medium-high	low-medium	7	25	0,91
LoginScreen	LoginScreen	119	low	medium-high	low-medium	low	3	7	0,75
Middleware	Middleware	113	low-medium	low-medium	low-medium	low	8	29	0,84
MyProfile	MyProfile	115	low	medium-high	low-medium	low	1	4	0,20
PGClass	PGClass	37	low	low	low	low	0	10	0,88
SaveUserToJson	SaveUserToJson	23	low	low	low	low	3	4	0,00
Trainer	Trainer	18	low	low	low	low	0	5	0,75
User	User	54	low	low	low-medium	low	0	13	0,83
UserList	UserList	27	low	low	low	low	1	15	0,60
Workout	Workout	174	low-medium	medium-high	low-medium	low	6	19	0,78
		1552					3,14	12,64	0,65

GymSupportRefactored

QualifiedName	Element	LOC	Coupling	Complexity	Size	Lack of Cohesion	CBO	WMC	LCOM
GSRef									
<Package>GUI	GUI	1297	low-medium	low	low-medium	low	140		
GUI.CreateAccountUI	CreateAccountUI	235	low	medium-high	low-medium	low	3	13	0,85
GUI.FreeWorkout	FreeWorkout	172	low-medium	medium-high	low-medium	low	6	19	0,82
GUI.FullWorkout	FullWorkout	200	low-medium	medium-high	low-medium	low-medium	6	26	0,87
GUI.GymSupportUI	GymSupportUI	411	low-medium	medium-high	medium-high	low-medium	8	25	0,91
GUI.LoginScreen	LoginScreen	119	low	medium-high	low-medium	low	3	7	0,75
GUI.MyProfile	MyProfile	116	low	medium-high	low-medium	low	1	4	0,20
GUI.Workout	Workout	44	low	medium-high	low	low	2	9	0,00
<Package>gym	gym	123	low	low	low	low	40		
gym.Exercise	Exercise	24	low	low	low	low	0	7	0,80
gym.Trainer	Trainer	18	low	low	low	low	0	5	0,75
gym.User	User	54	low	low	low-medium	low	0	13	0,83
gym.UserList	UserList	27	low	low	low	low	1	15	0,60
<Package>middleware	middleware	210	low	low	low-medium	low	54		
middleware.CreateUserFromJson	CreateUserFromJson	15	low-medium	low	low	low	6	4	0,00
middleware.IMiddleware	IMiddleware	4	low	low	low	low	1	3	0,00
middleware.MiddlewareJSON	MiddlewareJSON	52	low-medium	low	low-medium	low	6	13	0,00
middleware.MiddlewarePostgreSQL	MiddlewarePostgreSQL	79	low	low	low-medium	low	4	20	0,84
middleware.PGClass	PGClass	37	low	low	low	low	0	10	0,88
middleware.SaveUserToJson	SaveUserToJson	23	low	low	low	low	3	4	0,00
1630							2,94	11,59	0,54

Εξέλιξη από Sprint 1 (GymSupportFree) σε Sprint 2 (GymSupportPro):

METRICS	Free	Pro	Refactored	investigated project	optimal class	diff	interest per LOC
WMC	10,44	12,64		12,64	10,44	2,2	21%
CBO	3,11	3,14		3,14	3,11	0,03	1%
LCOM	0,55	0,65		0,65	0,55	0,1	18%
SIZE	1102	1552		1552			13%
k=	450						

Εξέλιξη από Sprint 2 (GymSupportPro) σε Sprint 3 (GymSupportRefactored):

METRICS	Free	Pro	Refactored	investigated project	optimal class	diff	interest per LOC
WMC	10,44	12,64	11,59	11,59	10,44	1,15	11%
CBO	3,11	3,14	2,94	2,94	2,94	0	0%
LCOM	0,55	0,65	0,54	0,54	0,54	0	0%
SIZE	1102	1552	1630	1630			4%
k=	78						

Παρατηρούμε ότι από το Sprint 1 στο Sprint 2 έχουμε αισθητή αύξηση σε γραμμές κώδικα, ενώ συγχρόνως έχουμε αύξηση σε όλες τις μετρικές, κάτι που είναι αναμενόμενο, εφόσον υπήρχε ο στόχος να αυξηθεί σε μεγάλο βαθμό η λειτουργικότητα. Φυσικά, αυτό είχε ως συνέπεια (αναμενόμενη) την αύξηση των κλάσεων του project, και συνεπώς την πολυπλοκότητα. Επειδή ο στόχος του Sprint 2 ήταν να πολλαπλασιαστεί η λειτουργικότητα, ακόμη και με όχι βέλτιστη σχεδίαση, είχαμε ως επιπλέον συνέπεια την αύξηση της σύζευξης και την μείωση της συνεκτικότητας, σε επίπεδο κλάσεων. Κάτι που είχε προβλεφθεί να διορθωθεί στο Sprint 3.

Από το Sprint 2 στο Sprint 3, έχουμε σχετικά μικρή αύξηση σε γραμμές κώδικα, έχουμε όμως προχωρήσει σε μία μικρή αναδόμηση στον κώδικα, η οποία μας οδήγησε στην μείωση της πολυπλοκότητας, μείωση της σύζευξης και αύξηση της συνεκτικότητας. Σε σχέση, βέβαια, με το Sprint 1, η επίδοση παραμένει χαμηλότερη σχετικά με την πολυπλοκότητα, όμως η σύγκριση του Sprint 3 με το 1 δεν αξιολογείται, εφόσον στην πρώτη φάση η λειτουργικότητα απείχε πολύ από το επιθυμητό επίπεδο.

Παρακάτω βλέπουμε και την εξέλιξη των τιμών των μετρικών, ως προς τα Sprint:

