ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ

Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών

Μελέτη της εξέλιξης έργων ανοικτού λογισμικού

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Βασίλης Περικλέους

Επιβλέπων : Μπίμπη Σταματία

Λέκτορας Τ.Μ.Π.Τ.

Κοζάνη,2019

Τίτλος: Μελέτη της εξέλιξης έργων ανοικτού λογισμικού

Περιγραφή: Διπλωματική εργασία στα πλαίσια των σπουδών για την απόκτηση του Διπλώματος που απονέμει το Πανεπιστήμιο Δυτικής Μακεδονίας με τίτλο «*Μηχανικός Πληροφορικής και Τηλεπικοινωνιών*»

Θέματα / Λέξεις κλειδιά: Maven Repository, Open-Source, Ckjm, Statistical Analysis, Software Evolution, Lehman's Laws

Δημιουργός: Βασίλης Περικλέους

Χρόνος έκδοσης: 2019

Χώρα έκδοσης: GR

Γλώσσα κειμένου: Gre

ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ

Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών

Μελέτη της εξέλιξης έργων ανοικτού λογισμικού

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Βασίλης Περικλέους

Επιβλέπων : Μπίμπη Σταματία

Λέκτορας Τ.Μ.Π.Τ.

Εγκρίθηκε από την εξεταστική επιτροπή την …/…/2019 αποτελούμενη από τους:

Ονοματεπώνυμο:	Βαθμίδα:	Υπογραφή:
1. ……	…….	……
2. …….....	…….	…….....

Δήλωση Πνευματικών Δικαιωμάτων

Δηλώνω ρητά ότι, σύμφωνα με το άρθρο 8 του Ν. 1599/1986 και τα άρθρα 2,4,6 παρ.3 του Ν. 1256/1982, η παρούσα Διπλωματική Εργασία με τίτλο "Μελέτη της εξέλιξης έργων ανοικτού λογισμικού" καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας και αναφέρονται ρητώς μέσα στο κείμενο που συνοδεύουν, και η οποία έχει εκπονηθεί στο Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών του Πανεπιστημίου Δυτικής Μακεδονίας, υπό την επίβλεψη του μέλους του Τμήματος κα. Ματίνα Μπίμπη αποτελεί αποκλειστικά προϊόν προσωπικής εργασίας και δεν προσβάλλει κάθε μορφής πνευματικά δικαιώματα τρίτων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή / και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα. Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και μόνο.

Copyright (C) Ονοματεπώνυμο Φοιτητή & Επιβλέποντα, Έτος, Πόλη

Copyright (C) Βασίλης Περικλέους, Ματίνα Μπίμπη, 2019, Κοζάνη

Υπογραφή Φοιτητή

Ευχαριστίες

Θα ήθελα να ευχαριστήσω την επιβλέπουσα καθηγήτρια της διπλωματικής μου εργασίας, την Λέκτορα Μπίμπη Σταματία, του τμήματος Μηχανικών Πληροφορικής και Τηλεπικοινωνιών του Πανεπιστημίου Δυτικής Μακεδονίας, για την εξαιρετική συνεργασία μας καθώς και για την πολύτιμη βοήθεια και καθοδήγηση της καθ' όλη τη διάρκεια εκπόνησης της διπλωματικής μου εργασίας.

Τέλος, θέλω να ευχαριστήσω τους γονείς μου Ανδρέα και Λένια, τα αδέλφια μου Θωμά και Κυριάκο για την ολόψυχη αγάπη και υποστήριξή τους όλα αυτά τα χρόνια. Αφιερώνω αυτή την εργασία στην μητέρα μου και στον πατέρα μου που με βλέπει από ψηλά.

Βασίλης Περικλεούς

Κοζάνη, Μάρτιος 2019

Περίληψη

Στόχος της παρούσας διπλωματικής εργασία είναι η εξαγωγή και ανάλυση δεδομένων βιβλιοθηκών ανοικτού λογισμικού, με απώτερο σκοπό τη μελέτη της εξέλιξης τους ανάμεσα από τις διαδοχικές τους εκδόσεις, καθώς και την εξαγωγή συμπερασμάτων από αυτή.

Συγκεκριμένα συλλέχθηκαν διάφορα δεδομένα και μετρικές από εκατό είκοσι δυο έργα ανοικτού λογισμικού μέσω της πλατφόρμας Maven Repository και τη βοήθεια του εργαλείου στατιστικής ανάλυσης CKJM αντίστοιχα, εκ των οποίων στα τριάντα έγινε στατιστική ανάλυση με τη βοήθεια του στατιστικού προγράμματος XLSTAT.

Συνοψίζοντας, τα βασικά στοιχεία αυτής της διπλωματικής εργασίας, επικεντρώνονται σε τέσσερίς θεματικές ενότητες. Στη πρώτη ενότητα, γίνεται μια αναλυτική αναφορά και περιγραφή των όρων του αντικειμενοστραφή προγραμματισμού, του ανοικτού λογισμικού, της επαναχρησιμοποίησης και εξέλιξης έργων λογισμικού, των μετρικών κώδικα καθώς και του εργαλείου CKJM. Στη δεύτερη ενότητα γίνεται παρουσίαση των απαιτήσεων της εφαρμογής και των εργαλείων που χρησιμοποιήθηκαν για την υλοποίηση της. Στη τρίτη ενότητα παρουσιάζεται η διεξοδικά η εφαρμογή και η χρήση της με ακριβή παραδείγματα. Τέλος, η τέταρτη ενότητα περιέχει τη στατιστική ανάλυση που πραγματοποιήθηκε και την παρουσίαση των στατιστικών αποτελεσμάτων που προέκυψαν από αυτήν.

Λέξεις κλειδιά: Maven Repository, Open-Source, Ckjm, Statistical Analysis, Software Evolution, Lehman’s Laws

Abstract

The objective of this diploma thesis is the extraction and analysis of open Source Library data, with the aim of studying their evolution between successive editions, as well as the extraction of conclusions from it.

In particular, various data and metrics were collected from one hundred twenty two open-source projects through the Maven Repository platform and the help of the CKJM statistical analysis tool respectively, of which thirty were statistically analyzed with the help of the XLSTAT statistical program.

Summarizing the essential elements of this Diploma Thesis, focus on four main topics On the first axis, a detailed description and description of the terms of object oriented programming, open source software, reuse and development of software projects, metric code and CKJM tool is made. The second axis presents the requirements of the application and the tools used to implement it. The third axis presents the exhaustive application and its use with precise examples. Finally, the fourth axis contains the statistical analysis carried out and the presentation of the statistical results obtained from it.

Keywords: Maven Repository, Open-Source, Ckjm, Statistical Analysis, Software Evolution, Lehman's Laws

Περιεχόμενα

1. ΕΙΣΑΓΩΓΗ 17
1.1 Αντικείμενο Διπλωματικής 17
1.2 Οργάνωση Κεφαλαίων 18
2. ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ 19
2.1 Γενικά για τον Αντικειμενοστραφή προγραμματισμό 19
2.1.1 Πλεονεκτήματα του Αντικειμενοστρεφούς Προγραμματισμού 20
2.1.2 Μελέτη της αντικειμενοστρεφούς γλώσσας προγραμματισμού Java 21
2.2 Ανοικτό Λογισμικό (Open Source) 22
2.2.1 Τι είναι το Maven repository 23
2.3 Εξέλιξη έργων λογισμικού 24
2.3.1 Στόχοι της Μελέτης της Εξέλιξης έργων λογισμικού 25
2.3.2 Νόμοι του Lehman 26
2.4 Επαναχρησιμοποίηση έργων λογισμικού 27
2.4.1 Πλεονεκτήματα Επαναχρησιμοποίησης 28
2.4.2 Τύποι επαναχρησιμοποίησης 28
2.5 Το εργαλείο CKJM 29
2.5.1 Ιστορία του εργαλείου CKJM 29
2.6 Μετρικές κώδικα 30
2.6.1 Depth of Inheritance Tree (DIT) 30
2.6.2 Number of Children (NOC) 31
2.6.3 Coupling Between Object Classes (CBO) 31
2.6.4 Afferent coupling (Ca 32
2.6.5 Days Between Releases (DBR) 32
3. ΑΝΑΛΥΣΗ ΚΑΙ ΣΧΕΔΙΑΣΗ ΕΦΑΡΜΟΓΗΣ 32
3.1 Περιγραφή Εφαρμογής 32
3.2 Περιγραφή απαιτήσεων εφαρμογής 33
3.2.1 Διάγραμμα οντοτήτων-συσχετίσεων (Er-Model) 33
3.2.2 Πίνακες Βάσης Δεδομένων και Επεξήγηση 34
3.3 Εργαλεία και προγράμματα για την υλοποίηση της εφαρμογής 39
3.3.1 php (Hypertext Preprocessor 39
3.3.2 HTML (HyperTextMarkupLanguage) 40
3.2.3 SQL (Structured Query Language) 40

3.3.4 CSS (Cascading Style Sheets) 41
3.3.5 PhpMyAdmin 41
3.3.6 XAMPP 42
3.3.7 JavaScript 42
3.3.8 MySQL 42
3.3.9 Apache Web Server 43
4. ΠΑΡΟΥΣΙΑΣΗ ΕΦΑΡΜΟΓΗΣ 43
4.1 Επιμέρους τμήματα εφαρμογής 43
4.2 Χαρακτηριστικές εικόνες από την εκτέλεση της εφαρμογής 44
4.2.1 Επιλογές Home Page & Add New Project 45
4.2.2 Επιλογή Access 46
4.2.4 Επιλογή Export Project Data 48
4.2.5 Επιλογή Fetch Project Versions 49
4.2.6 Επιλογή CKJM Analyze 50
4.2.6.1 Επιλογή Analyze 51
4.2.6.2 Επιλογή Result 51
4.2.6.3 Επιλογή Charts 52
4.2.7 Επιλογή Export All Project Data 53
4.3 Προγράμματα σε Java που αναλύθηκαν 54
5. ΕΞΑΓΩΓΗ ΣΤΑΤΙΣΤΙΚΩΝ ΣΤΟΙΧΕΙΩΝ 70
5.1 Βοηθητικές πλατφόρμες 70
5.1.1 Microsoft Excel 70
5.2.2 XLSTAT 365 70
5.2 Στατιστικά Αποτελέσματα 71
6. ΕΠΙΛΟΓΟΣ 77
6.1 Σύγκριση με άλλες μελέτες και Συμπεράσματα 77
ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΑΝΑΦΟΡΕΣ 80

Κατάλογος Εικόνων

Εικόνα 1: Διάγραμμα οντοτήτων-συσχετίσεων……...34
Εικόνα 2: Στιγμιότυπο κεντρικής σελίδας της εφαρμογής ….……44
Εικόνα 3: Στιγμιότυπο επιλογής Home Page & Add New Project ….……...45
Εικόνα 4: Στιγμιότυπο επιλογής Access….……46
Εικόνα 5: Στιγμιότυπο επιλογής Collect Project Data……47
Εικόνα 6: Στιγμιότυπο επιλογής Export Project Data ……48
Εικόνα 7: Στιγμιότυπο αναμονής Fetch Project Versions ……...49
Εικόνα 8: Στιγμιότυπο επιτυχούς λήψης εκδόσεων ……49
Εικόνα 9: Στιγμιότυπο επιλογής CKJM Analyze ……50
Εικόνα 10: Στιγμιότυπο επιλογής Analyze……..51
Εικόνα 11: Στιγμιότυπο επιλογής Result ……51
Εικόνα 12: Στιγμιότυπο επιλογής Charts ……52
Εικόνα 13α: Στιγμιότυπο επιλογής Export All Project Data ……..53
Εικόνα 13β: Στιγμιότυπο επιλογής Export All Project Data ……..54
Εικόνα 14: asm……..……...55
Εικόνα 15: wiquery-core……..55
Εικόνα 16: commons-cli ……56
Εικόνα 17: struts2-core ……56
Εικόνα 18: guava ……57
Εικόνα 19: imagej ……57
Εικόνα 20: plexus-util ……58
Εικόνα 21: poi……..58
Εικόνα 22: guice-bean-binders ……59
Εικόνα 23: wro4j-core ……59
Εικόνα 24: xtream……60
Εικόνα 25: jdom……...60
Εικόνα 26: commons-lang3 ……61
Εικόνα 27: spring-core ……61
Εικόνα 28: joda-time ……..62
Εικόνα 29: jopt-simple ……62
Εικόνα 30: lucene-core ……63
Εικόνα 31: log4j-core ……..63

Εικόνα 32: commons-io ……..64
Εικόνα 33: slf4j-core ……..64
Εικόνα 34: itextpd ……..65
Εικόνα 35: superfly-remote-api ……..65
Εικόνα 36: geotk-utility ……..66
Εικόνα 37: javax.servlet-api ……66
Εικόνα 38: commons-logging ……67
Εικόνα 39: httpclient ……..67
Εικόνα 40: velocity ……68
Εικόνα 41: commons-math3 ……68
Εικόνα 42: avro ……..69
Εικόνα 43: httpcore ……69
Εικόνα 44: Γραφική Παράσταση (DBR) ……72
Εικόνα 45: Γραφική Παράσταση (DIT) ……..73
Εικόνα 46: Γραφική Παράσταση (CBO) ……74
Εικόνα 47: Γραφική Παράσταση (NOC)……75
Εικόνα 48: Γραφική Παράσταση (CA)……76

Κατάλογος Πινάκων

Πίνακας 1: Πεδία basic_info……35
Πίνακας 2: Πεδία project ……35
Πίνακας 3: Πεδία dependencies ……..36
Πίνακας 4: Πεδία repositories ……37
Πίνακας 5: Πεδία version ……..37
Πίνακας 6: Πεδία analysis ……38
Πίνακας 7: Αποτελέσματα 1ου Νόμου ……72
Πίνακας 8: Αποτελέσματα 2ου Νόμου ……73
Πίνακας 9: Αποτελέσματα 5ου Νόμου ……74
Πίνακας 10: Αποτελέσματα 6ου Νόμου……75
Πίνακας 11: Αποτελέσματα 7ου Νόμου……76
Πίνακας 12: Μετρικές σε προηγούμενες μελέτες……77
Πίνακας 13: Επικύρωση νόμων του Lehman σε προηγούμενες μελέτες ……78
Πίνακας 14: Αποτελέσματα σε έργα Java ……79

1. ΕΙΣΑΓΩΓΗ

1.1 Αντικείμενο Διπλωματικής

Η ικανότητα εξέλιξης λογισμικού γρήγορα και αξιόπιστα αποτελεί μείζονα πρόκληση για την τεχνολογία λογισμικού του σήμερα. Η ανάλυση εξέλιξης λογισμικού αφορά τις αλλαγές λογισμικού, τις αιτίες τους και τις επιπτώσεις τους. Χρησιμοποιεί όλες τις πηγές ενός συστήματος λογισμικού για να εκτελέσει μια αναδρομική ανάλυση. Τέτοια δεδομένα περιλαμβάνουν το πηγαίο κώδικα και τις πληροφορίες αλλαγής, τα δεδομένα αναφορών σφαλμάτων και τα δεδομένα που μπορούν να εξαχθούν από το τρέχον σύστημα. Ειδικότερα, η ανάλυση των δεδομένων και των σφαλμάτων έχει αποκτήσει μεγάλη σημασία επειδή αποθηκεύει πολύτιμες πληροφορίες για την ανάλυση της εξέλιξης του λογισμικού. Ενώ η ανάκτηση των δεδομένων που διαμένουν σε συστήματα εκδόσεων όπως το CVS ή το Subversion γίνεται ένα καλά διερευνημένο θέμα, η τελική πρόκληση έγκειται στην ανάκτηση δεδομένων και την ερμηνεία τους.

Με βάση τη σημαντικότητα της εξέλιξης έργων λογισμικού ο βασικός στόχος της παρούσας διπλωματικής εργασίας είναι η μελέτη της εξέλιξης τριάντα έργων λογισμικού ανοιχτού κώδικα τα οποία είναι γραμμένα σε γλώσσα προγραμματισμού Java. Για το σκοπό αυτό δημιουργήσαμε μια πλατφόρμας που λειτουργεί τοπικά σε ένα υπολογιστή η διαδικτυακά αν τοποθετηθεί σε έναν διακομιστή για να εξάγει δεδομένα από τη πλατφόρμα Maven Repository. Επίσης να αναλύει δεδομένα και πληροφορίες καθώς και να συλλέγει μετρικές μέσω του εργαλείου CKJM σε διαδοχικές εκδόσεις έργων λογισμικού ανοικτού κώδικα για να μελετηθεί η εξέλιξη των έργων σε βάθος χρόνου.

Ακολούθως γίνεται η διαδικασία της στατιστικής ανάλυσης μέσω του εργαλείου στατιστικής ανάλυση XLSTAT. Με την κατάλληλη επεξεργασία και μελέτη των δεδομένων, γίνεται η εξαγωγή συμπερασμάτων και στατιστικών στοιχείων.

Τα στατιστικά στοιχεία σχετίζονται άμεσα με τους νόμους του Lehman αν ισχύουν ή και όχι κατά πλειοψηφία, αν η εξέλιξη διαδοχικών εκδόσεων έργων ανοικτού λογισμικού έχουν κάποια τάση ή όχι και αν έχουν αν είναι ανοδική η καθοδική.

1.2 Οργάνωση Κεφαλαίων

Η παρούσα διπλωματική εργασία χωρίζεται σε έξι κεφάλαια-θεματικές ενότητες όπως αναφέρονται πιο κάτω.

Στο δεύτερο κεφάλαιο αρχικά παρουσιάζονται οι βασικές αρχές του αντικειμενοστραφή προγραμματισμού και γίνεται αναφορά στη μεθοδολογία ανάπτυξης ανοικτού λογισμικού. Στη συνέχεια περιγράφονται μετρικές και κανόνες για την παρακολούθηση της εξέλιξης και της επαναχρησιμοποίησης έργων λογισμικού.

Στο τρίτο κεφάλαιο καταγράφονται οι απαιτήσεις της εφαρμογής που υλοποιήθηκε για την παρακολούθηση της εξέλιξης έργων ανοικτού κώδικα καθώς και αναφέρονται τα εργαλεία που απαιτήθηκαν για την ανάπτυξη αυτής. Ακολούθως, γίνεται αναλυτική παρουσίαση της βάσης δεδομένων που χρησιμοποιήθηκε για την αποθήκευση των δεδομένων.

Στο τέταρτο κεφάλαιο γίνεται αναλυτική παρουσίαση της εφαρμογή βήμα προς βήμα για την ανάλυση και εξαγωγή των δεδομένων, μέσα από μερικά στιγμιότυπα.

Στο πέμπτο κεφάλαιο γίνεται μια προεπισκόπηση των προγραμμάτων που χρησιμοποιήθηκαν για την εξαγωγή των στατιστικών μας στοιχείων καθώς και παρουσίαση των αποτελεσμάτων που προέκυψαν. Επιπλέον, υπάρχουν στιγμιότυπα από στατιστικά αποτελέσματα τα οποία εξήχθησαν από αυτά.

Στο έκτο κεφάλαιο γίνεται παρουσίαση των συμπερασμάτων που εξήχθησαν, ακολούθως μια σύγκριση με άλλες μελέτες που έγιναν σε άλλες γλώσσες προγραμματισμού για τους νόμους του Lehman.

Τέλος, παρουσιάζονται όλες οι βιβλιογραφικές αναφορές και οι πηγές που βοήθησαν για την συγγραφή της διπλωματικής εργασίας.

2. ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ

2.1 Γενικά για τον Αντικειμενοστραφή προγραμματισμό

Στην πληροφορική Αντικειμενοστραφής προγραμματισμός (object-oriented programming), είναι ένας τρόπος οργάνωσης των προγραμμάτων που συνδέεται άμεσα με τα αντικείμενα και τις κλάσης. [1] Ένα χαρακτηριστικό των αντικειμένων είναι ότι οι διαδικασίες ενός αντικειμένου μπορούν να έχουν πρόσβαση και συχνά να τροποποιούν τα πεδία δεδομένων του αντικειμένου με το οποίο συσχετίζονται. Στον αντικειμενοστραφή προγραμματισμό, τα προγράμματα σχεδιάζονται κάνοντάς τα από αντικείμενα που αλληλοεπιδρούν μεταξύ τους. [2] [7]

Ο όρος αντικειμενοστραφής προγραμματισμός εμφανίστηκε στα τέλη της δεκαετίας του 1960 [3] και καθιερώθηκε κατά τη δεκαετία του 1990. Προέρχεται από τη γλώσσα προγραμματισμού Simula 67 [4] η οποία θεωρείται ως η πρώτη γλώσσα προγραμματισμού που χρησιμοποίησε αντικειμενοστραφή χαρακτηριστικά όπως κλάση (class), αντικείμενο (object), αφαίρεση (abstraction), κληρονομικότητα (inheritance), συσχέτιση (association), συναρμολόγηση (aggregation), ενθυλάκωση (encaplulation) και πολυμορφισμό (polymorphism). Την ίδια περίπου εποχή, και επίσης με επιρροές από τη Simula, ολοκληρωνόταν η ανάπτυξη της C++ ως μίας ισχυρής επέκτασης της δημοφιλούς γλώσσας προγραμματισμού C στην οποία είχαν "μεταμοσχευθεί" αντικειμενοστραφή χαρακτηριστικά. Η επιρροή της C++ καθ' όλη της δεκαετία του '80 ήταν καταλυτική με αποτέλεσμα τη σταδιακή κυκλοφορία αντικειμενοστρεφών εκδόσεων πολλών γνωστών διαδικαστικών γλωσσών προγραμματισμού. [11]

Πολλές από τις πιο ευρέως χρησιμοποιούμενες γλώσσες προγραμματισμού όπως C ++, Java, Python κ.λπ. είναι γλώσσες προγραμματισμού πολλών παραδειγμάτων που υποστηρίζουν αντικειμενοστρεφή προγραμματισμό σε μεγαλύτερο ή μικρότερο βαθμό. Σημαντικές αντικειμενοστραφές γλώσσες είναι Java, C ++, C #, Python, PHP, JavaScript, Ruby, Perl, Object Pascal, Objective-C, Dart, Swift, Scala, Common Lisp και Smalltalk.[4] [5] [9] [10]

Οι αιτίες που ώθησαν στην ανάπτυξη του ΑΠ ήταν οι ίδιες με αυτές που οδήγησαν στην ανάπτυξη του δομημένου προγραμματισμού (ευκολία συντήρησης, οργάνωσης, χειρισμού και επαναχρησιμοποίησης κώδικα μεγάλων και πολύπλοκων εφαρμογών), όμως τελικώς η

αντικειμενοστρέφεια επικράτησε καθώς μπορεί να ανταπεξέλθει σε προγράμματα πολύ μεγαλύτερου όγκου και πολυπλοκότητας. Πολλά από τα προβλήματα που υπήρχαν στην ανάπτυξη του λογισμικού εξακολουθούν να υπάρχουν και στον αντικειμενοστρεφή προγραμματισμό αλλά σε μικρότερο βαθμό.[3]

Ο Αντικειμενοστραφής προγραμματισμός μας παρέχει την δυνατότητα περιγραφής και αναπαράστασης των στοιχείων του πραγματικού χώρου ως αντικείμενα έτσι ώστε η επίλυση ενός προβλήματος να χρησιμοποιεί έννοιες οι οποίες πηγάζουν άμεσα από το πρόβλημα. Έτσι η περιγραφή του προβλήματος μέσω του προγράμματος γίνεται μέ τους όρους του προβλήματος και όχι της μηχανής. [61]

2.1.1 Πλεονεκτήματα του Αντικειμενοστρεφούς Προγραμματισμού

- Τα προγράμματα που γράφονται με αντικειμενοστραφη προγραμματισμό αναλύονται σε μικρότερα απλούστερα αντικείμενα, γεγονός που καθιστά τον κώδικα πιο εύκολο στη κατανοηση. [6] [60]
- Δεδομένου ότι όλα αντιμετωπίζονται ως αντικείμενα, μπορούμε να μοντελοποιήσουμε μια έννοια πραγματικού κόσμου.
- Προσφέρει επαναχρησιμοποίηση των κλάσεων. Μπορούμε να ξαναχρησιμοποιούμε τις κλάσης που έχουν ήδη δημιουργηθεί χωρίς να τα γράφουμε ξανά και ξανά.
- Δεδομένου ότι η παράλληλη ανάπτυξη των κλάσεων είναι εφικτή με αντικειμενοστραφη προγραμματισμό αυτό μας οδηγεί σε αύξηση της παραγωγικότητας των προγραμμάτων.
- Μείωση του χρόνου ανάπτυξης των προγραμμάτων.
- Τα προγράμματα που είναι γραμμένα με ΑΠ είναι ευκολότερα για δοκιμή, διαχείριση, διατήρηση και εντοπισμό σφαλμάτων στον κώδικα λόγω της καλά δομημένης δομής του.
- Είναι μια τεχνική ασφαλούς ανάπτυξης δεδομένου ότι τα δεδομένα είναι κρυμμένα και δεν είναι προσβάσιμα από εξωτερικές λειτουργίες.
- Τα αντικείμενα ως ανεξάρτητες δομές, μπορούν πλέον εύκολα να αντιστοιχηθούν με τις οντότητες του πραγματικού κόσμου, καθώς αυτές έχουν κατάσταση και συμπεριφορά.
- Η επεκτασιμότητα σε νέες λειτουργίες γίνεται σχετικά εύκολα με την χρήση των μηχανισμών της κελυφοποίησης και της κληρονομικότητας.

2.1.2 Μελέτη της αντικειμενοστρεφούς γλώσσας προγραμματισμού Java

Η γλώσσα προγραμματισμού Java είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού σχετικά υψηλού επιπέδου, γενικής χρήσης βασισμένη στις κλάσης . Έχει σχεδιαστεί έτσι ώστε να έχει όσο το δυνατόν λιγότερες εξαρτήσεις εφαρμογής [12] και να είναι αρκετά απλή για τους προγραμματιστές που θέλουν να τη μάθουν. Η γλώσσα προγραμματισμού Java μοιάζει συντακτικώς και σχετίζεται με τη C και τη C ++ αλλά είναι δομημένη διαφορετικά.

Σκοπός είναι να επιτρέψει στους προγραμματιστές εφαρμογών να γράψουν μια φορα και να μπορούν να τρέξουν το κώδικα οπουδήποτε ("write once, run anywhere" (WORA)) [13] σύμφωνα με την Sun Microsystems που κυκλοφόρησε την πρώτη δημόσια εφαρμογή ως Java 1.0 το 1996 [21]. Δηλαδή ο μεταγλωττισμένος κώδικας Java να μπορεί να εκτελεστεί σε όλες τις πλατφόρμες που υποστηρίζουν τη Java [13]. Οι εφαρμογές Java συνήθως μεταγλωττίζονται σε bytecode, δηλαδή μπορεί να εκτελεστεί σε οποιαδήποτε εικονική μηχανή Java (JVM) ανεξάρτητα από την αρχιτεκτονική του υπολογιστή.

Ο James Gosling, ο Mike Sheridan και ο Patrick Naughton ξεκίνησαν το πρόγραμμα γλώσσας Java τον Ιούνιο του 1991. [17] Η Java σχεδιάστηκε αρχικά για διαδραστική τηλεόραση, αλλά ήταν πολύ προηγμένη για τη βιομηχανία ψηφιακής καλωδιακής τηλεόρασης την εποχή εκείνη [18]. Η γλώσσα αρχικά ονομάστηκε Oak μετά από μια βελανιδιά που βρισκόταν έξω από το γραφείο του Gosling. Αργότερα το έργο πήρε το όνομα Green και τελικά μετονομάστηκε σε Java, από Java coffee [19]. Ο Gosling σχεδίασε τη Java με σύνταξη και στυλ C / C ++, το οποίο οι προγραμματιστές του συστήματος και των εφαρμογών θα έβρισκαν οικεία [20]. Τα μεγάλα προγράμματα περιήγησης ιστού ενσωμάτωσαν τη δυνατότητα εκτέλεσης μικροεφαρμογών Java σε ιστοσελίδες έτσι η Java γρήγορα έγινε ως μια από τις πιο δημοφιλής γλώσσες προγραμματισμού που χρησιμοποιούνται, για εφαρμογές ιστού πελάτη-διακομιστή [14] [15] [16]

2.2 Ανοικτό Λογισμικό (Open Source)

Στις πρώτες μέρες της πληροφορικής, οι προγραμματιστές μοιράστηκαν λογισμικό για να μάθουν ο ένας από τον άλλο και να εξελιχθούν στον τομέα της πληροφορικής. Η ιστορία του λογισμικού ανοιχτού κώδικα μπορεί να χρονολογηθεί από το 1950, στην ακαδημαϊκή κατανομή του Unix και στο λειτουργικό σύστημα GNU.[24] [26]

Ως όρος το Ανοικτό Λογισμικό ή Λογισμικό Ανοικτού Κώδικα (Open Source Software, OSS) είναι το λογισμικό του οποίου ο κάτοχος πνευματικών δικαιωμάτων του λογισμικού, παρέχει στους χρήστες το δικαιώματα μελέτης, τροποποίησης, αναβάθμισης του πηγαίου κώδικα καθώς και τη διανομή του λογισμικού σε οποιονδήποτε και για οποιονδήποτε σκοπό [22] [25].

Οι άδειες χρήσης του λογισμικού ανοικτού κώδικα προωθούν την συνεργασία και την ανταλλαγή, διότι επιτρέπουν σε όλους να κάνουν τροποποιήσεις στον πηγαίο κώδικα και να ενσωματώσουν αυτές τις αλλαγές σε δικά τους έργα.[23] Μερικές άδειες Ανοικτού Κώδικα διασφαλίζουν ότι τα άτομα που τροποποιούν και στη συνέχεια μοιράζονται ένα πρόγραμμα με τους άλλους, θα πρέπει να μοιράζονται και τον πηγαίο κώδικα αυτού του προγράμματος, χωρίς την επιβολή κάποιας μορφής αδειοδότησης για αυτό. Σύμφωνα με το Open Source Initiative, “open source” δεν σημαίνει μόνο πρόσβαση στον πηγαίο κώδικα αλλά κ ότι ο καθένας που τροποποιεί τον πηγαίο κώδικα για να το ταιριάξει στις απαίτησης του δεν πρέπει να εμποδίζει άλλους χρήστες-προγραμματιστές να κάνουν το ίδιο. [66][62]

Το λογισμικό ανοικτού κώδικα ξεκινάει συνήθως από κάποιον (φορέα ή ιδιώτη) ο οποίος η αρχική έκδοση του προγράμματος έχει γραφτεί από αυτόν. Ακολούθως, διαθέτει το συγκεκριμένο πηγαίο κώδικα της έκδοση του λογισμικού σε όλους όσους θέλουν να τον δουν, να τον αντιγράψουν, να τον τροποποιήσουν και να το βελτιώσουν προς προσαρμογή για τις δικές τους ανάγκες .

Το λογισμικό ανοικτού κώδικα δεν σημαίνει απαραιτήτως δωρεάν λογισμικό, ούτε ελεύθερο λογισμικό, αλλά αναφέρεται μόνο στο γεγονός πως επιτρέπει σε κάθε χρήστη να εξετάσει και να χρησιμοποιήσει τις δυνατότητες που προσφέρει ο πηγαίος κώδικας [63]. Υπάρχουν κάποιες σημαντικές διαφορές μεταξύ ελεύθερου λογισμικού και λογισμικού ανοικτού κώδικα όσον αφορά τα χαρακτηριστικά που εκχωρούνται στο αντίστοιχο λογισμικό, πώς πρέπει να αναπτυχθεί το λογισμικό για τη τροποποίηση, επαναχρησιμοποίηση και αναδιανομή [27].

Ένα μεγάλο μέρος του διαδικτύου είναι κτισμένο σε τεχνολογίες ανοικτού κώδικα, όπως το Linux και ο Apache, έτσι ένα μεγάλο ποσοστό χρηστών και προγραμματιστών επιλέγουν να χρησιμοποιούν λογισμικά ανοιχτού κώδικα για τους εξής λόγους: [63]

- Έχουν περισσότερο έλεγχο πάνω στο λογισμικό που χρησιμοποιούν.
- Μπορούν να εξετάσουν τον κώδικα του για να βεβαιωθούν ότι δεν κάνει κάτι χωρίς την θέληση τους.
- Μπορούν να το χρησιμοποιήσουν για οποιοδήποτε σκοπό θέλουν και όχι μόνο με τον τρόπο που κάποιος έχει καθορίσει.
- Συμβάλλει στην ανάπτυξη δεξιοτήτων για τους προγραμματιστές.
- Κοστίζουν λιγότερο η και καθόλου.
- Επειδή ο κώδικας είναι προσβάσιμος στο κοινό, οι μαθητές/φοιτητές μπορούν να μάθουν και να βελτιώνονται μελετώντας τι έχουν γράψει άλλοι.
- Είναι πολύ πιο εύκολος ο εντοπισμός και η διόρθωση σφαλμάτων ή παραλείψεων.
- Πολλοί προγραμματιστές μπορούν να εργαστούν σε ένα κομμάτι του λογισμικού ανοικτού κώδικα χωρίς να ζητήσουν την άδεια από τους αρχικούς δημιουργούς έτσι το λογισμικό ανοιχτού κώδικα είναι σταθερότερο, ενημερώνεται και αναβαθμίζεται πιο γρήγορα.
- Επειδή ο πηγαίος κώδικας του είναι ελεύθερος, όσοι βασίζονται σε λογισμικό για κρίσιμες εργασίες μπορούν να είναι βέβαιοι ότι τα εργαλεία τους δεν θα εξαφανιστούν ή δεν θα σταματήσουν να ενημερώνονται αν οι αρχικοί του δημιουργοί σταματήσουν να εργάζονται πάνω σε αυτό, ή αν κλείσει η εταιρία που τα στηρίζει.

2.2.1 Τι είναι το Maven repository

Το Maven repository είναι μια τοποθεσία, γενικά σε ένα σύστημα αρχείων (είτε απομακρυσμένο είτε τοπικό), όπου αποθηκεύονται και διαχειρίζονται τα αντικείμενα του Maven. [64] [65]

Ο τοπικός χώρος αποθήκευσης είναι ένας κατάλογος στον υπολογιστή όπου τρέχει το Maven. Καλύπτει απομακρυσμένες λήψεις και περιέχει προσωρινά αντικείμενα δημιουργίας που δεν έχουν ακόμα κυκλοφορήσει στην αγορά. Τα απομακρυσμένα αποθετήρια είναι διακομιστές ιστού που αναφέρονται σε οποιονδήποτε άλλο τύπο αποθετηρίου, στον οποίο έχουν πρόσβαση διάφορα

πρωτόκολλα όπως το HTTP και επιτρέπουν GET και PUT αιτήματα για δημοσίευση και ανάκτηση Maven Artifacts. [64] [65]

2.3 Εξέλιξη έργων λογισμικού

Η εξέλιξη έργων λογισμικού αναφέρεται στη διαδικασία με την οποία τα προγράμματα μπορούν να τροποποιηθούν και να προσαρμόζονται σε ένα μεταβαλλόμενο περιβάλλον. Η διαδικασία αυτή επικεντρώνεται στο τι συμβαίνει σε ένα σύστημα με την πάροδο του χρόνου και δίνει έμφαση στην τροποποίηση του. [59] [39] Η εξέλιξη του λογισμικού είναι όρος που εφαρμόζεται στην τεχνολογία λογισμικού και πιο συγκεκριμένα στον κλάδο συντήρησης λογισμικού που σχετίζεται με τη μέθοδο ανάπτυξης λογισμικού [29] . Όπως δηλώνει ο Fred Brooks στο βιβλίο του, πάνω από το 90% του κόστους σε ένα σύστημα λογισμικού δίνεται στο στάδιο συντήρησης [29]. Η συντήρηση λογισμικού χωρίζεται σε τέσσερις κύριες κατηγορίες: [30]

- Διορθωτική συντήρηση: Η επιδιόρθωση ενός προϊόντος λογισμικού που εκτελείται μετά την παράδοση για ικανοποίηση των απαιτήσεων.
- Προσαρμοσμένη συντήρηση: Η τροποποίηση ενός προϊόν λογισμικού που εκτελείται μετά την παράδοση, έτσι ώστε να διατηρηθεί το προϊόν λογισμικού χρήσιμο σε ένα μεταβαλλόμενο περιβάλλον.
- Συντήρηση έκτακτης ανάγκης: Μια μη προγραμματισμένη τροποποίηση του προϊόντος λογισμικού που εκτελείται μετά την παράδοση, για την προσωρινή διατήρηση ενός λειτουργικού συστήματος που εκκρεμεί διορθωτική συντήρηση.
- Προληπτική συντήρηση: Η τροποποίηση ενός προϊόντος λογισμικού μετά την παράδοση του για την ανίχνευση και διόρθωση σφάλματος στο προϊόν.

Η συντήρηση λογισμικού, επικεντρώνεται στην επιδιόρθωση σφαλμάτων λογισμικού καθώς και στην πραγματοποίηση μικρών βελτιώσεων στο σύστημα, χωρίς όμως αυτό να συνεπάγεται σημαντικές αλλαγές στην αρχιτεκτονική του συστήματος. Επειδή στη συντήρηση λογισμικού τα συστήματα συνδέονται στενά με το περιβάλλον τους, οι απαιτήσεις του συστήματος ενδέχεται να αλλάζουν κατά την ανάπτυξη του λογισμικού αν αλλάξει το περιβάλλον. Η συντήρηση των συστημάτων λογισμικού κρίνεται απαραίτητη ώστε να μπορούν να συμβαδίζουν με τις διαρκώς

μεταβαλλόμενες απαιτήσεις της αγοράς αλλά και να προσαρμόζονται σε διαφορετικά περιβάλλοντα λειτουργίας. [31]

Στην τεχνολογία λογισμικού, οι νόμοι της εξέλιξης του λογισμικού διατυπώθηκαν από τους Lehman και Belady το 1974 *[32] [33]* και παρουσιάστηκαν για πρώτη φορά στη διάλεξη του Lehman για στο Imperial College London το 1974 *[41]*. Οι νόμοι αυτοί καθορίζουν τι να προσέξουμε ώστε το έργο-λογισμικό να παραμείνει διαχρονικό στο πέρασμα των χρόνων. Τις τελευταίες δεκαετίες οι νόμοι αναθεωρήθηκαν και επεκτάθηκαν αρκετές φορές. *[34]*. Στο άρθρο του το 1980 ο Lehman προσδιόρισε την εφαρμογή τέτοιων νόμων διαχωρίζοντας τα σε τρεις κατηγορίες λογισμικού:[32]

- Ένα λογισμικό τύπου S (Specified) γράφεται σύμφωνα με τις ακριβείς προδιαγραφές του τι μπορεί να κάνει αυτό το πρόγραμμα, μια αλλαγή στην προδιαγραφή καθορίζει ένα νέο πρόβλημα, άρα ένα νέο πρόγραμμα. Τα προγράμματα τύπου S τυπικά επιλύουν τυπικά μαθηματικά προβλήματα. [37]
- Ένα λογισμικό τύπου P (Problem solving) είναι γραμμένο για την εφαρμογή ορισμένων διαδικασιών που καθορίζουν πλήρως τι μπορεί να κάνει το πρόγραμμα. Τα προγράμματα τύπου P επιχειρούν να λύσουν τα προβλήματα που μπορούν να δημιουργηθούν, αλλά δεν είναι υπολογιστικά προσιτά.[36]
- Ένα λογισμικό τύπου E (Evolutionary) είναι γραμμένο για συστήματα που χρησιμοποιούνται και ενσωματώνονται ενεργά σε έναν πραγματικό κόσμο [35]. Τα προγράμματα τύπου E επηρεάζονται από το περιβάλλον, έτσι πρέπει να είναι προσαρμοσμένα ανάλογα [31].

2.3.1 Στόχοι της Μελέτης της Εξέλιξης έργων λογισμικού

Οι κύριοι στόχοι της μελέτης της εξέλιξης του λογισμικού είναι η υλοποίηση και επικύρωση των πιθανών αλλαγών στο σύστημα καθώς και η διασφάλιση της αξιοπιστίας και της ευελιξίας του συστήματος. Ένα σύστημα πρέπει να εξελιχθεί μία φορά κάθε μερικούς μήνες ώστε να εξασφαλιστεί ότι προσαρμόζεται στο πραγματικό περιβάλλον με τις απαιτήσεις του σήμερα. Αυτό οφείλεται στην ταχεία ανάπτυξη του Παγκόσμιου Ιστού και των Διαδικτυακών Πόρων που διευκολύνουν τους χρήστες να βρουν σχετικές πληροφορίες. Ο θετικός αντίκτυπος σε αυτήν την

περίπτωση είναι ότι θα ανακαλυφθούν και θα δημιουργηθούν μεγάλες ποσότητες νέων ιδεών που στοχεύουν στη καλύτερη βελτίωση του συστήματος. [31] [38]

2.3.2 Νόμοι του Lehman

Οι νόμοι του Lehman αναφέρονται σε μια σειρά παρατηρήσεων σχετικών με την εξέλιξη των συστημάτων λογισμικού στην πάροδο του χρόνου. Ο Lehman αντιλαμβανόταν τη δυσκολία βελτίωσης ενός προγράμματος, έτσι δημιούργησε ορισμένους νόμους για να συνοψίσει τα συμπεράσματά και τις παρατηρήσεις του για συστήματα λογισμικού τύπου Ε.[35] Οι οκτώ νόμοι και η χρονιά διατύπωσης τους είναι οι εξής: [39] [41] [42] [43]

1) Συνεχής Αλλαγή (1974)

Τα σύστημα τύπου Ε πρέπει να προσαρμόζεται συνεχώς ανάλογα με τις ανάγκες του χρήστη και του περιβάλλοντος, διαφορετικά θα γίνονται σταδιακά λιγότερο ικανοποιητικά.

2) Αύξηση της Πολυπλοκότητας (1974)

Καθώς εξελίσσεται ένα σύστημα τύπου Ε, η πολυπλοκότητά του αυξάνεται αν δεν γίνει κάποια διεργασία για να διατηρηθεί ή να μειωθεί.

3) Αυτορρύθμιση (1974)

Η διαδικασία εξέλιξης συστημάτων τύπου Ε είναι αυτορρυθμιζόμενη και οδηγεί σε σταθερή διαδικασία .

4) Διατήρηση της Οργανωτικής Σταθερότητας (1978)

Το μέσο ενεργό ποσοστό παγκόσμιας δραστηριότητας σε ένα εξελισσόμενο σύστημα τύπου Ε είναι αμετάβλητο σε όλη τη διάρκεια ζωής του προϊόντος

5) Διατήρηση της Εξοικείωσης (1978)

Ένα σύστημα λογισμικού τύπου Ε έχει το ίδιο ή και λιγότερο νέο περιεχόμενο με την πάροδο του χρόνου σε σύγκριση με κάθε διαδοχική έκδοση.

6) Συνεχής Ανάπτυξη (1991)

Κατά τη διάρκεια της ζωής ενός συστήματος τύπου Ε, το λειτουργικό περιεχόμενο του πρέπει να αυξάνεται συνεχώς, ώστε να διατηρείται η ικανοποίηση του χρήστη

7) Φθίνουσα Ποιότητα (1996)

Η ποιότητα ενός συστήματος τύπου Ε μειώνεται αν δεν τηρηθούν αυστηρά οι αλλαγές του επιχειρησιακού περιβάλλοντος και δεν προσαρμόζονται σε αυτές.

8) Σύστημα Ανάδρασης (1996)

Οι διαδικασίες εξέλιξης τύπου Ε αποτελούν συστήματα πολλαπλών επιπέδων, πολλαπλών βρόχων, πολλαπλών παραγόντων και πρέπει να αντιμετωπίζονται ως έχουν ώστε να επιτυγχάνεται βελτίωση για άλλες από τις πιο πρωτόγονες διαδικασίες

2.4 Επαναχρησιμοποίηση έργων λογισμικού

Επαναχρησιμοποίηση λογισμικού είναι η εκμετάλλευση και η αξιοποίηση των γνώσεων υφιστάμενων στοιχείων λογισμικού κατά τη διαδικασία δημιουργίας νέου λογισμικού ή ενσωμάτωσης υπάρχοντων στοιχείων λογισμικού σε νέο λογισμικό.[44]

Η ιδέα της επαναχρησιμοποίησης έργων λογισμικού δεν είναι νέα αφού οι ρίζες της χρονολογούνται από το 1968 [45] [47] όταν ο Douglas McIlroy παρουσίασε την εργασία του για την επαναχρησιμοποίηση στοιχείων λογισμικού στη διάσκεψη του NATO στο Garmisch της Γερμανίας. Πιο συγκεκριμένα, ο McIlroy πρότεινε τη χρήση ήδη υπάρχοντος λογισμικού, ή την εκμετάλλευση της γνώσης που προκύπτει από αυτό, με σκοπό τη δημιουργία νέου λογισμικού.[46]

Ωστόσο, υπήρξε περιορισμένη πρακτική εμπειρία σχετικά με τις μεθόδους επαναχρησιμοποίησης μέχρι τα τέλη της δεκαετίας του 1980, όταν υιοθετήθηκαν μεγάλης κλίμακας τεχνικές επαναχρησιμοποίησης σε προγράμματα που υλοποιήθηκαν από εταιρείες, κυρίως των ΗΠΑ (π.χ. IBM και Hewlett Packard) και την Ιαπωνία (π.χ. από την Toshiba και Fujitsu).[46]

2.4.1 Πλεονεκτήματα Επαναχρησιμοποίησης

Χρησιμοποιώντας την επαναχρησιμοποίηση λογισμικού μειώνεται ο χρόνος και το κόστος ανάπτυξης λογισμικού με άμεσο αντίκτυπο την αύξηση της παραγωγικότητας. Αν και απαιτείται χρόνος για να εκτιμηθεί η δυνατότητα εφαρμογής ενός συγκεκριμένου επαναχρησιμοποιήσιμου στοιχείου σε ένα νέο λογισμικό ή προϊόν, ο χρόνος αυτός είναι ελάχιστος σε σύγκριση με τον χρόνο ανάπτυξης ενός καινούριου λογισμικού.[48] [49] [50]

Κάθε επιτυχής επαναχρησιμοποίηση ενός στοιχείου αυξάνει το επίπεδο αξιοπιστίας, της ποιότητας του προϊόντος, αυξάνει τη χρησιμότητά του στην αποθήκη επαναχρησιμοποίησης και μειώνει τον κίνδυνο σφαλμάτων. Επίσης, καθώς οι διαδικασίες επαναχρησιμοποιούνται, συσσωρεύεται περισσότερη εμπειρία στον τομέα αυτό και η επαναχρησιμοποίησης λογισμικού γίνεται περισσότερο γνωστή και υψηλότερη ακρίβεια. [48] [49] [50]

Με τη διαδικασία επαναχρησιμοποίησης, κάθε νέο σύστημα μπορεί να κατασκευαστεί από ένα σύνολο βασικών στοιχείων παρά να δημιουργηθεί από το μηδέν με αποτέλεσμα το λογισμικό να δημιουργηθεί πιο γρήγορα και το τελικό προϊόν να έχει ταχύτερο χρόνο διάθεσης στην αγορα (time-to-market) [48] [49] [50]

2.4.2 Τύποι επαναχρησιμοποίησης

- **Οριζόντια επαναχρησιμοποίηση**

Η οριζόντια επαναχρησιμοποίηση αναφέρεται σε στοιχεία λογισμικού που χρησιμοποιούνται σε μεγάλη ποικιλία εφαρμογών. Επίσης αναφέρεται για χρήση μιας εμπορικής εφαρμογής off-the-shelf (COTS) ή τρίτου μέρους σε ένα μεγαλύτερο σύστημα, όπως ένα πακέτο ηλεκτρονικού ταχυδρομείου ή ένα πρόγραμμα επεξεργασίας κειμένου. Μια ποικιλία βιβλιοθηκών λογισμικού

και χώρων αποθήκευσης που περιέχουν οριζόντια επαναχρησιμοποίηση υπάρχουν σήμερα σε διάφορες τοποθεσίες στο Διαδίκτυο.[51]

- **Κατακόρυφη επαναχρησιμοποίηση**

Η κατακόρυφη επαναχρησιμοποίηση, δεν αξιοποιείται σημαντικά από την κοινότητα λογισμικού αν είναι πολύ χρήσιμη έτσι υπάρχουν σημαντικές επιπτώσεις για τις τρέχουσες και μελλοντικές προσπάθειες ανάπτυξης λογισμικού. Η βασική ιδέα στη κατακόρυφη επαναχρησιμοποίηση είναι η επαναχρησιμοποίηση των λειτουργικών περιοχών συστημάτων, ή των τομέων να μπορούν να χρησιμοποιηθούν από μια οικογένεια συστημάτων με παρόμοια λειτουργία. [51]

2.5 Το εργαλείο CKJM

Το CKJM (Chidamber and Kemerer Java Metrics) είναι ένα εργαλείο το οποίο χρησιμοποιείται για τον υπολογισμό βασικών μετρικών ποιότητας κώδικα ο οποίος είναι γραμμένος σε γλώσσα προγραμματισμού Java. Είναι ένα εργαλείο ανοικτού κώδικα το οποίο υπολογίζει κάποιες μετρικές που θα αναλύσουμε εκτενέστερα στο παράρτημα 2.7

Το εν λόγω εργαλείο υπολογίζει τις αντικειμενοστραφείς μετρήσεις Chidamber και Kemerer επεξεργάζοντας και αναλύοντας τους δυαδικούς κώδικες των αρχείων Java που έχουν μεταγλωττιστεί. Συγκεκριμένα, δέχεται σαν είσοδο μια κλάση ή ένα σύνολο κλάσεων και εξάγει ένα αρχείο XML με τα αποτελέσματα της ανάλυσης. Το αρχείο αυτό περιέχει μια μοναδική εγγραφή για κάθε κλάση του έργου που αναλύθηκε η οποία περιέχει όλες τις μετρικές που υπολογίστηκαν για αυτή την κλάση. [52]

2.5.1 Ιστορία του εργαλείου CKJM

Το εργαλείο Ckjm είναι του Διομήδη Σπινέλλη και Πρόεδρος του Τμήματος Διοικητικής Επιστήμης και Τεχνολογίας στο Οικονομικό Πανεπιστήμιο Αθηνών, αρχισυντάκτης του επιστημονικού περιοδικού IEEE Software και διευθυντής του εργαστηρίου Επιχειρηματικής Αναλυτικής (BALab).

Έγραψε αυτό το πρόγραμμα όταν διαπίστωσε ότι υπήρχε έλλειψη αξιόπιστων προγραμμάτων για τον υπολογισμό των αντικειμενοστραφών μετρήσεων του Chidamber και του Kemerer. Τα προγράμματα που βρήκε ήταν ελλιπή, δηλαδή υπολόγιζαν μόνο μερικές από τις μετρήσεις, ήταν αναξιόπιστα γιατί υπολόγισαν προφανώς λάθος αποτελέσματα ή ήταν εξαιρετικά αναποτελεσματικά δηλαδή απαιτούσαν GBs RAM και αρκετές ώρες επεξεργασίας. Το Ckjm ακολουθεί την παράδοση των Unix και κάνει τα πράγματα πιο εύκολα αφού κάνει αυτή τη δουλειά σχολαστικά και αποτελεσματικά: σε μια μηχανή 1,6 GHz Pentium-M , έκδοση 1.1 επεξεργάστηκε τα 33MB των αρχείων jar Eclipse 3.0 (κλάσης 19717) σε 95 δευτερόλεπτα.[53]

2.6 Μετρικές κώδικα

Οι μετρικές που χρησιμοποιήσαμε και η επεξήγηση τους είναι από την αναφορά του Διομήδη Σπινέλλη: ckjm - A Tool for Calculating Chidamber and Kemerer Java Metrics [52] και από το άρθρο των Shyam R. Chidamber και Chris F. Kemerer: A metrics suite for object-oriented design [54].

Μετρικές είναι διάφορα χαρακτηριστικά από τα οποία, ανάλογα με το βαθμό που έχουν για ένα συγκεκριμένο έργο εξάγονται κάποιες πληροφορίες σχετικά με αυτά, με σκοπό η εξέλιξη του έργου να είναι όσο το δυνατό με λιγότερα σφάλματα.

Ποιο κάτω θα περιγράψουμε τις πέντε μετρικές που χρησιμοποιήσαμε για την ανάλυση των έργων μας.

2.6.1 Depth of Inheritance Tree (DIT)

Η μετρική DIT υπολογίζεται ως η απόσταση που έχει μια κλάση από την ρίζα του δέντρου.

Από αυτή την μετρική εξάγονται τρία συμπεράσματα:

- Όσο πιο βαθιά είναι μια κλάση στο δέντρο της ιεραρχίας, τόσο μεγαλύτερος θα είναι ο αριθμός των μεθόδων που κληρονομεί με αποτέλεσμα η μεγάλη τιμή DIT να αυξάνει την πολυπλοκότητα της σχεδίασης, την πιθανότητα σφαλμάτων αλλά και την επαναχρησιμοποίηση κώδικα μέσω κληρονομικότητας.

- Δέντρα με μεγαλύτερο βάθος σχετίζονται με μεγαλύτερη πολυπλοκότητα στη σχεδίαση, αφού μετέχουν περισσότερες κλάσεις και μέθοδοι.
- Όσο πιο βαθιά είναι μια κλάση στην ιεραρχία, τόσο μεγαλύτερη είναι η πιθανότητα χρήσης κληρονομημένων μεθόδων

2.6.2 Number of Children (NOC)

Η μετρική NOC υπολογίζεται ως ο αριθμός των κλάσεων που κληρονομούν τα χαρακτηριστικά μιας κλάση. Μηδενική τιμή της μετρικής σημαίνει ότι η συγκεκριμένη κλάση δεν επαναχρησιμοποιείται.

Από αυτή την μετρική εξάγονται τρία συμπεράσματα:

- Όσο μεγαλύτερη είναι η τιμή της NOC, τόσο πιο διαδεδομένη είναι η επαναχρησιμοποίηση κώδικα μέσω κληρονομικότητας.
- Όσο μεγαλύτερη είναι η τιμή της NOC, τόσο αυξάνονται οι πιθανότητες να υπάρχει κακός σχεδιασμός και εσφαλμένη χρήση κληρονομικότητας.
- Εάν μια κλάση έχει μεγάλο αριθμό υποκλάσεων δηλαδή μεγάλη τιμή NOC, ενδέχεται να έχει μεγάλο ποσοστό επιρροής στο συνολικό πρόγραμμα και επομένως θα πρέπει να ετοιμαστούν περισσότερα σενάρια ελέγχου, αυξάνοντας έτσι τον χρόνο δοκιμής της κλάσης αυτής

2.6.3 Coupling Between Object Classes (CBO)

Η μετρική CBO υπολογίζεται ως το σύνολο των κλάσεων από τις οποίες συνδέεται η τρέχουσα κλάση. Δύο κλάσεις συνδέονται μεταξύ τους (έχουν σύζευξη), όταν κάποια από τις μεθόδους μιας κλάσεις, χρησιμοποιεί τις μεθόδους της άλλης κλάσης. Αν μια κλάση καλεί μεθόδους ή τοπικές μεταβλητές μιας άλλης κλάσης, τότε η τιμή της CBO αυξάνεται κατά ένα.

Από αυτή την μετρική εξάγονται τρία συμπεράσματα:

- Υψηλή τιμη CBO σημαίνει μεγάλη σύζευξη μεταξύ δύο κλάσεων, κάτι που δεν είναι επιθυμητό λόγω του ότι γίνεται πιο δύσκολα η επαναχρησιμοποίηση.

- Όσο χαμηλότερη είναι η τιμή CBO μίας κλάσης τόσο πιο μικρή είναι η ευαισθησία στις αλλαγές και το πρόγραμμα είναι πιο εύκολο να συντηρηθεί.
- Η CBO είναι χρήσιμη για να τον καθορισμό της πολυπλοκότητας των ελέγχων που πραγματοποιούνται σε μία κλάση.

2.6.4 Afferent coupling (Ca)

Η μετρική Ca υπολογίζεται ως ο αριθμός του πόσες άλλες κλάσεις χρησιμοποιούν τη συγκεκριμένη κλάση. Οι υψηλές τιμές της μετρικής συνήθως υποδηλώνουν υψηλή σταθερότητα των στοιχείων των κλάσεων. Αυτό οφείλεται στο γεγονός ότι η κλάση εξαρτάται από πολλές άλλες κλάσεις. Η σύζευξη είναι μία από τις κλασικές εσωτερικές μετρήσεις που χρησιμοποιούνται για την αξιολόγηση της ποιότητα ενός έργου. [28]

2.6.5 Days Between Releases (DBR)

Η μετρική DBR υπολογίζεται ως η διαφορά μεταξύ των ημερομηνιών δημιουργίας μιας έκδοσης ενός έργου από την άλλη. Με άλλα λόγια, η DBR ποσοτικοποιεί τη συχνότητα με την οποία δημοσιεύονται νέες κυκλοφορίες. Μια αύξηση του DBR με την πάροδο του χρόνου σημαίνει ότι ο ρυθμός δημοσίευσης νέων εκδόσεων μειώνεται. [28]

3. ΑΝΑΛΥΣΗ ΚΑΙ ΣΧΕΔΙΑΣΗ ΕΦΑΡΜΟΓΗΣ

3.1 Περιγραφή Εφαρμογής

Σκοπός της εφαρμογής, η οποία υλοποιήθηκε στα πλαίσια της διπλωματικής μου εργασίας είναι η εξαγωγή και η ανάλυση μετρικών από έργα τα οποία είναι καταχωρημένα στη πλατφόρμα

Maven Repository (http://mvnreposirory.com) για να δούμε κατά πόσο υπάρχει διαφορά στις μετρικές των έργων από έκδοση σε έκδοση και να εξαχθούν κάποια συμπεράσματα.

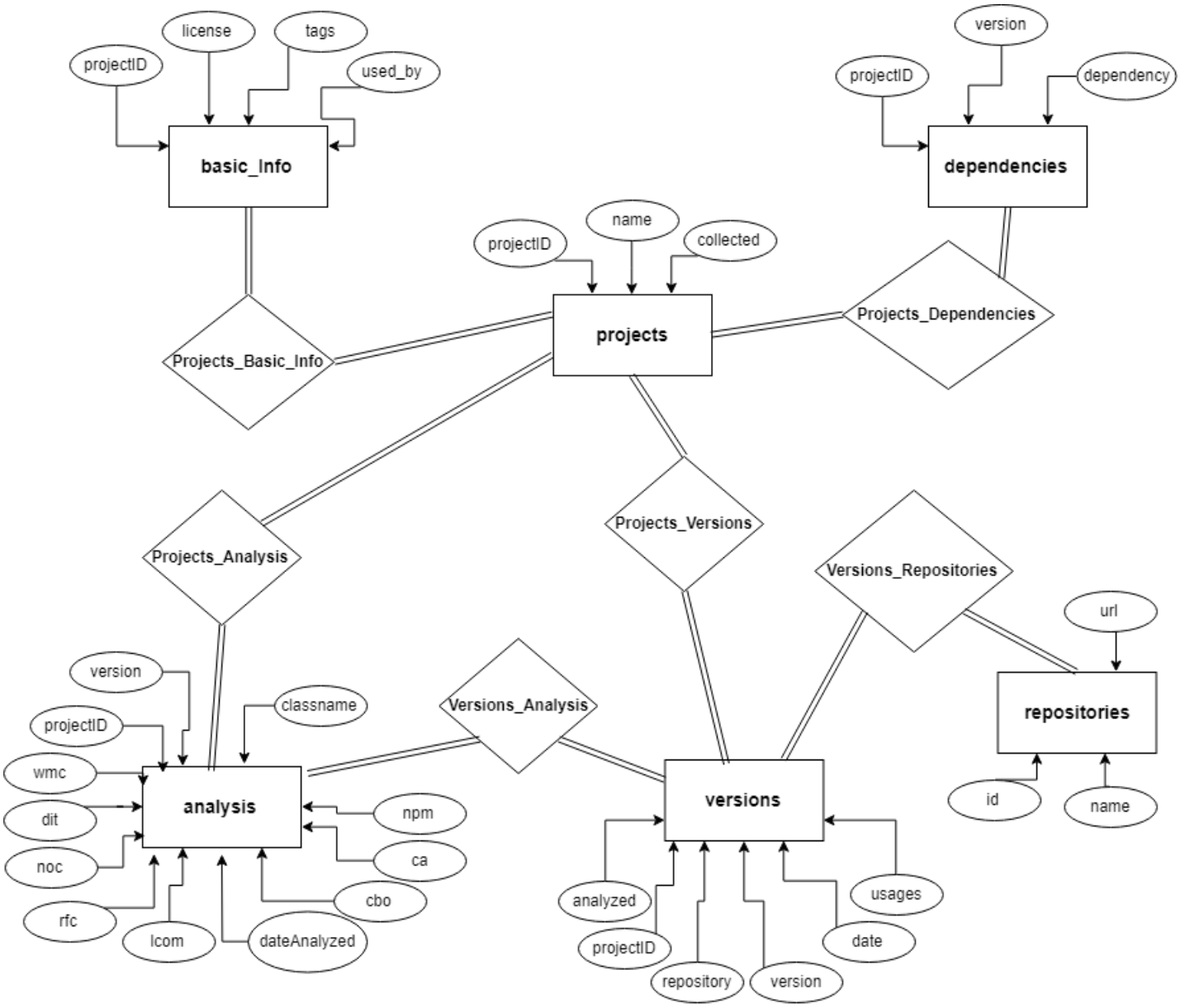
3.2 Περιγραφή απαιτήσεων εφαρμογής

3.2.1 Διάγραμμα οντοτήτων-συσχετίσεων (Er-Model)

Το μοντέλο οντοτήτων-συσχετίσεων (μοντέλο Ο/Σ - ER model) είναι ένα αφαιρετικό ιδεατό μοντέλο δεδομένων, το οποίο έχουν καθορισμένη δομή. Προτάθηκε αρχικά το 1976 από τον Peter Chen, ωστόσο στη συνέχεια επινοήθηκαν πάρα πολλές παραλλαγές της διαδικασίας. [67]

Χρησιμοποιείται στο πρώτο στάδιο σχεδίασης ενός συστήματος πληροφοριών, κατά την ανάλυση των απαιτήσεών του. Σκοπός του είναι να περιγράφει τις αναγκαίες πληροφορίες οι οποίες πρόκειται να αποθηκευτούν στη βάση δεδομένων ή τον τύπο τους. Η μοντελοποίηση δεδομένων γίνεται για την περιγραφή των χρησιμοποιούμενων όρων και των σχέσεων τους σε έναν ορισμένο τομέα ενδιαφέροντος. [67]

Στην εικόνα 1 βλέπουμε το διάγραμμα οντοτήτων-συσχετίσεων που σχεδιάστηκε για την υλοποίηση της εφαρμογής. Στο διάγραμμα απεικονίζονται οι πίνακες που χρειάστηκε να δημιουργηθούν, τα στοιχεία των πινάκων αυτών, καθώς και οι συσχετίσεις μεταξύ των έξι πινάκων.



Εικόνα 1: Διάγραμμα οντοτήτων-συσχετίσεων

3.2.2 Πίνακες Βάσης Δεδομένων και Επεξήγηση

Η βάση δεδομένων που δημιουργήσαμε περιέχει 6 πίνακες εκ των οποίων η περιγραφή γίνεται παρακάτω.

❖ Πίνακας basic_info

Όνομα Μεταβλητής	Τύπος	Περιγραφή
projectID	int	Ταυτότητα έργου
licence	varchar	Άδεια χρήσης
tags	varchar	Ετικέτα έργου
used_by	Varchar	Αρ. έργων που το χρησιμοποιούν

Πίνακας 1: Πεδία basic_info

Ο πίνακας 1 αποτελείται από τα εξής πεδία:

- Το projectID είναι η ταυτότητα του κάθε έργου καθώς και το πρωτεύον κλειδί του πίνακα πχ : 1, 2 ,3 …
- Το license είναι από που παρέχεται η άδεια χρήσης για το κάθε έργο πχ. BSD, Apache 2.0, LGPL 2.1
- Το tags μας λέει την κατηγορία η τις κατηγορίες που βρίσκεται το έργο πχ. bytecode, command-line, cli, parser, apache, typesafe
- Το used_by είναι ο αριθμός των έργων που χρησιμοποιούν το συγκεκριμένο έργο που αναλύουμε πχ. 15

❖ Πίνακας project

Όνομα Μεταβλητής	Τύπος	Περιγραφή
projectID	int	Ταυτότητα έργου
name	text	Όνομα έργου
collected	enum	Συλλογή έργου

Πίνακας 2: Πεδία project

Ο πίνακας 2 αποτελείται από τα εξής πεδία:

- Το projectID είναι η ταυτότητα του κάθε έργου καθώς και το πρωτεύον κλειδί του πίνακα πχ. 1, 2 ,3
- Το name είναι το όνομα κάθε έργου πχ. asm, poi, guava, imagej…
- Το collected μας δείχνει με “y” αν το έργο έχει συλλεχθεί και με “n” αν το έργο δεν έχει συλλεχθεί

❖ Πίνακας dependencies

Όνομα Μεταβλητής	**Τύπος**	**Περιγραφή**
projectID	int	Ταυτότητα έργου
Version	varchar	Έκδοση έργου
dependency	varchar	Εξαρτόμενα έργου

Πίνακας 3: Πεδία dependencies

Ο πίνακας 3 αποτελείται από τα εξής πεδία:

- Το projectID είναι η ταυτότητα του κάθε έργου καθώς είναι και το πρωτεύον κλειδί του πίνακα πχ. 1, 2 ,3
- Το version είναι η αναθεωρημένη βελτιωμένη έκδοση του κάθε έργου πχ. 3,1 5,2, 6.1-beta
- Το dependency είναι τα έργα που εξαρτάτε το συγκεκριμένο έργο που αναλύουμε πχ. asm-test, junit-jupiter-api, junit-jupiter-engine

❖ Πίνακας repositories

Όνομα Μεταβλητής	Τύπος	Περιγραφή
Id	int	Ταυτότητα αποθήκης
name	varchar	Όνομα αποθήκης
url	varchar	Διεύθυνση αποθήκης

Πίνακας 4: Πεδία repositories

Ο πίνακας 4 αποτελείται από τα εξής πεδία:

- Το id είναι η ταυτότητα της κάθε αποθήκης πχ. 1, 2, 3….
- Το name είναι το όνομα της κάθε αποθήκης πχ. Central, Redhat GA, Spring Plugins
- Το url είναι η διεύθυνση που βρίσκετε η αποθήκη πχ. http://repo.spring.io/pluins-release/

❖ Πίνακας version

Όνομα Μεταβλητής	Τύπος	Περιγραφή
projectID	int	Ταυτότητα έργου
version	varchar	Έκδοση έργου
repositories	varchar	Όνομα αποθήκης
analyzed	enum	Ανάλυση έργου
usages	varchar	Αρ. έργων
date	date	Ημ/νια έκδοσης έργου

Πίνακας 5: Πεδία version

Ο πίνακας 5 αποτελείται από τα εξής πεδία:

- Το projectID είναι η ταυτότητα του κάθε έργου καθώς είναι και το πρωτεύον κλειδί του πίνακα πχ. 1, 2, 3
- Το version είναι η αναθεωρημένη βελτιωμένη έκδοση του κάθε έργου πχ: 4.0, 5.0_ALPHA, 5.0.1..
- Το repository το όνομα της αποθήκης πχ. Central
- Το analyzed μας δείχνει με "y" αν το έργο έχει αναλυθεί και με "n" αν το έργο δεν έχει αναλυθεί
- Το usages είναι ο αριθμός των διαφόρων έργων που χρησιμοποιούν το τρέχων έργο που βλέπουμε πχ.70, 112, 408
- Το date είναι η ημερομηνία έκδοσης του έργου πχ. 2014-03-01

❖ Πίνακας analysis

Όνομα Μεταβλητής	**Τύπος**	**Περιγραφή**
projectID	int	Ταυτότητα έργου
version	varchar	Έκδοση έργου
classname	varchar	Όνομα κλάσης
wmc	int	Μετρική
dit	int	Μετρική
noc	int	Μετρική
cbo	int	Μετρική
rfc	int	Μετρική
lcom	int	Μετρική
ca	int	Μετρική
npm	int	Μετρική
dateAnalyzed	datetime	Ημ/νια ανάλυσης

Πίνακας 6: Πεδία analysis

Ο πίνακας 6 αποτελείται από τα εξής πεδία:

- Το projectID είναι η ταυτότητα του κάθε έργου καθώς είναι και το πρωτεύον κλειδί του πίνακα πχ. 1, 2, 3
- Το version είναι η αναθεωρημένη βελτιωμένη έκδοση του κάθε έργου πχ: 4.0 , 4.3, 5.2
- Το classname είναι το όνομα της κλάσης πχ. org.objectweb.asm.AnnotationVisitor
- Το wmc είναι μετρική μέτρησης των μεθόδων που ορίζονται σε μια κλάση.
- Το dit είναι μετρική υπολογισμού βάθους μια κλάσης
- Το noc είναι μετρική μέτρησης των κλάσεων που κληρονομούν τη κλάση που αναλύεται στο συστήματος.
- Το cbo είναι μετρική υπολογισμού σύζευξης μεταξύ των αντικειμένων των κλάσεων
- Το rfc είναι μετρική υπολογισμού των αριθμών των μεθόδων που μπορεί να εκτελεστούν ως απόκριση σε ένα μήνυμα
- Το lcom είναι μετρική υπολογισμού έλλειψης συνεκτικότητας μεταξύ των μεθόδων μιας κλάσης
- Το ca είναι μετρική μέτρησης του πόσες άλλες κλάσεις χρησιμοποιούν τη συγκεκριμένη κλάση.
- Το npm είναι μετρική μέτρησης όλων των δημοσίων μεθόδων που έχουν δηλωθεί σε μια κλάση
- Το dateAnalyzed η ημερομηνία που έχει αναλυθεί το έργο πχ.2018-05-23

3.3 Εργαλεία και προγράμματα για την υλοποίηση της εφαρμογής

3.3.1 php (Hypertext Preprocessor)

Η PHP (Hypertext Pre-processor) είναι μια ευρέως χρησιμοποιούμενη γλώσσα προγραμματισμού ανοιχτού κώδικα που είναι κατάλληλη για ανάπτυξη και δημιουργία ιστοσελίδων καθώς γράφεται συνήθως πλαισιωμένη από HTML, για μορφοποίηση των αποτελεσμάτων.

Η PHP αποτελεί μια από τις πιο διαδεδομένες γλώσσες προγραμματισμού στο Παγκόσμιο Ιστό, καθώς χρησιμοποιείται από πληθώρα εφαρμογών και ιστότοπων. Είναι μια γλώσσα η οποία βρίσκεται δικαίως σχεδόν σε κάθε διακομιστή για το λόγο ότι ο κώδικας PHP μπορεί να θέσει ερωτήματα σε βάσεις δεδομένων, να διαβάσει και να γράψει αρχεία, να συνδεθεί με απομακρυσμένους υπολογιστές κλπ. [68]

Η PHP ξεκίνησε το 1994 από τον φοιτητή Rasmus Lerdorf δημιουργώντας ένα απλό script για προσωπική του χρήση με όνομα php.cgi, χρησιμοποιώντας τη γλώσσα προγραμματισμού C. Το script διατηρούσε μια λίστα στατιστικών για τα άτομα που έβλεπαν το online βιογραφικό του σημείωμα. Μεταγενέστερα αυτό το script το έδωσε σε φίλους του, οι οποίοι άρχισαν να του ζητούν να προσθέσει περισσότερες δυνατότητες. Η γλώσσα τότε ήταν γνωστή ως PHP/FI από τα αρχικά Personal Home Page/Form Interpreter. [69]

3.3.2 HTML (HyperTextMarkupLanguage)

Τα αρχικά HTML προέρχονται από τις λέξεις HyperText Markup Language δηλαδή γλώσσα σήμανσης υπερκειμένου και είναι η κύρια γλώσσα σήμανσης για τις ιστοσελίδες, αφού τα στοιχεία της είναι τα βασικά δομικά στοιχεία των ιστοσελίδων. Ως πρώτη και πιο διαδεδομένη γλώσσα περιγραφής της δομής μιας ιστοσελίδας χρησιμοποιεί τις ειδικές ετικέτες για να δώσει οδηγίες στον περιηγητή. Είναι ένα πολύ σημαντικό εργαλείο για τη δημιουργία μίας ιστοσελίδας καθώς με τις εντολές της μας βοηθάει να τοποθετούμε τα κείμενα, τις εικόνες, τα links κλπ στο site μας όπως θέλουμε, να τα μορφοποιούμε και να τα προβάλουμε με τον τρόπο που έχουμε προγραμματίσει. [70]

3.2.3 SQL (Structured Query Language)

Η SQL (Structured Query Language) είναι μία γλώσσα για τις βάσεις δεδομένων και σχεδιάστηκε για τη διαχείριση δεδομένων σε συστήματα διαχείρισης σχεσιακών βάσεων δεδομένων η οποία, αρχικά, βασίστηκε στη σχεσιακή άλγεβρα. Είναι μία σχετικά απλή γλώσσα υψηλού επιπέδου που περιλαμβάνει δυνατότητες ανάκτησης, ενημέρωσης δεδομένων, δημιουργίας και τροποποίησης σχημάτων και σχεσιακών πινάκων, αλλά και ελέγχου πρόσβασης στα δεδομένα. [71]

3.3.4 CSS (Cascading Style Sheets)

Η CSS (Cascading Style Sheets) είναι μια γλώσσα υπολογιστή που χρησιμοποιείται για την μορφοποίηση εγγράφων που έχουν γραφτεί με μια γλώσσα σήμανσης όπως είναι η HTML. Μπορεί να χρησιμοποιηθεί για να ορίσουμε το στυλ κειμένου, τα μεγέθη των πινάκων και άλλες πτυχές των ιστοσελίδων που προηγουμένως δεν μπορούσαν να οριστούν παρά μόνο στον κώδικα HTML μιας σελίδας. Το CSS βοηθάει τους προγραμματιστές ιστοσελίδων να δημιουργήσουν μια ομοιόμορφη εμφάνιση σε αρκετές σελίδες μιας ιστοσελίδας αφού σε ένα αρχείο CSS μπορούν να οριστούν διαστάσεις, χρώματα, στοίχιση και άλλες εντολές μορφοποίησης. [72]

3.3.5 PhpMyAdmin

Το phpMyAdmin είναι ένα εργαλείο ανοικτού κώδικα, γραμμένο σε PHP κώδικα, με σκοπό τη διαχείριση της MySQL με τη χρήση ενός προγράμματος περιήγησης. Το phpMyAdmin υποστηρίζει ένα ευρύ φάσμα λειτουργιών στις υπηρεσίες MySQL και MariaDB. Έχει αρκετά πλεονεκτήματα όπως : [73]

- Όμορφο web interface
- Διαχείριση βάσης δεδομένων MySql
- Εισαγωγή δεδομένων από το αρχείο CSV και SQL
- Εξαγωγή δεδομένων σε διάφορες μορφές: CSV, SQL, XML, PDF, ISO, Word, LATEX κ.α.
- Μπορεί να εκτελεί διάφορες εργασίες όπως την δημιουργία, τροποποίηση ή διαγραφή βάσεων δεδομένων, πινάκων, πεδίων ή γραμμών.
- Διαχείριση πολλών διακομιστών
- Pdf γραφικά της βάσης δεδομένων
- Δημιουργία σύνθετων ερωτημάτων χρησιμοποιώντας QBE (Query-by-example)
- Αναζήτηση βάσης δεδομένων σε παγκόσμιο πεδίο
- Ενεργή εποπτεία ερωτημάτων
- Εκτελεί SQL ερωτήματα
- Διαχειρίζεται τους χρήστες και τις άδειες-δικαιώματα τους
- Είναι διαθέσιμο σε 72 γλώσσες για τη διευκόλυνση της χρήσης του.

3.3.6 XAMPP

Το Xampp είναι ένα ελεύθερο εργαλείο ανάπτυξης και δοκιμής ιστοσελίδων τοπικά στον υπολογιστή χωρίς να είναι απαραίτητη η σύνδεση στο διαδίκτυο. Πρόκειται για μια απλή, ελαφριά διανομή Apache που καθιστά εξαιρετικά εύκολο για τους προγραμματιστές να δημιουργήσουν έναν τοπικό διακομιστή web για σκοπούς δοκιμών Το ακρωνύμιό του είναι Cross-Platform (X), Apache (A), MySql (M), PHP (P), Perl (P) [57]. Τα παραπάνω λογισμικά είναι ότι χρειάζεται κάποιος για να δημιουργήσει ένα web server. Το XAMPP είναι επίσης cross-platform, πράγμα που σημαίνει ότι λειτουργεί εξίσου καλά σε Linux, Mac και Windows. Δεδομένου ότι οι περισσότερες πραγματικές υλοποιήσεις διακομιστή web χρησιμοποιούν τα ίδια στοιχεία με το XAMPP, κάνει την μετάβαση από έναν τοπικό διακομιστή δοκιμών σε ένα ζωντανό διακομιστή είναι εξαιρετικά εύκολη. [74] [75]

3.3.7 JavaScript

Η JavaScript είναι μία από τις πιο δημοφιλείς γλώσσες προγραμματισμού στον κόσμο. Σχεδόν κάθε προσωπικός υπολογιστής στον κόσμο έχει τουλάχιστον ένα διερμηνέα JavaScript εγκατεστημένο σε αυτό σε ενεργή χρήση. Η δημοτικότητα του JavaScript οφείλεται εξ ολοκλήρου στο ρόλο του ως γλώσσα γραφής του World Wide Web (WWW).

Η γλώσσα προγραμματισμού JavaScript δημιουργήθηκε αρχικά από τον Brendan Eich της εταιρείας Netscape με την επωνυμία Mocha. Αργότερα, η Mocha μετονομάστηκε σε LiveScript, και τελικά σε JavaScript, κυρίως επειδή η ανάπτυξή της επηρεάστηκε περισσότερο από τη γλώσσα προγραμματισμού Java. Σχεδιάστηκε για να τρέχει στο Netscape Navigator και λόγω της μεγάλης της επιτυχίας η JavaScript έγινε βασικός εξοπλισμός σχεδόν σε όλα τα προγράμματα περιήγησης ιστού. [55]

3.3.8 MySQL

Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων ανοικτού κώδικα [55] Έλαβε το όνομά της από την κόρη του Μόντυ Βιντένιους, τη Μάι (αγγλ. My). Το πρόγραμμα τρέχει έναν εξυπηρετητή (server) παρέχοντας πρόσβαση πολλών χρηστών σε ένα σύνολο βάσεων δεδομένων [77]. Η MySQL είναι δημοφιλής βάση δεδομένων για διαδικτυακά προγράμματα και

ιστοσελίδες. Χρησιμοποιείται σε κάποιες από τις πιο διαδεδομένες διαδικτυακές υπηρεσίες, όπως το Flickr, το YouTube, το Spotify, το Google, το Github, το Pinterest. [76]

3.3.9 Apache Web Server

Ο Apache Web Server είναι ο πιο δημοφιλής και ευρέως διαδεδομένος διακομιστής Web.Το Apache είναι ελαφρύ, ισχυρό και χρησιμοποιείται σε μικρές επιχειρήσεις καθώς και σε μεγάλες εταιρείες. Το Apache είναι επίσης ελεύθερο και ανοιχτό. Ο διακομιστής Web Apache έχει σχεδόν απεριόριστες δυνατότητες, λόγω της μεγάλης διαμόρφωσης του, γεγονός που του επιτρέπει να ενσωματώνεται σε πολλές άλλες εφαρμογές. Ένα από τα πιο δημοφιλή πακέτα είναι η στοίβα εφαρμογών διακομιστή Web LAMP, η οποία περιλαμβάνει τον διακομιστή ApacheWeb μαζί με MySQL, PHP, Perl και Python. Ο διακομιστής Apache Web αναπτύχθηκε από το Apache Software Foundation. [58]

4. ΠΑΡΟΥΣΙΑΣΗ ΕΦΑΡΜΟΓΗΣ

Στο κεφάλαιο αυτό θα γίνει εκτενή αναφορά στα επιμέρους τμήματα που χωρίστηκε η εφαρμογή με στιγμιότυπα των διαφόρων επιλογών κατά την ώρα εκτέλεσης της εφαρμογής για την εξαγωγή μετρικών και άλλων πληροφοριών καθώς και την ανάλυση των έργων που έγιναν λήψη από το Maven (https://mvnrepository.com).

4.1 Επιμέρους τμήματα εφαρμογής

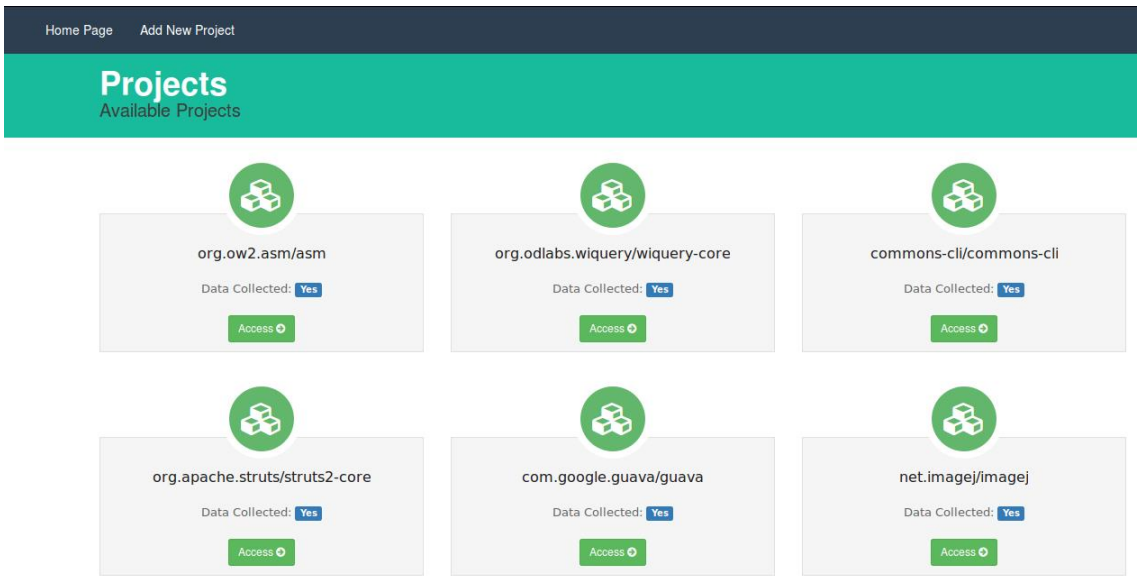
Η εφαρμογή που δημιούργησα αποτελείται από τις εξής επιλογές που θα εξηγήσουμε αναλυτικά πιο κάτω.

- Home Page
- Add New Project
- Access για το κάθε έργο

- Collect Project Data
- Export project Data
- Fetch Project Version
- CKJM Analyze
- Analyze για το κάθε Version
- Result για το κάθε Version
- Charts για το κάθε Version
- Export All Project Data

4.2 Χαρακτηριστικές εικόνες από την εκτέλεση της εφαρμογής

Στην εικόνα 2 βλέπουμε την αρχικής σελίδα της εφαρμογής στην οποία έχουμε προβολή όλων των έργων που θέλουμε να μελετήσουμε, την επιλογή να διαλέξω όποιο έργο επιθυμώ για ανάλυση, εξαγωγή κλπ. καθώς και τις επιλογές Home Page και Add New Project.

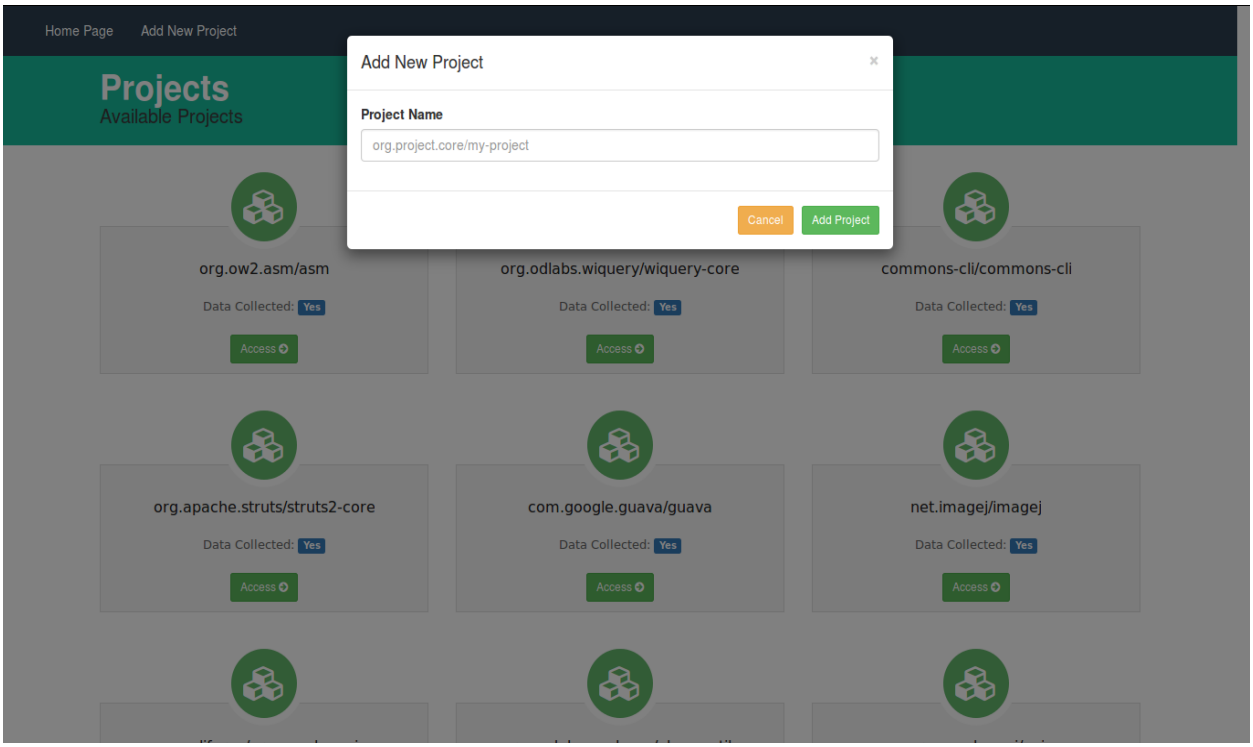


Εικόνα 2: Στιγμιότυπο κεντρικής σελίδας της εφαρμογής

4.2.1 Επιλογές Home Page & Add New Project

Στην εικόνα 3 βλέπουμε τις επιλογές Home Page και Add New Project.

- Home Page: μας επιστρέφει στην αρχική σελίδα όπου και αν βρισκόμαστε μέσα στην εφαρμογή.
- Add New Project: μας επιτρέπει να προσθέσουμε ένα καινούριο έργο από το Maven στην εφαρμογή μας. Για παράδειγμα θέλουμε να προσθέσουμε το έργο ASM που βρίσκετε σε στο σύνδεσμο https://mvnrepository.com/artifact/org.ow2.asm/asm, κάνουμε αντιγραφή το org.ow2.asm/asm και το κάνουμε επικόλλησή στο παράθυρο Add New Project. Ακολούθως επιλέγουμε την επιλογή Add Project και το έργο μας έχει προστεθεί και είναι έτοιμο για ανάλυση.



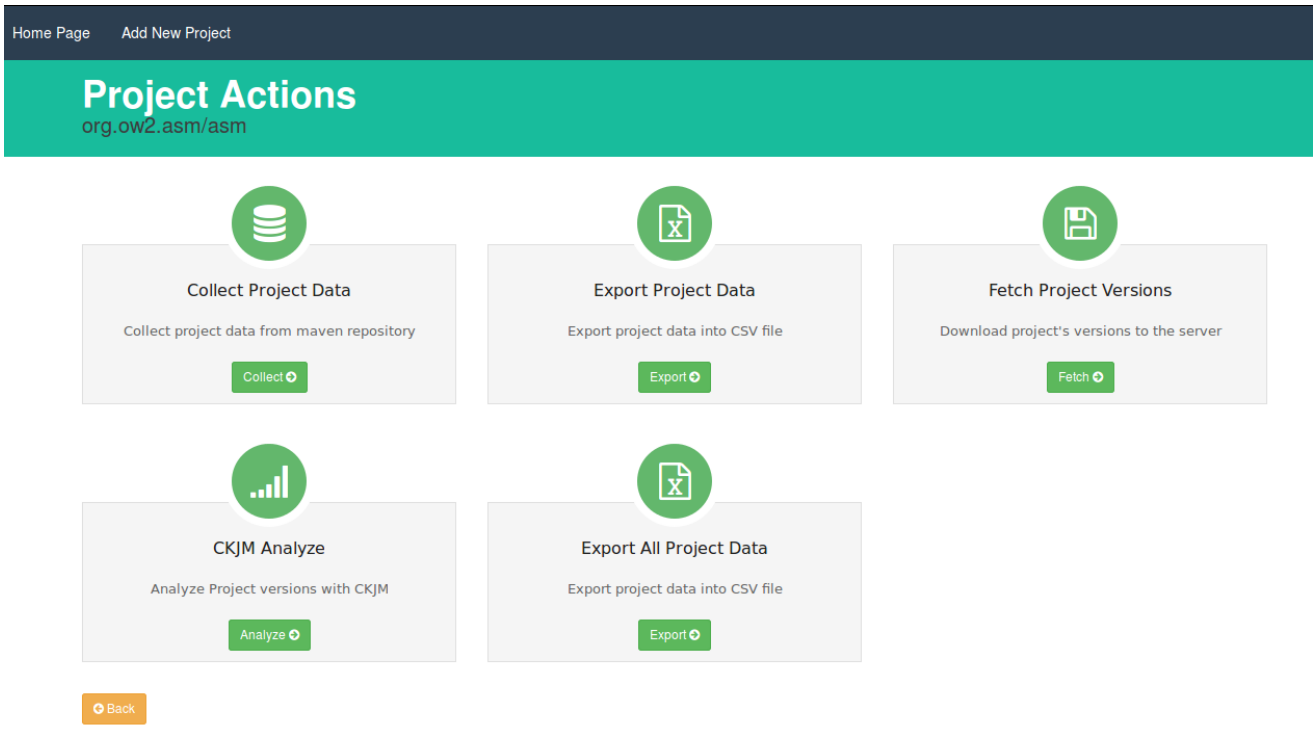
Εικόνα 3: Στιγμιότυπο επιλογής Home Page & Add New Project

4.2.2 Επιλογή Access

Στην εικόνα 4 βλέπουμε την επιλογή Access η οποία μας μεταφέρει στη σελίδα με τις επιλογές που μπορούμε να διαλέξουμε για ένα συγκεκριμένο αρχείο , στην προκειμένη περίπτωση για το αρχείο ASM.

Οι επιλογές είναι οι εξής:

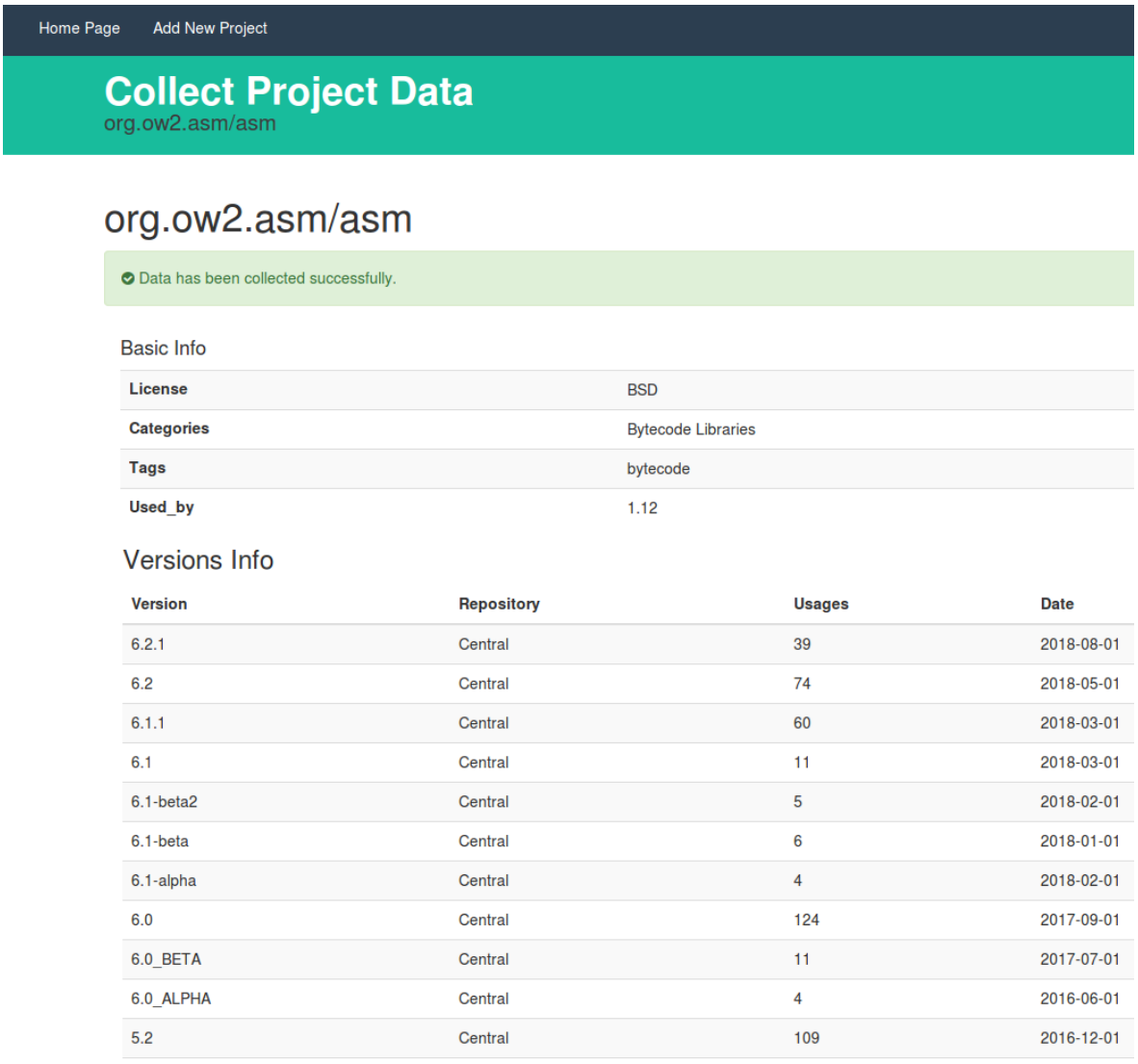
- Collect Project Data
- Export project Data
- Fetch Project Version
- CKJM Analyze
- Export All Project Data



Εικόνα 4: Στιγμιότυπο επιλογής Access

4.2.3 Επιλογή Collect Project Data

Στην εικόνα 5 έχουμε την επιλογή Collect Project Data.η οποία διαλέγοντας την, μας εμφανίζεται ένα μήνυμα ότι τα δεδομένα έχουν συλλεχθεί επιτυχώς και μας εμφανίζει κάποιες βασικές πληροφορίες καθώς και πληροφορίες για τις διάφορες εκδόσεις του συγκεκριμένου έργου, που είχαμε αναλύσει στο κεφάλαιο 3 Ανάλυση και Σχεδίαση Εφαρμογής στην υποκατηγορία 3.2.2 Πίνακες βάσης δεδομένων και Επεξήγηση.



Εικόνα 5: Στιγμιότυπο επιλογής Collect Project Data

4.2.4 Επιλογή Export Project Data

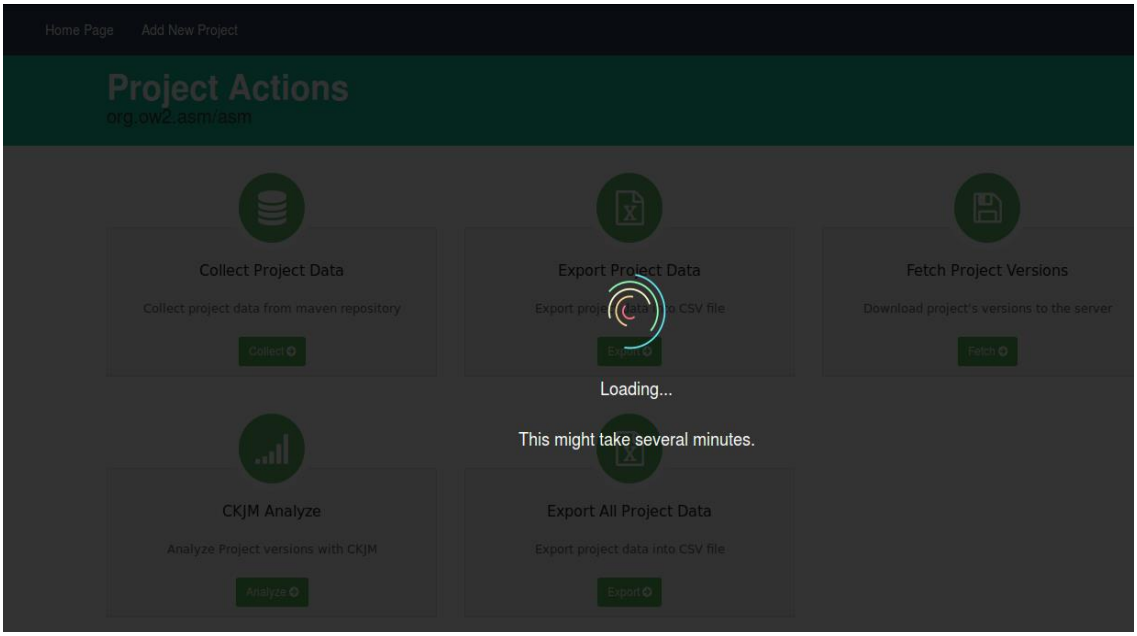
Στην εικόνα 6 έχουμε την επιλογή Export Project Data. Διαλέγοντας την επιλογή αυτή η εφαρμογή μας εξάγει σε ένα αρχείο csv κάποιες βασικές πληροφορίες του έργου, πληροφορίες για τις διάφορες εκδόσεις του που είχαμε αναλύσει στο κεφάλαιο 3 Ανάλυση και Σχεδίαση Εφαρμογής στην υποκατηγορία 3.2.2 Πίνακες βάσης δεδομένων και Επεξήγηση καθώς και από ποια έργα εξαρτούνται οι διάφορες εκδόσεις του συγκεκριμένου έργου.

	A	B	C	D	E	F
1	Έργο	org.ow2.asm/asm				
2						
3	License	BSD				
4	Tags	bytecode				
5	Used_by	1.12				
6						
7						
8	version	repository	analyzed	usages	date	repo_url
9	4	Central	y	73	01-12-11	http://central.maven.org/maven2/
10	4.1	Central	y	112	01-10-12	http://central.maven.org/maven2/
11	4.2	Central	y	38	01-10-13	http://central.maven.org/maven2/
12	5	Central	y	16	01-03-14	http://central.maven.org/maven2/
13	5.0_ALPHA	Central	y	2	01-08-13	http://central.maven.org/maven2/
14	5.0_BETA	Central	y	8	01-10-13	http://central.maven.org/maven2/
15	5.0.1	Central	y	36	01-03-14	http://central.maven.org/maven2/
16	5.0.2	Central	y	38	01-04-14	http://central.maven.org/maven2/
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
34	VERSIONS DEPENDENCIES					
35						
36	Versions					
37	6.1					
38		asm-test				
39		junit-jupiter-api				
40		junit-jupiter-params				
41	6.1-alpha					
42		asm-test				
43		junit-jupiter-api				
44		junit-jupiter-engine				
45		junit-jupiter-params				
46	6.1-beta					
47		asm-test				
48		junit-jupiter-api				
49		junit-jupiter-engine				
50		junit-jupiter-params				

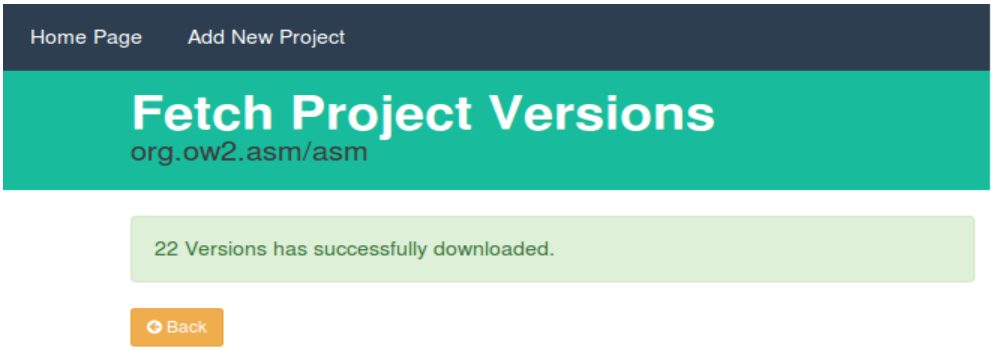
Εικόνα 6: Στιγμιότυπο επιλογής Export Project Data

4.2.5 Επιλογή Fetch Project Versions

Στην εικόνα 7 έχουμε την επιλογή Fetch Project Versions η οποία διαλεγοντας τη μας εμφανίζει ένα μήνυμα ότι η διαδικασία ίσως διαρκέσει κάποια λεπτά και ακολούθως στην εικόνα 8 έχουμε ένα μήνυμα ότι οι είκοσι δύο (22) εκδόσεις του έργου έχουν κατέβει με επιτυχία. Αυτή η επιλογή μας δίνει την δυνατότητα να κατεβάσουμε το συγκεκριμένο έργο μαζί με όλες τις εκδόσεις τους για να μπορέσουμε να τα αναλύσουμε και να πάρουμε τα αποτελέσματα των μετρικών και άλλων πληροφοριών που θέλουμε.



Εικόνα 8: Στιγμιότυπο επιτυχούς λήψης εκδόσεων

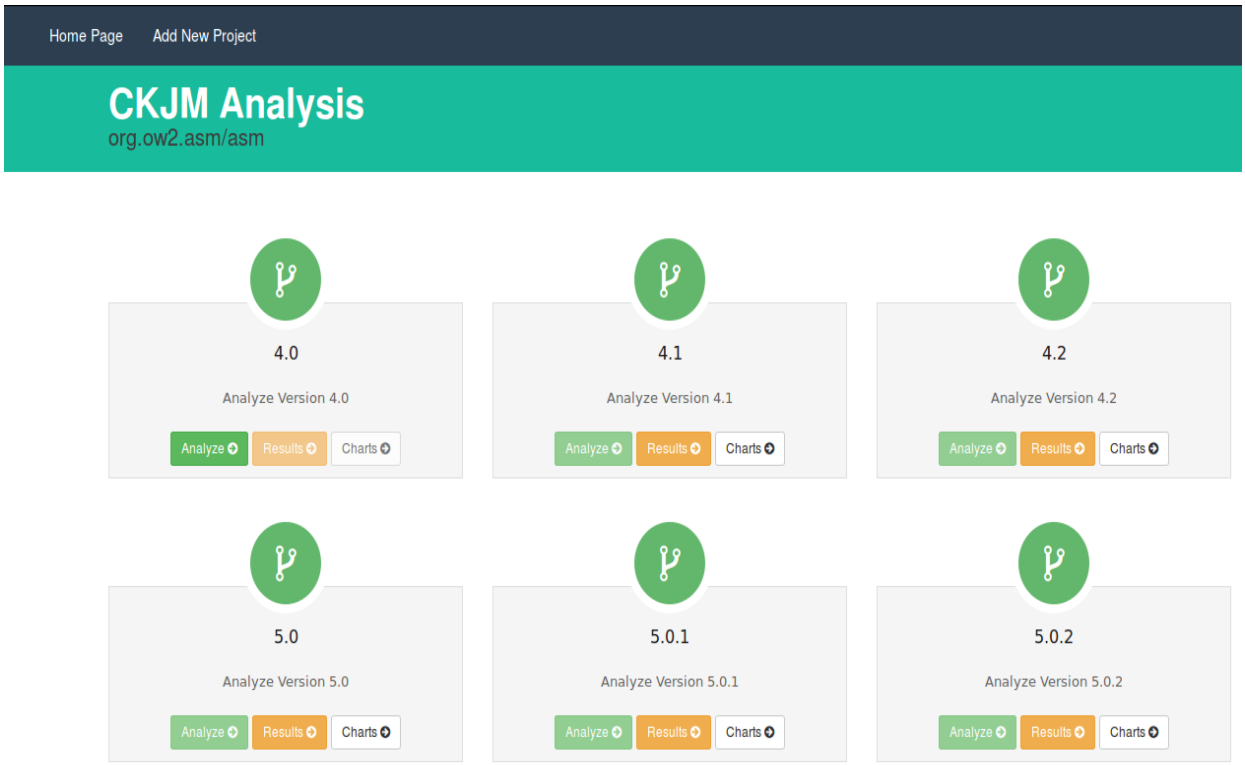


Εικόνα 7: Στιγμιότυπο αναμονής Fetch Project Versions

4.2.6 Επιλογή CKJM Analyze

Στην εικόνα 7 έχουμε την επιλογή CKJM Analysis η οποία μας μεταφέρει σε μια σελίδα όπου φαίνονται όλες οι εκδόσεις του έργου καθώς και τρείς επιλογές για τη κάθε έκδοση.
Οι επιλογές είναι οι εξής:

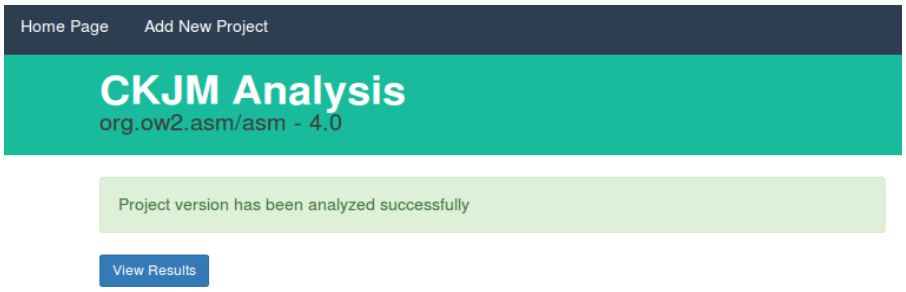
- Analyze
- Results
- Charts



Εικόνα 9: Στιγμιότυπο επιλογής CKJM Analyze

4.2.6.1 Επιλογή Analyze

Στην εικόνα 10 έχουμε την επιλογή Analyze η οποία διαλέγοντας τη μας εμφανίζει ένα μήνυμα ότι η συγκεκριμένης έκδοση του έργου έχει αναλυθεί με επιτυχία και μας παραθέτει να δούμε τα αποτελέσματα (View Result).

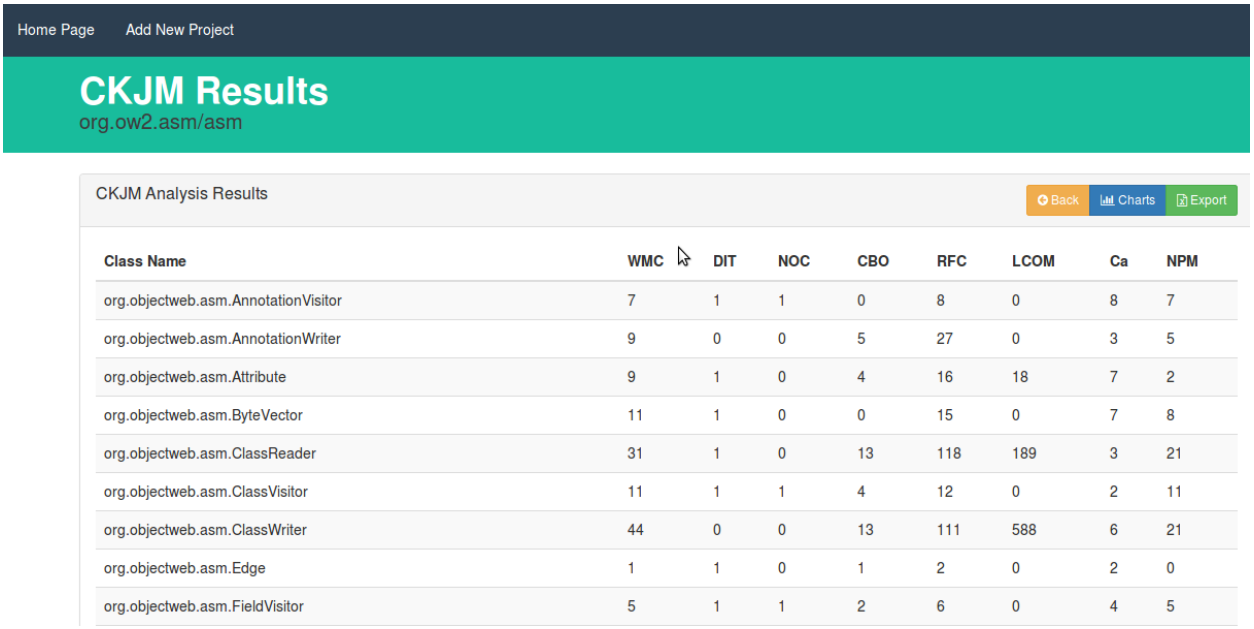


Εικόνα 10: Στιγμιότυπο επιλογής Analyze

4.2.6.2 Επιλογή Result

Στην εικόνα 11 βλέπουμε την επιλογή Result η οποία μας εμφανίζει όλες τις κλάσης της συγκεκριμένης έκδοσης του έργου καθώς και τα αποτελέσματα των μετρικών για την κάθε κλάση.

<u>Σημείωσή</u>: Δεν μπορούμε να δούμε τα αποτελέσματα αν δεν γίνει πρώτα επιτυχής ανάλυση.



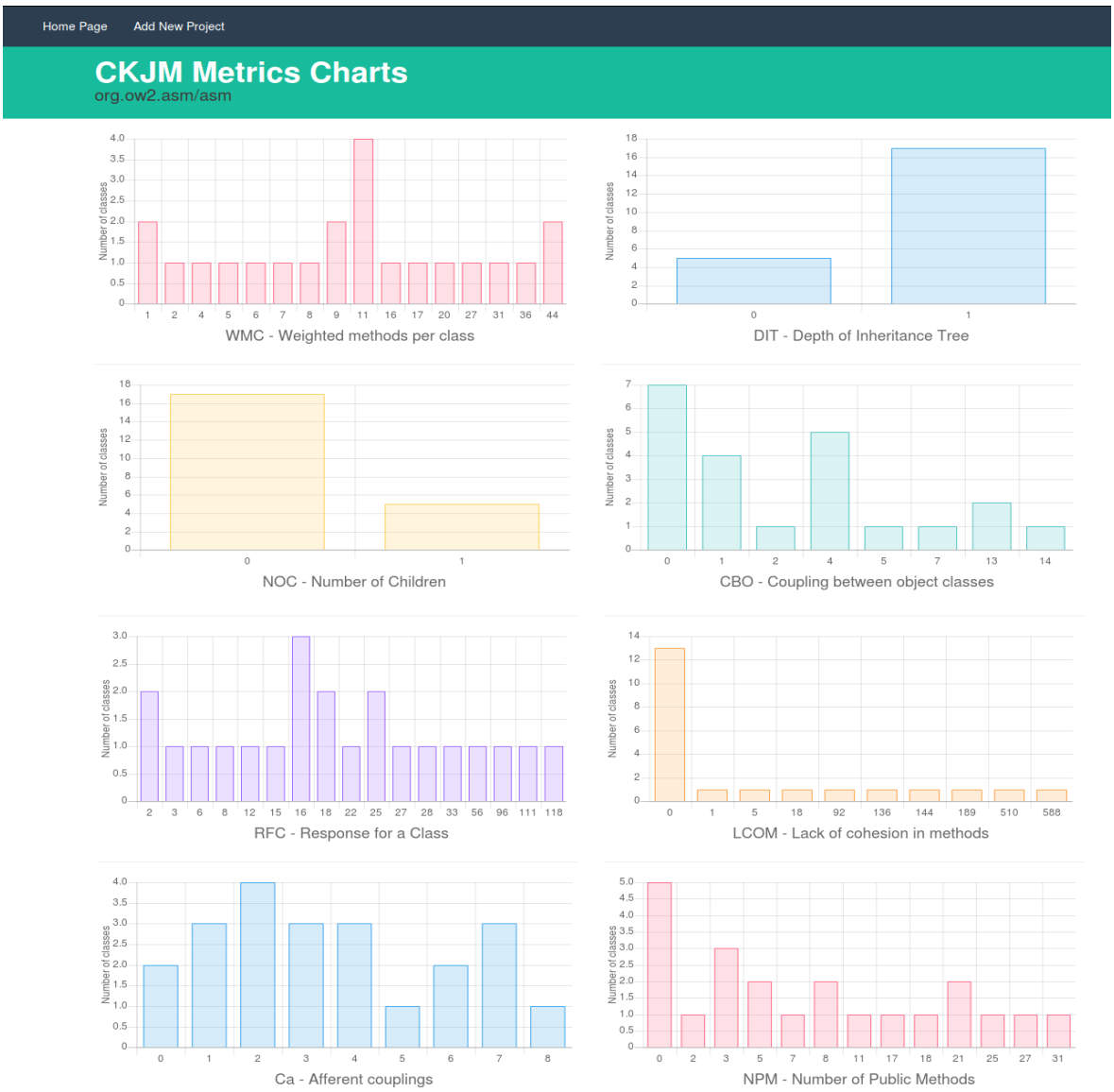
Class Name	WMC	DIT	NOC	CBO	RFC	LCOM	Ca	NPM
org.objectweb.asm.AnnotationVisitor	7	1	1	0	8	0	8	7
org.objectweb.asm.AnnotationWriter	9	0	0	5	27	0	3	5
org.objectweb.asm.Attribute	9	1	0	4	16	18	7	2
org.objectweb.asm.ByteVector	11	1	0	0	15	0	7	8
org.objectweb.asm.ClassReader	31	1	0	13	118	189	3	21
org.objectweb.asm.ClassVisitor	11	1	1	4	12	0	2	11
org.objectweb.asm.ClassWriter	44	0	0	13	111	588	6	21
org.objectweb.asm.Edge	1	1	0	1	2	0	2	0
org.objectweb.asm.FieldVisitor	5	1	1	2	6	0	4	5

Εικόνα 11: Στιγμιότυπο επιλογής Result

4.2.6.3 Επιλογή Charts

Στην εικόνα 12 έχουμε την επιλογή Charts με την οποία η εφαρμογή μας εμφανίζει γραφικές παραστάσεις με όλες τις μετρικές του έργου που θέλουμε να αναλύσουμε. Στον κάθετο άξονα είναι ο αριθμός των κλάσεων και στον οριζόντιο είναι ο βαθμός της κάθε μετρικής που έχουν οι κλάσης, δηλαδή για τη μετρική WMC υπάρχουν δύο κλάσης που έχουν βαθμό WMC εννέα.

Σημείωσή: Δεν μπορούμε να δούμε τις γραφικές παραστάσεις αν δεν γίνει πρώτα επιτυχής ανάλυση.



Εικόνα 12: Στιγμιότυπο επιλογής Charts

4.2.7 Επιλογή Export All Project Data

Στην εικόνα 13α και 13β έχουμε την επιλογή Export All Project Data με την οποία διαλέγοντας τη εξάγει ένα αρχείο csv με όλες τις πληροφορίες του έργου που είχαμε αναλύσει στο κεφάλαιο 3 Ανάλυση και Σχεδίαση Εφαρμογής στην υποκατηγορία 3.2.2 Πίνακες βάσης δεδομένων και Επεξήγηση, τα αποτελέσματα των μετρικών για την κάθε κλάση καθώς και τα έργα από τα οποία εξαρτάτε η εκάστοτε κλάση του έργου.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	PROJECT	LICENSE	TAGS	VERSION	CLASSNAME	Wmc	Dit	Noc	Cbo	Rfc	Lcom	Ca	Npm	REPOSITORY
2														
3	org.ow2.asm/asm	BSD	bytecode											
4				4										Central
5					org.objectweb.asm.AnnotationVisitor	7	1	1	0	8	0	8	7	
6					org.objectweb.asm.AnnotationWriter	9	0	0	5	27	0	3	5	
7					org.objectweb.asm.Attribute	9	1	0	4	16	18	7	2	
8					org.objectweb.asm.ByteVector	11	1	0	0	15	0	7	8	
9					org.objectweb.asm.ClassReader	31	1	0	13	118	189	3	21	
10					org.objectweb.asm.ClassVisitor	11	1	1	4	12	0	2	11	
11					org.objectweb.asm.ClassWriter	44	0	0	13	111	588	6	21	
12					org.objectweb.asm.Edge	1	1	0	1	2	0	2	0	
13					org.objectweb.asm.FieldVisitor	5	1	1	2	6	0	4	5	
14					org.objectweb.asm.FieldWriter	6	0	0	7	18	0	1	3	
15					org.objectweb.asm.Frame	16	1	0	4	33	92	2	0	
16					org.objectweb.asm.Handle	8	1	0	0	16	0	4	8	
17					org.objectweb.asm.Handler	2	1	0	1	3	1	1	0	
18					org.objectweb.asm.Item	11	1	0	0	16	0	6	0	
19					org.objectweb.asm.Label	11	1	0	4	22	5	7	3	
20					org.objectweb.asm.MethodVisitor	27	1	1	4	28	0	4	27	
21					org.objectweb.asm.MethodWriter	44	0	0	14	96	144	3	25	
22					org.objectweb.asm.Opcodes	1	1	0	0	2	0	1	0	
23					org.objectweb.asm.signature.SignatureReader	4	1	0	1	25	0	0	3	
24				.	.	.	.	.	.	.	.	.	.	.
25				.	.	.	.	.	.	.	.	.	.	.
353				.	.	.	.	.	.	.	.	.	.	.
354				6.0_BETA										Central
355														
356				6.1										Central
357														
358				6.1-alpha										Central
359														
360				6.1-beta										Central
361														
362				6.1-beta2										Central
363														
364				6.1.1										Central

Εικόνα 13α: Στιγμιότυπο επιλογής Export All Project Data

O	P	Q	R	S	T
ANALYZED	NUMBER OF USES	DATE	URL	NO. OF DEPENDENCIES	DEPENDENCIES
y	70	01-12-11	http://central.maven.org/maven2/		
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
n	11	01-07-17	http://central.maven.org/maven2/		
n	9	01-03-18	http://central.maven.org/maven2/	3	asm-test,junit-jupiter-api,junit-jupiter-params
n	4	01-02-18	http://central.maven.org/maven2/	4	asm-test,junit-jupiter-api,junit-jupiter-engine,junit-jupiter-params
n	6	01-01-18	http://central.maven.org/maven2/	4	asm-test,junit-jupiter-api,junit-jupiter-engine,junit-jupiter-params
n	5	01-02-18	http://central.maven.org/maven2/	4	asm-test,junit-jupiter-api,junit-jupiter-engine,junit-jupiter-params
n	22	01-03-18	http://central.maven.org/maven2/	3	asm-test,junit-jupiter-api,junit-jupiter-params

Εικόνα 13β: Στιγμιότυπο επιλογής Export All Project Data

4.3 Προγράμματα σε Java που αναλύθηκαν

Σε αυτή την υποενότητα θα δείξουμε μέσα απ τις εικόνες 14 – 43 τα τριάντα έργα που αναλύσαμε μαζί με κάποια χαρακτηριστικά τους, τα οποία είναι από το Πίνακα 1: Πεδία basic_Info και Πίνακα 5: Πεδία version του Παραρτήματος 3.2.2. Επιπλέον είναι και τα πεδία Project που είναι το όνομα του έργου και Num of Versions που είναι ο αριθμός των εκδόσεων που έχει το κάθε έργο.

Project	asm			
License	BSD			
Tags	bytecode			
Used_by	1.237			
Num of Versions	24			
version	repository	analyzed	usages	date
4	Central	y	78	01-12-11
4.1	Central	y	112	01-10-12
4.2	Central	y	38	01-10-13
5	Central	y	17	01-03-14
5.0_ALPHA	Central	y	2	01-08-13
5.0_BETA	Central	y	8	01-10-13
5.0.1	Central	y	37	01-03-14
5.0.2	Central	y	40	01-04-14
5.0.3	Central	y	324	01-05-14
5.0.4	Central	y	422	01-05-15
5.1	Central	y	152	01-03-16
5.2	Central	y	120	01-12-16
6	Central	n	138	01-09-17
6.0_ALPHA	Central	y	5	01-06-16
6.0_BETA	Central	n	11	01-07-17
6.1	Central	n	11	01-03-18

Εικόνα 14: asm

Project	wiquery-core			
License	-			
Tags				
Used_by	15			
Num of Versions	12			
version	repository	analyzed	usages	date
1.5.6	Central	y	1	01-05-12
1.5.7	Central	y	3	01-06-12
1.5.8	Central	y	1	01-11-12
6.0.0	Central	y	1	01-08-12
6.0.0-beta3	Central	y	1	01-07-12
6.12.0	Central	y	1	01-12-13
6.13.0	Central	y	5	01-01-14
6.2.0	Central	y	2	01-11-12
6.5.0	Central	y	2	01-02-13
6.6.0	Central	y	4	01-02-13
6.9.1	Central	y	3	01-07-13
6.9.2	Central	y	2	01-07-13

Εικόνα 15: wiquery-core

Project	commons-cli			
License	Apache 2.0			
Tags	command-line,cli,parser			
Used_by	2.541			
Num of Versions	9			
version	repository	analyzed	usages	date
1	Central	y	198	01-11-05
1.0-beta-1	Central	y	0	01-11-05
1.0-beta-2	Central	y	3	01-11-05
1.1	Central	y	199	01-07-07
1.2	Central	y	1.301	01-03-09
1.3	Central	y	121	01-05-15
1.3.1	Central	y	447	01-06-15
1.4	Central	y	491	01-03-17
20040117	Central	y	4	01-11-05

Εικόνα 16: commons-cli

Project	struts2-core			
License	Apache 2.0			
Tags	web-framework,apache,web,framework			
Used_by	177			
Num of Versions	25			
version	repository	analyzed	usages	date
2.0.11	Central	y	6	01-10-07
2.0.11.1	Central	y	6	01-03-08
2.0.11.2	Central	y	4	01-06-08
2.0.12	Central	y	4	01-10-08
2.0.14	Central	y	7	01-11-08
2.0.5	Central	y	6	01-02-07
2.0.6	Central	y	8	01-02-07
2.0.8	Central	y	3	01-06-07
2.0.9	Central	y	10	01-07-07
2.1.2	Central	y	10	01-05-08
2.1.6	Central	y	10	01-01-09
2.1.8	Central	y	12	01-09-09
2.1.8.1	Central	y	29	01-11-09
2.2.1	Central	y	18	01-07-10
2.2.1.1	Central	y	16	01-12-10

Εικόνα 17: struts2-core

Project	guava			
License	Apache 2.0			
Tags	google			
Used_by	15.325			
Num of Versions	80			
version	repository	analyzed	usages	date
10	Central	y	112	01-09-11
10.0-rc1	Central	y	3	01-09-11
10.0-rc2	Central	y	2	01-09-11
10.0-rc3	Central	y	2	01-09-11
10.0.1	Central	y	733	01-10-11
11	Central	y	109	01-12-11
11.0-rc1	Central	y	4	01-12-11
11.0.1	Central	y	290	01-01-12
11.0.2	Central	y	903	01-02-12
12	Central	y	400	01-04-12
12.0-rc1	Central	y	12	01-03-12
12.0-rc2	Central	y	5	01-04-12
12.0.1	Central	y	164	01-07-12
13	Central	y	352	01-08-12

Εικόνα 18: guava

Project	imagej			
License	BSD			
Tags				
Used_by	37			
Num of Versions	66			
version	repository	analyzed	usages	date
2.0.0-beta-8	ImageJ Releases	y	0	01-05-14
2.0.0-rc-1	ImageJ Releases	y	0	01-05-14
2.0.0-rc-10	ImageJ Releases	y	0	01-08-14
2.0.0-rc-11	ImageJ Releases	y	0	01-08-14
2.0.0-rc-12	ImageJ Releases	y	0	01-08-14
2.0.0-rc-13	ImageJ Releases	y	1	01-08-14
2.0.0-rc-14	ImageJ Releases	y	1	01-08-14
2.0.0-rc-15	ImageJ Releases	y	0	01-10-14
2.0.0-rc-16	ImageJ Releases	y	0	01-10-14
2.0.0-rc-17	ImageJ Releases	y	1	01-11-14
2.0.0-rc-18	ImageJ Releases	y	0	01-12-14

Εικόνα 19: imagej

Project	plexus-utils			
License	Apache 2.0			
Tags	codehaus			
Used_by	2.73			
Num of Versions	68			
version	repository	analyzed	usages	date
1.0.4	Central	y	155	01-11-05
1.0.5	Central	y	53	01-12-05
1.1	Central	y	167	01-01-06
1.2	Central	y	57	01-04-06
1.3	Central	y	10	01-09-06
1.4	Central	y	38	01-01-07
1.4-alpha-1	Central	y	3	01-12-06
1.4.1	Central	y	60	01-03-07
1.4.2	Central	y	15	01-05-07
1.4.3	Central	y	10	01-07-07
1.4.4	Central	y	6	01-07-07

Εικόνα 20: plexus-util

Project	poi			
License	Apache 2.0			
Tags	apache,excel,spreadsheet			
Used_by	802			
Num of Versions	49			
version	repository	analyzed	usages	date
3.0-FINAL	Central	n	24	01-05-07
3.0.1-FINAL	Central	y	16	01-07-07
3.0.2-beta1	Central	y	2	01-12-07
3.0.2-beta2	Central	y	2	01-01-08
3.0.2-FINAL	Central	y	27	01-02-08
3.1-beta1	Central	y	2	01-04-08
3.1-beta2	Central	y	3	01-05-08
3.1-FINAL	Central	y	17	01-06-08
3.10-beta1	Central	y	8	01-07-13
3.10-beta2	Central	y	17	01-09-13
3.10-FINAL	Central	y	73	01-02-14

Εικόνα 21: poi

Project	guice-bean-binders			
License	Apache 2.0,EPL 1.0			
Tags	beans,dependency-injection			
Used_by	3			
Num of Versions	15			
version	repository	analyzed	usages	date
1.4.0	Central	y	3	01-09-10
1.4.1	Central	y	3	01-09-10
1.4.2	Central	y	3	01-09-10
1.4.3	Central	y	3	01-11-10
1.4.3-server	Central	y	3	01-01-11
1.4.3.1	Central	y	3	01-11-10
1.4.3.2	Central	y	3	01-12-10
2.0.0	Central	y	3	01-01-11
2.1.1	Central	y	3	01-02-11
2.1.1-NEXUS-4312	Central	y	3	01-05-11
2.2.0	Central	y	3	01-03-11
2.2.1	Central	y	3	01-04-11

Εικόνα 22: guice-bean-binders

Project	wro4j-core			
License	Apache 2.0			
Tags				
Used_by	26			
Num of Versio	46			
version	repository	analyzed	usages	date
1.2.0	Central	y	3	01-03-10
1.2.1	Central	y	3	01-03-10
1.2.2	Central	y	3	01-04-10
1.2.3	Central	y	3	01-04-10
1.2.4	Central	y	3	01-04-10
1.2.5	Central	y	3	01-04-10
1.2.6	Central	y	2	01-04-10
1.2.7	Central	y	2	01-05-10
1.2.8	Central	y	3	01-06-10
1.3.0	Central	y	2	01-11-10
1.3.0-rc1	Central	y	2	01-09-10

Εικόνα 23: wro4j-core

Project	xstream			
License	BSD			
Tags	xml			
Used_by	1604			
Num of Versions	29			
version	repository	analyzed	usages	date
0.1	Central	y	0	01-09-06
0.2	Central	y	0	01-09-06
0.3	Central	y	0	01-09-06
0.5	Central	y	0	01-09-06
0.6	Central	y	0	01-09-06
1	Central	y	0	01-11-06
1.0.1	Central	y	0	01-11-06
1.0.2	Central	y	0	01-11-06
1.1	Central	n	0	01-11-06
1.1.1	Central	n	13	01-11-06
1.1.2	Central	n	6	01-11-06
1.1.3	Central	n	9	01-11-06
1.2	Central	n	22	01-08-06
1.2.1	Central	n	29	01-11-06
1.2.2	Central	n	98	01-05-07
1.3	Central	n	100	01-02-08
1.3.1	Central	n	364	01-12-08

Εικόνα 24: xtream

Project	jdom			
License	-			
Tags	xml			
Used_by	590			
Num of Version:	6			
version	repository	analyzed	usages	date
1.1	Central	y	344	01-09-08
1.1.2	Central	y	29	01-10-11
1.1.3	Central	y	154	01-02-12
2.0.0	Central	y	3	01-04-12
2.0.1	Central	y	7	01-04-12
2.0.2	Central	y	96	01-06-12

Εικόνα 25: jdom

Project	commons-lang3			
License	Apache 2.0			
Tags	apache,commons			
Used_by	12,457			
Num of Versions	11			
version	repository	analyzed	usages	date
3	Central	y	277	01-07-11
3.0.1	Central	n	183	01-08-11
3.1	Central	n	1.825	01-11-11
3.2	Central	n	94	01-12-13
3.2.1	Central	n	260	01-01-14
3.3	Central	n	136	01-02-14
3.3.1	Central	n	180	01-03-14
3.3.2	Central	n	2.083	01-04-14
3.4	Central	n	3.3	01-04-15
3.5	Central	n	1.731	01-10-16
3.6	Central	n	1.344	01-06-17
3.7	Central	n	1.261	01-11-17

Εικόνα 26: commons-lang3

Project	spring-core			
License	Apache 2.0			
Tags	spring			
Used_by	5,281			
Num of Versions				
version	repository	analyzed	usages	date
1	Central	y	0	01-12-05
1.0-rc1	Central	y	0	01-12-05
1.0.1	Central	y	0	01-12-05
1.1	Central	y	0	01-12-05
1.1-rc1	Central	y	0	01-12-05
1.1-rc2	Central	y	0	01-12-05
1.1.1	Central	y	0	01-12-05
1.1.2	Central	y	0	01-12-05
1.1.3	Central	y	0	01-12-05
1.1.4	Central	y	0	01-12-05
1.1.5	Central	y	0	01-12-05
1.2	Central	y	2	01-12-05
1.2-rc1	Central	y	0	01-12-05

Εικόνα 27: spring-core

Project	joda-time			
License	Apache 2.0			
Tags	time,date			
Used_by	6,577			
Num of Versio	35			
version	repository	analyzed	usages	date
0.95	Central	y	0	01-11-05
1	Central	y	2	01-11-05
1.1	Central	y	1	01-11-05
1.2	Central	y	4	01-01-06
1.2.1	Central	y	9	01-02-06
1.3	Central	y	4	01-08-06
1.4	Central	y	60	01-11-06
1.5	Central	y	7	01-10-07
1.5.1	Central	y	35	01-12-07
1.5.2	Central	y	46	01-01-08
1.6	Central	y	453	01-12-08
1.6.1	Central	y	11	01-08-10
1.6.2	Central	y	289	01-09-10

Εικόνα 28: joda-time

Project	jopt-simple			
License	MIT			
Tags	cli,parser			
Used_by	283			
Num of Versio	44			
version	repositor	analyzed	usages	date
2.3.6	Central	y	0	01-03-08
2.4	Central	y	0	01-06-08
2.4-beta1	Central	y	0	01-04-08
2.4-beta2	Central	y	0	01-05-08
2.4-rc1	Central	y	0	01-05-08
2.4-rc2	Central	y	0	01-05-08
2.4.1	Central	y	2	01-07-08
3	Central	y	6	01-01-09
3.0-beta1	Central	y	0	01-12-08
3.0-rc1	Central	y	0	01-12-08
3.0-rc2	Central	y	1	01-01-09
3.0.1	Central	y	0	01-03-09

Εικόνα 29: jopt-simple

Project	lucene-core			
License	Apache 2.0			
Tags	lucene,indexing,apache,full-text,index			
Used_by	950			
Num of Versions	89			
version	repository	analyzed	usages	date
1.9.1	Central	y	6	01-06-06
2.0.0	Central	y	29	01-06-06
2.1.0	Central	y	4	01-06-07
2.2.0	Central	y	39	01-06-07
2.3.0	Central	y	8	01-01-08
2.3.1	Central	y	14	01-02-08
2.3.2	Central	y	56	01-05-08
2.4.0	Central	y	13	01-10-08
2.4.1	Central	y	43	01-03-09
2.9.0	Central	y	6	01-09-09
2.9.1	Central	y	13	01-11-09
2.9.2	Central	y	20	01-02-10
2.9.3	Central	y	16	01-06-10

Εικόνα 30: lucene-core

Project	log4j-core			
License	Apache 2.0			
Tags	logging,apache			
Used_by	3,922			
Num of Versions	33			
version	repository	analyzed	usages	date
2	Central	y	36	01-07-14
2.0-alpha1	Central	y	4	01-07-12
2.0-alpha2	Central	y	4	01-08-12
2.0-beta1	Central	y	7	01-09-12
2.0-beta2	Central	y	5	01-10-12
2.0-beta3	Central	y	22	01-11-12
2.0-beta4	Central	y	13	01-01-13
2.0-beta5	Central	n	17	01-04-13
2.0-beta6	Central	y	19	01-05-13
2.0-beta7	Central	y	21	01-06-13
2.0-beta8	Central	y	24	01-07-13
2.0-beta9	Central	y	60	01-09-13

Εικόνα 31: log4j-core

Project	commons-io			
License	Apache 2.0			
Tags	io			
Used_by	15,850			
Num of Version	17			
version	repository	analyzed	usages	date
0.1	Central	y	1	01-11-05
1	Central	y	32	01-11-05
1.1	Central	n	72	01-11-05
1.2	Central	n	223	01-03-06
1.3	Central	n	37	01-01-07
1.3.1	Central	n	168	01-02-07
1.3.2	Central	n	296	01-01-08
1.4	Central	n	1.904	01-01-08
2	Central	n	197	01-10-10
2.0.1	Central	n	861	01-12-10
2.1	Central	n	1.078	01-10-11
2.2	Central	n	381	01-03-12
2.3	Central	n	266	01-04-12
2.4	Central	n	6.552	01-06-12

Εικόνα 32: commons-io

Project	slf4j-api			
License	MIT			
Tags	api,logging,slf4j			
Used_by	34,514			
Num of Versions	64			
version	repository	analyzed	usages	date
1.1.0	Central	y	6	01-12-06
1.1.0-beta0	Central	y	6	01-09-06
1.1.0-RC0	Central	y	6	01-11-06
1.1.0-RC1	Central	y	9	01-11-06
1.2	Central	y	11	01-01-07
1.3.0	Central	y	63	01-02-07
1.3.1	Central	y	18	01-04-07
1.4.0	Central	y	25	01-05-07
1.4.1	Central	y	13	01-07-07
1.4.2	Central	y	41	01-07-07

Εικόνα 33: slf4j-core

Project	itextpd			
License	AGPL 3.0			
Tags	pdf			
Used_by	169			
Num of Versions	31			
version	repository	analyzed	usages	date
5.0.6	Central	y	10	01-04-11
5.1.0	Central	y	7	01-05-11
5.1.1	Central	y	11	01-06-11
5.1.2	Central	y	9	01-08-11
5.1.3	Central	y	17	01-11-11
5.2.0	Central	y	9	01-03-12
5.2.1	Central	y	3	01-04-12
5.3.0	Central	y	6	01-06-12
5.3.1	Central	y	4	01-07-12
5.3.2	Central	y	11	01-08-12
5.3.4	Central	y	10	01-11-12

Εικόνα 34: itextpd

Project	superfly-remote-api			
License	Apache 2.0			
Tags	remote,api			
Used_by	9			
Num of Versions	46			
version	repository	analyzed	usages	date
1.1-1	Central	y	5	01-08-10
1.1-2	Central	y	5	01-10-10
1.2-1	Central	y	5	01-10-10
1.2-10	Central	y	7	01-05-11
1.2-11	Central	y	7	01-07-11
1.2-12	Central	y	7	01-07-11
1.2-15	Central	y	7	01-09-11
1.2-2	Central	y	5	01-11-10
1.2-3	Central	y	5	01-11-10
1.2-4	Central	y	6	01-11-10
1.2-5	Central	y	6	01-11-10
1.2-6	Central	y	2	01-12-10
1.2-7	Central	y	6	01-12-10

Εικόνα 35: superfly-remote-api

Project	geotk-utility			
License	LGPL 2.1			
Tags				
Used_by	62			
Num of Versions	3			
version	repository	analyzed	usages	date
3.18-geoapi-3.0	Central	y	5	01-06-11
3.19-geoapi-3.0	Central	y	5	01-09-11
3.20-geoapi-3.0	Central	y	5	01-11-12

Εικόνα 36: geotk-utility

Project	javax.servlet-api			
License	CDDL,GPL 2.0			
Tags	standard,servlet,javax,api,specs			
Used_by	10,208			
Num of Versions	20			
version	repository	analyzed	usages	date
3.0.1	Central	y	3.889	01-07-11
3.1-b01	Central	y	12	01-07-12
3.1-b02	Central	y	16	01-09-12
3.1-b03	Central	y	0	01-12-12
3.1-b04	Central	y	2	01-12-12
3.1-b05	Central	y	27	01-01-13
3.1-b06	Central	y	3	01-02-13
3.1-b07	Central	y	4	01-03-13
3.1-b08	Central	y	3	01-04-13
3.1-b09	Central	y	1	01-04-13
3.1.0	Central	y	4.445	01-04-13
4.0.0	Central	y	193	01-08-17
4.0.0-b01	Central	y	18	01-10-15
4.0.0-b02	Central	y	3	01-02-17
4.0.0-b03	Central	y	2	01-03-17
4.0.0-b04	Central	y	0	01-03-17
4.0.0-b05	Central	y	4	01-03-17
4.0.0-b06	Central	y	0	01-05-17
4.0.0-b07	Central	y	21	01-06-17
4.0.1	Central	y	40	01-04-18

Εικόνα 37: javax.servlet-api

Project	commons-logging			
License	Apache 2.0			
Tags	logging			
Used_by	8,485			
Num of Versions	9			
version	repository	analyzed	usages	date
1	Central	y	29	01-11-05
1.0.1	Central	y	4	01-11-05
1.0.2	Central	y	35	01-11-05
1.0.3	Central	y	193	01-11-05
1.0.4	Central	y	1.025	01-11-05
1.1	Central	y	1.122	01-05-06
1.1.1	Central	y	3.17	01-11-07
1.1.2	Central	y	75	01-03-13
1.1.3	Central	y	768	01-05-13
1.2	Central	y	1.868	01-07-14

Εικόνα 38: commons-logging

Project	httpclient			
License	Apache 2.0			
Tags	http,apache,client			
Used_by	6,578			
Num of Versions	46			
version	repository	analyzed	usages	date
4	Central	y	101	01-08-09
4.0-alpha1	Central	y	0	01-07-07
4.0-alpha2	Central	y	2	01-11-07
4.0-alpha3	Central	y	4	01-02-08
4.0-alpha4	Central	y	6	01-05-08
4.0-beta1	Central	y	12	01-08-08
4.0-beta2	Central	y	26	01-12-08
4.0.1	Central	y	259	01-12-09
4.0.2	Central	y	24	01-09-10
4.0.3	Central	y	80	01-09-10
4.1	Central	y	127	01-01-11
4.1-alpha1	Central	y	16	01-12-09
4.1-alpha2	Central	y	6	01-05-10
4.1-beta1	Central	y	11	01-11-10

Εικόνα 39: httpclient

Project	velocity			
License	Apache 2.0			
Tags	apache,template,engine			
Used_by	1,104			
Num of Versions	10			
version	repository	analyzed	usages	date
1.5	Central	y	148	01-03-07
1.6	Central	y	9	01-12-08
1.6-beta1	Central	y	4	01-09-08
1.6-beta2	Central	y	0	01-10-08
1.6.1	Central	y	23	01-12-08
1.6.2	Central	y	83	01-03-09
1.6.3	Central	y	44	01-12-09
1.6.4	Central	y	114	01-05-10
1.7	Central	y	800	01-11-10
1.7-beta1	Central	y	1	01-04-10

Εικόνα 40: velocity

Project	commons-math3			
License	Apache 2.0			
Tags	math,apache,commons			
Used_by	1,021			
Num of Versions	10			
version	repository	analyzed	usages	date
3	Central	y	70	01-03-12
3.1	Central	y	7	01-12-12
3.1.1	Central	y	96	01-01-13
3.2	Central	y	192	01-04-13
3.3	Central	y	90	01-05-14
3.4	Central	y	5	01-12-14
3.4.1	Central	y	123	01-01-15
3.5	Central	y	162	01-04-15
3.6	Central	y	65	01-01-16
3.6.1	Central	y	431	01-03-16

Εικόνα 41: commons-math3

Project	avro			
License	Apache 2.0			
Tags	apache,protocol,serialization,avro			
Used_by	780			
Num of Versions	22			
version	repository	analyzed	usages	date
1.4.0	Central	y	32	01-09-10
1.4.1	Central	y	1	01-10-10
1.5.0	Central	y	4	01-03-11
1.5.1	Central	y	10	01-04-11
1.5.2	Central	y	2	01-07-11
1.5.3	Central	y	22	01-08-11
1.5.4	Central	y	11	01-09-11
1.6.0	Central	y	6	01-10-11
1.6.1	Central	y	13	01-11-11
1.6.2	Central	y	9	01-02-12
1.6.3	Central	y	18	01-03-12
1.7.0	Central	y	8	01-06-12
1.7.1	Central	y	14	01-07-12

Εικόνα 42: avro

Project	httpcore			
License	Apache 2.0			
Tags	http,apache,client			
Used_by	2.302			
Num of Versions	45			
version	repository	analyzed	usages	date
4	Central	y	27	01-02-09
4.0-alpha3	Central	n	0	01-12-06
4.0-alpha4	Central	n	0	01-03-07
4.0-alpha5	Central	y	18	01-07-07
4.0-alpha6	Central	y	15	01-10-07
4.0-beta1	Central	y	14	01-01-08
4.0-beta2	Central	y	6	01-06-08
4.0-beta3	Central	y	10	01-10-08
4.0.1	Central	y	120	01-06-09
4.1	Central	y	75	01-11-10
4.1-alpha1	Central	y	7	01-09-09
4.1-beta1	Central	y	4	01-04-10

Εικόνα 43: httpcore

5. ΕΞΑΓΩΓΗ ΣΤΑΤΙΣΤΙΚΩΝ ΣΤΟΙΧΕΙΩΝ

5.1 Βοηθητικές πλατφόρμες

Σε αυτή την ενότητα θα γίνει αναφορά στις πλατφόρμες που συνέβαλαν στην εξαγωγή των στοιχείων από τη βάση δεδομένων και στην συνέχεια στην εξαγωγή των στατιστικών στοιχείων από αυτά. Τέλος, υπάρχουν στιγμιότυπα πινάκων με τη στατιστική ανάλυση για κάθε ένα από τους νόμους του Lehman με τα τριάντα έργα που αναλύσαμε.

5.1.1 Microsoft Excel

Το Microsoft Excel είναι πρόγραμμα λογιστικών φύλλων που αναπτύχθηκε από τη Microsoft για τα Microsoft Windows, macOS, Android και iOS. Διαθέτει υπολογισμούς, εργαλεία γραφημάτων, συγκεντρωτικούς πίνακες και μια γλώσσα προγραμματισμού macro με όνομα Visual Basic for Applications. Είναι μια πολύ ευρέως διαδεδομένη εφαρμογή υπολογιστικών φύλλων, ειδικά από την έκδοση 5 το 1993, και έχει αντικαταστήσει το Lotus 1-2-3 ως το βιομηχανικό πρότυπο για υπολογιστικά φύλλα. Το Microsoft Excel αποτελεί μέρος του Microsoft Office.[78]

5.2.2 XLSTAT 365

Το XLSTAT 365 είναι μια φιλική προς το χρήστη εφαρμογή για στατιστική ανάλυση και ανάλυση δεδομένων, η οποία λειτουργεί άψογα ως πρόσθετο στο Microsoft Excel 365. Το XLSTAT 365 κάνει την ανάλυση δεδομένων πιο εύκολη από ποτέ, καθώς μας παρέχει πρόσβαση σε όλες τις στατιστικές λειτουργίες mainstream οποτεδήποτε, οπουδήποτε από τον υπολογιστή, το MAC, το Tablet ή το Smartphone. Το μόνο που χρειάζεται για να έχουμε πρόσβαση στο λογισμικό, πρέπει να δημιουργήσουμε έναν λογαριασμό XLSTAT 365 το οποίο μας παρέχει 15 μέρες δωρεάν λειτουργείας ή αν επιθυμούμε την αγορά κάποιου από τα πακέτα του έχει. Τα εξαγόμενα αποτελέσματα μπορούν να προσαρμοστούν και να μοιραστούν με άλλους.

Το XLSTAT 365 παρουσιάζεται από την Addinsoft, τον κατασκευαστή του XLSTAT και τον ηγέτη στο στατιστικό λογισμικό για το Microsoft Excel.[79]

5.2 Στατιστικά Αποτελέσματα

Σε αυτό το κεφάλαιο, θα παρουσιάσουμε τα αποτελέσματα 30 έργων για κάθε ένα από τους νόμους του Lehman για να δούμε κατά πόσο γίνεται επικύρωση τους ή όχι. Έχουμε αποκτήσει μετρήσεις για συνολικά 30 έργα σε Java. Για κάποιους νόμους υπάρχουν στατιστικά σημαντικά συμπεράσματα που λείπουν, έτσι δεν υπάρχουν αρκετά στοιχεία για να δηλωθεί ο νόμος ως επικυρωμένος ή όχι. Κάθε δοκιμή έχει δύο υποθέσεις: [28]

- H0: Η μετρική x δεν παρουσιάζει τάση
- H1: Η μετρική x παρουσιάζει τάση

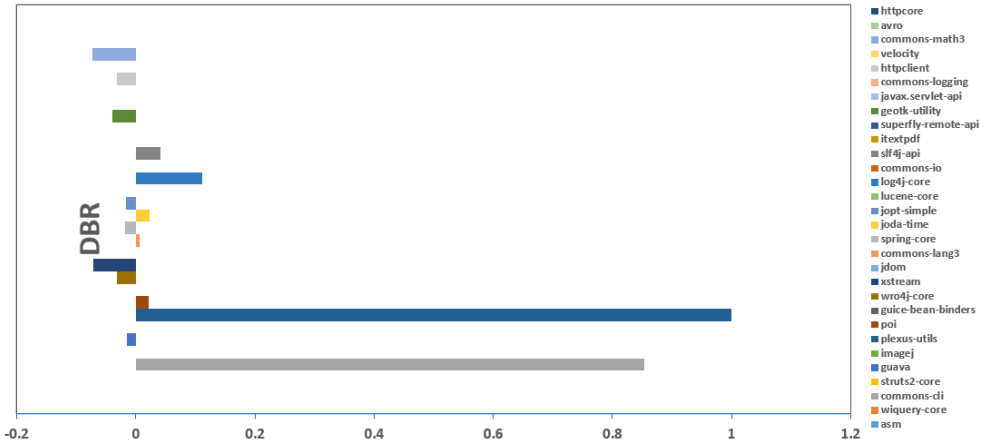
Όπου το "x" είναι η ακριβής μετρική που εξετάσαμε για κάθε νόμο. Το "p-value" είναι η τιμή που μας διασφαλίζει αν το αποτέλεσμα μας θεωρείται ασφαλές για να γίνει αποδεκτό ή όχι. Δεδομένου ότι η υπολογιζόμενη τιμή p-value είναι χαμηλότερη από 0,05 (επίπεδο σημαντικότητας alpha = 0,05) πρέπει να απορριφθεί η υπόθεση H0 και να γίνει αποδεκτή η εναλλακτική υπόθεση H1. Η τάση για κάθε έργο παράγεται από το πρόσημο του αριθμού του Slope δηλαδή της κλίσης του. Αν είναι θετικό, θα έχουμε μια θετική τάση, αν είναι αρνητικό θα έχουμε μια αρνητική τάση . Η κλίση που λαμβάνεται από τον εκτιμητή είναι η ενδιάμεση κλίση μεταξύ όλων των γραμμών μέσω όλων των σημείων στο σύνολο δεδομένων. [28]

5.2.1 Νόμος 1: Συνεχής Αλλαγή

Για να ελέγξουμε την επικύρωση του 1ου νόμου, πήραμε τη μετρική "Days Between Releases" (DBR) και έγινε στατιστικός έλεγχος. Όπως παρατηρούμε στο Πίνακα7: Αποτελέσματα 1^{ου} Νόμου έχουμε 14 έργα που δεν έχουν κάποια κλίση, το οποίο υποδηλώνει ότι δεν υπάρχουν αποδείξεις για κάποια τάση, σε 9 έργα έχουμε αρνητική τάση και σε 7 έργα θετική τάση. Μια θετική τάση στην DBR υποδηλώνει ότι η συχνότητα με την οποία δημοσιεύονται οι νέες κυκλοφορίες μειώνεται. Πιο συγκεκριμένη μια θετική τάση αποδυναμώνει την εγκυρότητα του νόμου. Ωστόσο, όπως μπορούμε να δούμε, έχουμε περισσότερα έργα με αρνητική τάση που σημαίνει ότι ο νόμος I επικυρώνεται.

Project	DBR				Num_Project	Project	DBR		
	p-value	Trend	Slope(%)				p-value	Trend	Slope(%)
asm	0.114				16	jopt-simple	0.016	↑	0.023
wiquery-core	0.342				17	lucene-core	0.005	↓	-0.016
commons-cli	0.034	↑	0.853		18	log4j-core	0.245		
struts2-core	0.868				19	commons-io	0.026	↑	0.112
guava	0.027	↓	-0.015		20	slf4j-api	0.909		
imagej	0.525				21	itextpdf	0.005	↑	0.042
plexus-utils	<0.0001	↑	1		22	superfly-remote-api	0.076		
poi	0.02	↑	0.022		23	geotk-utility	0.317		
guice-bean-binders	0.532				24	javax.servlet-api	0.005	↓	-0.04
wro4j-core	0.001	↓	-0.031		25	commons-logging	0.273		
xstream	0.002	↓	-0.072		26	httpclient	0.216		
jdom	0.001	↓	-0.022		27	velocity	0.002	↓	-0.032
commons-lang3	0.637				28	commons-math3	0.527		
spring-core	0.009	↑	0.007		29	avro	0.003	↓	-0.073
joda-time	0.039	↓	-0.018		30	httpcore	0.201		

Πίνακας 7: Αποτελέσματα 1ου Νόμου



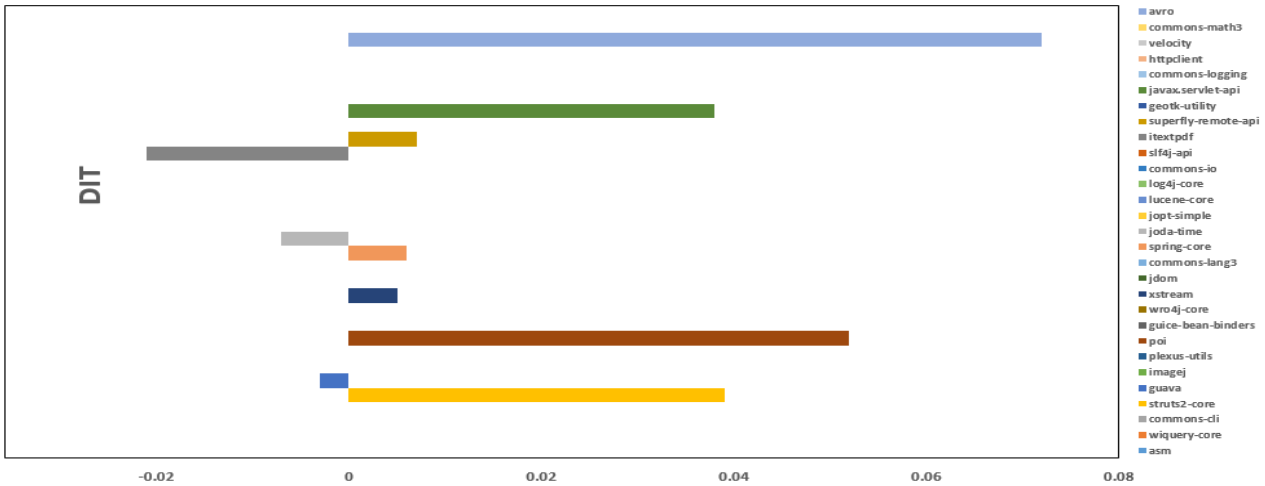
Εικόνα 44: Γραφική Παράσταση (DBR)

5.2.2 Νόμος 2: Αύξηση της Πολυπλοκότητας

Για να ελέγξουμε την επικύρωση του 2ου νόμου, πήραμε τη μετρική " Depth of Inheritance Tree " (DIT) και έγινε στατιστικός έλεγχος. Όπως μπορούμε να παρατηρήσουμε στο Πίνακα 8: Αποτελέσματα 2ου Νόμου σε 20 δείγματα, δεν έχουμε καμιά κλίση που σημαίνει ότι δεν υπάρχουν στοιχεία στατιστικής τάσης, σε 3 έργα έχουμε αρνητική τάση και σε 7 θετική. Μια αρνητική τάση ή μια καθόλου τάση σημαίνει ότι η πολυπλοκότητα παραμένει στα ίδια επίπεδα και αποδυναμώνει την εγκυρότητα του νόμου. Από τη γραφική παράσταση (Εικόνα 15) μπορούμε να διαπιστώσουμε ότι τα περισσότερα έργα είναι με καμία τάση, άρα μη επικύρωση του νόμου. Ίσως να υπάρχει προσπάθεια συντήρησης από τους προγραμματιστές για να διατηρήσουν χαμηλό το επίπεδο πολυπλοκότητας του λογισμικού που παράγουν. Συμπερασματικά, ο νόμος ΙΙ δεν είναι στατιστικά επικυρωμένος.

Num_Project	Project	DIT				Num_Project	Project	DIT		
		p-value	Trend	Slope(%)				p-value	Trend	Slope(%)
1	asm	0.804				16	jopt-simple	0.072		
2	wiquery-core	0.099				17	lucene-core	0.283		
3	commons-cli	0.797				18	log4j-core	0.642		
4	struts2-core	<0.0001	↑	0.039		19	commons-io	N/A	N/A	N/A
5	guava	<0.0001	↓	-0.003		20	slf4j-api	N/A	N/A	N/A
6	imagej	0.534				21	itextpdf	<0.0001	↓	-0.032
7	plexus-utils	0.11				22	superfly-remote-api	<0.0001	↑	0.007
8	poi	<0.0001	↑	0.052		23	geotk-utility	0.317		
9	guice-bean-binders	0.056				24	javax.servlet-api	0.003	↑	0.038
10	wro4j-core	0.192				25	commons-logging	0.121		
11	xstream	0.01	↑	0.005		26	httpclient	0.000		0
12	jdom	0.083				27	velocity	0.434		
13	commons-lang3	N/A	N/A	N/A		28	commons-math3	0.673		
14	spring-core	0.004	↑	0.006		29	avro	0.005	↑	0.072
15	joda-time	<0.0001	↓	-0.007		30	httpcore	0.000		0

Πίνακας 8: Αποτελέσματα 2ου Νόμου



Εικόνα 45: Γραφική Παράσταση (DIT)

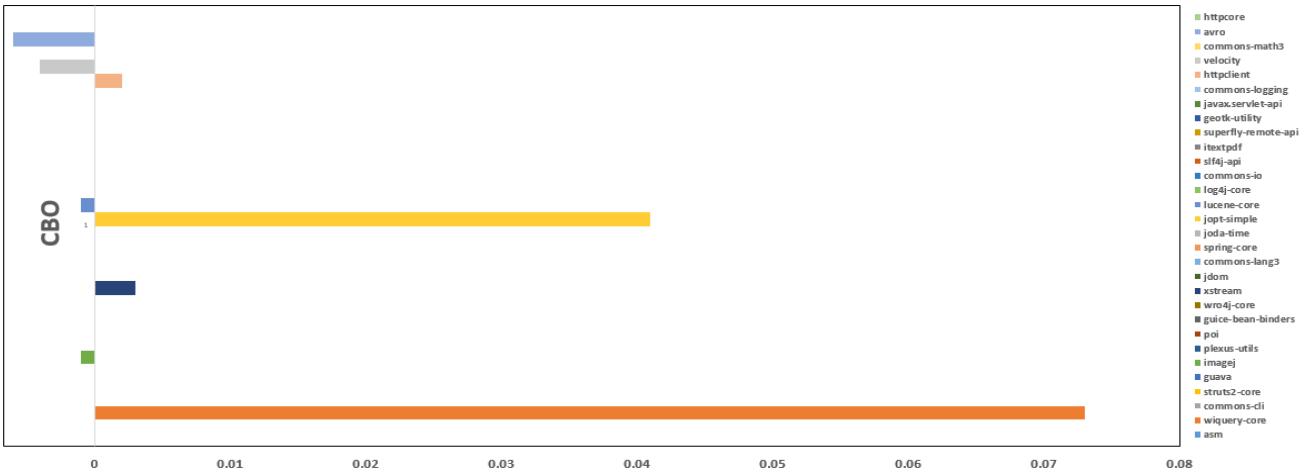
5.2.3 Νόμος 5: Διατήρηση της Εξοικείωσης

Για να ελέγξουμε την επικύρωση του 5ου νόμου, πήραμε τη μετρική "Coupling Between Object Classes" (CBO) και έγινε στατιστικός έλεγχος. Όπως μπορούμε να παρατηρήσουμε στο Πίνακα 9: Αποτελέσματα 5ου Νόμου σε 22 δείγματα, δεν έχουμε καμιά κλίση που σημαίνει ότι δεν υπάρχουν στοιχεία στατιστικής τάσης, σε 4 έργα έχουμε αρνητική τάση και σε 4 θετική. Το 5ο νόμο μπορούμε να το μετρήσουμε και την μετρική DBR που ελέγξαμε το 1ο νόμο. Συγκεντρώσαμε τα στατιστικά αποτελέσματα των 2 και παρατηρήσαμε ότι ορισμένα έργα που

έχουν μια θετική τάση σημαίνει ότι ο αριθμός των λειτουργιών προστίθεται. Από την άλλη πλευρά, άλλα έργα που έχουν μια αρνητική τάση σημαίνει ότι λιγότερες λειτουργίες προστίθενται ή αλλάζουν με την πάροδο του χρόνου και το έργο είναι σχεδόν εξαφανισμένο. Δεν υπάρχουν αρκετά στοιχεία για την επικύρωση του νόμου. Εν κατακλείδι, ο νόμος V δεν είναι στατιστικά επικυρωμένος.

Project	CBO				Num_Project	Project	CBO		
	p-value	Trend	Slope(%)				p-value	Trend	Slope(%)
asm	0.116				16	jopt-simple	0.009	↑	0.041
wiquery-core	0.008	↑	0.073		17	lucene-core	0.000	↓	-0.001
commons-cli	0.061				18	log4j-core	0.63		
struts2-core	<0.0001				19	commons-io	N/A	N/A	N/A
guava	N/A	N/A	N/A		20	slf4j-api	<0.0001		
imagej	<0.0001	↓	-0.001		21	itextpdf	<0.0001		
plexus-utils	<0.0001				22	superfly-remote-api	<0.0001		
poi	0.003				23	geotk-utility	0.317		
guice-bean-binders	0.119				24	javax.servlet-api	N/A	N/A	N/A
wro4j-core	0.136				25	commons-logging	0.9		
xstream	0.044	↑	0.003		26	httpclient	<0.0001	↑	0.002
jdom	0.083				27	velocity	0.001	↓	-0.004
commons-lang3	N/A	N/A	N/A		28	commons-math3	0.061		
spring-core	0.194				29	avro	0.002	↓	-0.006
joda-time	0.009				30	httpcore	<0.0001		

Πίνακας 9: Αποτελέσματα 5ου Νόμου



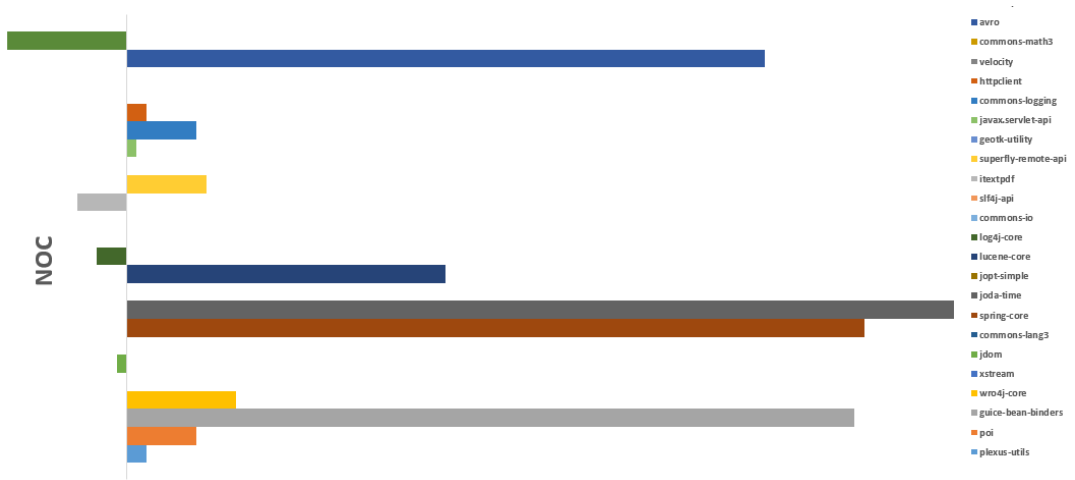
Εικόνα 46: Γραφική Παράσταση (CBO)

5.2.4 Νόμος 6: Συνεχής Ανάπτυξη

Για να ελέγξουμε την επικύρωση του 6ου νόμου, πήραμε τη μετρική “ Number of Children" (NOC) και έγινε στατιστικός έλεγχος. Όπως μπορούμε να παρατηρήσουμε στο Πίνακα 10: Αποτελέσματα 6ου Νόμου σε 12 δείγματα, δεν έχουμε καμιά κλίση που σημαίνει ότι δεν υπάρχουν στοιχεία στατιστικής τάσης, σε 4 έχουμε αρνητική τάση και σε 14 θετική. Η θετική τάση σημαίνει ότι γίνεται επαναχρησιμοποίηση κώδικα μέσω κληρονομικότητας. Από τη γραφική παράσταση (Εικόνα 17) φαίνεται να υπάρχει μια αυξανόμενη τάση σε σχέση με τα έργα που είναι αρνητική η τάση οπότε συμπερασματικά, ο νόμος VI είναι στατιστικά επικυρωμένος.

Project	NOC				Num_Project	Project	NOC		
	p-value	Trend	Slope(%)				p-value	Trend	Slope(%)
asm	0.804				16	jopt-simple	0.000		
wiquery-core	0.02	↑	0.041		17	lucene-core	0.007	↑	0.032
commons-cli	0.199				18	log4j-core	0.007	↓	-0.003
struts2-core	<0.0001	↑	0.005		19	commons-io	N/A	N/A	N/A
guava	0.141				20	slf4j-api	N/A	N/A	N/A
imagej	N/A	N/A	N/A		21	itextpdf	<0.0001	↓	-0.005
plexus-utils	0.006	↑	0.002		22	superfly-remote-api	<0.0001	↑	0.008
poi	<0.0001	↑	0.007		23	geotk-utility	0.317		
guice-bean-binders	0.028	↑	0.073		24	javax.servlet-api	0.000	↑	0.001
wro4j-core	<0.0001	↑	0.011		25	commons-logging	0.004	↑	0.007
xstream	0.411				26	httpclient	<0.0001	↑	0.002
jdom	0.083	↓	-0.001		27	velocity	0.664		
commons-lang3	N/A	N/A	N/A		28	commons-math3	0.917		
spring-core	0.05	↑	0.074		29	avro	0.009	↑	0.064
joda-time	0.026	↑	0.083		30	httpcore	0.003	↓	-0.032

Πίνακας 10: Αποτελέσματα 6ου Νόμου



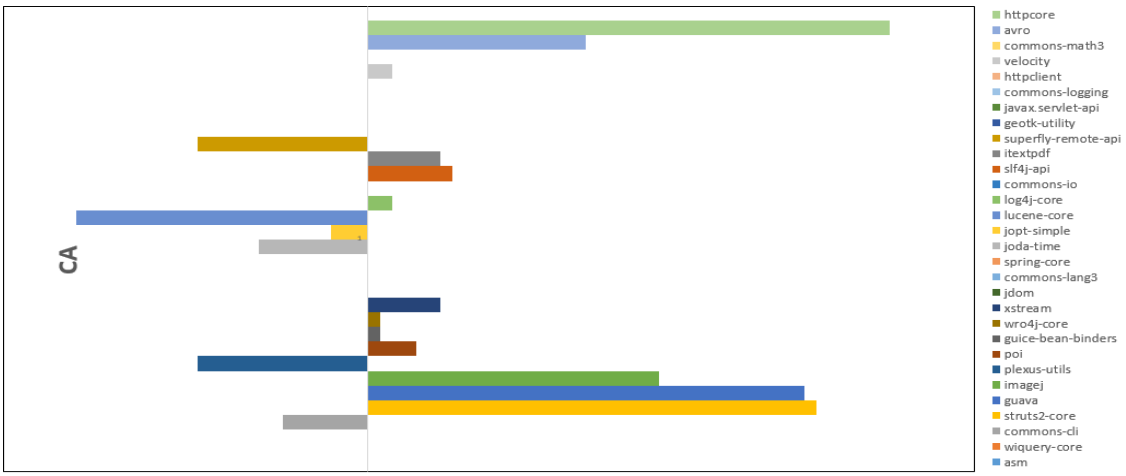
Εικόνα 47: Γραφική Παράσταση (NOC)

5.2.5 Νόμος 7: Φθίνουσα Ποιότητα

Για να ελέγξουμε την επικύρωση του 7ου νόμου, πήραμε τη μετρική “ Afferent coupling " (CA) και έγινε στατιστικός έλεγχος. Όπως μπορούμε να παρατηρήσουμε στο Πίνακα 11: Αποτελέσματα 7^{ου} Νόμου σε 12 δείγματα, δεν έχουμε καμιά κλίση που σημαίνει ότι δεν υπάρχουν στοιχεία στατιστικής τάσης, 6 έργα έχουν αρνητική τάση και 12 θετική τάση. Το 7ο νόμο μπορούμε να το μετρήσουμε και την μετρική DIT που ελέγξαμε το 2ο νόμο. Συγκεντρώσαμε τα στατιστικά αποτελέσματα των δύο και παρατηρήσαμε ότι η ποιότητα είναι σε ίσα επίπεδα και αποδυναμώνει την εγκυρότητα του νόμου. Συμπερασματικά, ο νόμος VII δεν έχει επικυρωθεί στατιστικά.

Num_Project	Project	CA				Num_Project	Project	CA		
		p-value	Trend	Slope(%)				p-value	Trend	Slope(%)
1	asm	0.116				16	jopt-simple	0.005	↓	-0.003
2	wiquery-core	0.095				17	lucene-core	<0.0001	↓	-0.024
3	commons-cli	0.044	↓	-0.007		18	log4j-core	<0.0001		0.002
4	struts2-core	0.004	↑	0.037		19	commons-io	N/A	N/A	N/A
5	guava	<0.0001	↑	0.036		20	slf4j-api	<0.0001	↑	0.007
6	imagej	<0.0001	↑	0.024		21	itextpdf	<0.0001	↑	0.006
7	plexus-utils	<0.0001	↓	-0.014		22	superfly-remote-api	0.026	↓	-0.014
8	poi	<0.0001	↑	0.004		23	geotk-utility	0.317		
9	guice-bean-binders	0.013	↑	0.001		24	javax.servlet-api	N/A	N/A	N/A
10	wro4j-core	<0.0001	↑	0.001		25	commons-logging	0.297		
11	xstream	0.004	↑	0.006		26	httpclient	0.000		
12	jdom	0.083				27	velocity	0.001	↑	0.002
13	commons-lang3	N/A	N/A	N/A		28	commons-math3	0.835		
14	spring-core	0.537				29	avro	<0.0001	↑	0.018
15	joda-time	<0.0001	↓	-0.009		30	httpcore	<0.0001	↑	0.043

Πίνακας 11: Αποτελέσματα 7^{ου} Νόμου



Εικόνα 48: Γραφική Παράσταση (CA)

6. ΕΠΙΛΟΓΟΣ

Στο κεφάλαιο αυτό θα γίνει αναφορά όλων των συμπερασμάτων της μελέτης που προέκυψαν από τη στατιστική ανάλυση που έγινε μέσω του XLSTAT καθώς και σύγκριση με άλλες μελέτες.

6.1 Σύγκριση με άλλες μελέτες και Συμπεράσματα

Στο Πίνακα 12 έχουμε τις μετρικές που χρησιμοποιήθηκαν για τη διερεύνηση των νόμων του Lehman σε προηγούμενες μελέτες.[28]

Αναφορά	I	II	III	IV	V	VI	VII	VIII
Godfrey & Tu	SLOC			SLOC		SLOC		SLOC
Robles et al.	SLOC			SLOC		SLOC		SLOC
Mens et al.	File change	LOC, addition/modification, #defects, CC				Several size measures includind LOC		
Xie et al.	Cumulative #change, type of changes	CC, function calls, coupling	# functions	Changes per day, handled function/total function	#modules, new function	LOC, #function, #definition	#defects, defect density, complexity measures	#function
Israeli & Feitelson	#Source files	CC	#files	Percentage of handled files	Releases per month, intervals between releases	#system calls, #configuration option	Maintainability Index	No quantitative approach
Businge et al.	Cumulative number of added/deleted dependencies		#dependencies		Percentage of handled files, percentage of added dependencies	Unique dependencies	Indicator of balance between abstractness and stability	
Neamtiu et al.	Cumulative changes	Calls per function, CC coupling	#modules #functions	Change per day Change rate Growth rate	Net module growth #new function #changes	LOC #modules #definitions	#defects defect density calls per function CC coupling	#modules LOC #function
Kaur et al.	#Function and #classes	CBO,RFC, WMC, FIT, LOCH	#function and #classes	No quantitative approach	#functions and #classes	LOC, #funtions and #classes	CC	No quantitative
Amanatidis & Chatzigeorgiou	Days between releases	CC	#function	Maintenance effort and #commits	#function	LOC	CA, CE, DIT, CR, MI , bug-related commits	#function
Αυτή η εργασία	Days between releases (DBR)	DIT			CBO,DBR	NOC	CA,DIT	

Πίνακας 12: Μετρικές σε προηγούμενες μελέτες

Εάν συγκρίνουμε τα αποτελέσματά μας με τις άλλες μελέτες με βάση το Πίνακα 13: Επικύρωση νόμων του Lehman σε προηγούμενες μελέτες, καταλήγουμε στις ακόλουθα συμπεράσματα:

- Ο 1ος νόμος είναι επικυρωμένος σε όλες τις γλώσσες.
- Ο 2ος νόμος δεν είναι επικυρωμένος σε Java και PHP. Είναι επικυρωμένος σε C και C++.
- Ο 3ος νόμος είναι επικυρωμένος σε C, Java, C++ και PHP.
- Ο 4ος νόμος είναι επικυρωμένος σε C και PHP. Σε C++ και Java δεν είναι επικυρωμένος και σε κάποιες μελέτες σε C και C++ δεν επαληθεύουν το νόμο.
- Ο 5ος νόμος είναι επικυρωμένος σε C++ και PHP. Δεν είναι επικυρωμένος σε Java και σε C ενώ δεν επαληθεύει το νόμο μια μελέτη σε C.
- Ο 6ος νόμος είναι επικυρωμένος σε όλες τις γλώσσες.
- Ο 7ος νόμος είναι επικυρωμένος σε C++. Δεν είναι επικυρωμένος σε C και Java ενώ δεν επαληθεύουν το νόμο 2 μελέτες σε Java και PHP.
- Ο 8ος νόμος δεν είναι επικυρωμένος σε C++, Java και PHP ενώ δεν επαληθεύει το νόμο μια μελέτη σε C++. Είναι επικυρωμένος σε C

Αναφορά	Έτος	Γλώσσα προγραμματισμου	Αριθμός έργων	I	II	III	IV	V	VI	VII	VIII
Godfrey & Tu	2000	C	1	✓			✗		✓		✗
Robles et al.	2005	C, C++, Java	19	✓			✗		✓		✗
Mens et al.	2008	Java	1	✓	✗				✓		
Xie et al.	2009	C	7	✓	✓	✓	~	✗	✓	✗	✗
Israeli & Feitelson	2010	C	1	✓	✗	✓	✓	~	✓	✗	✓
Businge et al.	2010	Java	21	✓		✓		✗	✓	~	
Neamtiu et al.	2013	C	9	✓	✗	✗	✗	✗	✓	✗	✗
Kaur et al.	2014	C++	2	✓	✓	✓	~	✓	✓	✓	~
Amanatidis & Chatzigeorgiou	2015	PHP	30	✓	✗	✓	✓	✓	✓	~	✗
Αυτή η εργασία	2019	Java	130	✓	✗			✗	✓	✗	

Πίνακας 13: Επικύρωση νόμων του Lehman σε προηγούμενες μελέτες

Σύμφωνα με τη μελέτη εξέλιξης έργων Java καταλήξαμε στα ακόλουθα αποτελέσματα όπως είναι στο Πίνακα 14

Υπεύθυνο για :	**Νόμοι – Απαιτήσεις (Lehman)**	**Τα αποτελέσματα μας για έργα Java**
Ποιότητα	**II** - Αύξηση πολυπλοκότητας	Δεν ισχύει
	VII - Πτώση ποιότητας	Δεν ισχύει
Αλλαγές	**I** - Το σύστημα αλλάζει συνεχώς	Ισχύει
	IV - Το ποσοστό εργασίας παραμένει σταθερό	Κανένα αποτέλεσμα
	V - Οι αυξητικές αλλαγές παραμένουν αμετάβλητες	Δεν ισχύει
Ανάπτυξη	**III** - Η αυξητική ανάπτυξη εξάγει αρνητικές και θετικές προσαρμογές	Κανένα αποτέλεσμα
	VI - Το σύστημα αυξάνεται συνεχώς	Ισχύει
	VIII - Ο ρυθμός αύξησης μειώνεται με ρυθμό ανάλογο του $t^{-2/3}$	Κανένα αποτέλεσμα

Πίνακας 14: Αποτελέσματα σε έργα Java

ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΑΝΑΦΟΡΕΣ

[1] Kindler, E.; Krivy, I. (2011). "Object-Oriented Simulation of systems with sophisticated control". International Journal of General Systems: 313–343.

[2] Lewis, John; Loftus, William (2008). Java Software Solutions Foundations of Programming Design 6th ed. Pearson Education Inc. ISBN 978-0-321-53205-3., section 1.6 "Object-Oriented Programming"

[3] Basic Concepts in Object Oriented Programming. By Kristen Nygaard, University of Oslo, Norway.(October 1986)

[4] Kristen Nygaard (1978). "The Development of the SIMULA Languages" by Kristen Nygaard Norwegian Computing Center and University of Oslo and Ole-Johan dahl University of Oslo..

[5]Kay, Alan. "The Early History of Smalltalk". Retrieved 2007-09-13.

[6]Σφέτσος,Παναγιώτης (2007, Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης (ΑΠΘ)), Διδακτορική διατριβή Πειραματικές μέθοδοι στην αντικειμενοστρεφή τεχνολογία λογισμικού και στις ευέλικτες προγραμματιστικές πρακτικές

[7] Object-Oriented Programming: Themes and Variations. Mark Stefik & Daniel G. Bobrow Intelligent Systems Laboratory, Xerox Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, California 94304

[8]"Dr. Alan Kay on the Meaning of "Object-Oriented Programming"". 2003. Retrieved 11 February 2010.

[9] Introduction to Computation and Programming Using Python. Revised and Expanded Edition by John v. Guttag

[10] Object-Oriented Thought Process (book), The, 4th Edition By Matt Weisfeld Published Mar 13, 2013 by Addison-Wesley Professional. Part of the Developer's Library series.

[11]Evolving a language in and for the real world: C++ 1991-2006 Bjarne StroustrupTexas A&M University www.research.att.com/~bs

[12] JSR-384 Java SE 11 (18.9) ("Specification")Version: 11Status: Final ReleaseSpecification Lead: Oracle America, Inc. ("Specification Lead")Release: September 2018Copyright © 1997, 2018, Oracle America, Inc.

[13] Computer Weekly. May 2, 2002. Retrieved 2009-07-27.

[14] RedMonk Index Archived March 26, 2015, at the Wayback Machine (Stephen O'Grady, January 2015)

[15] "Programming Language Popularity". langpop.com. October 25, 2013. Archived from the original on January 16, 2009. Retrieved April 2, 2015. Normalized Comparison: 1st C, 2nd Java, 3rd PHP

[16] J. Gosling, H. McGilton, "The Java Language Environment", 1996. Sun Microsystems, Inc. All rights reserved.2550 Garcia Avenue, Mountain View, California 94043-1100 U.S.A.

[17] Byous, Jon (c. 1998). "Java technology: The early years". Sun Developer Network. Sun Microsystems.

[18] Object-oriented programming "The History of Java Technology". Sun Developer Network. c. 1995.

[19] "So why did they decide to call it Java? Archived November 15, 2013, at the Wayback Machine", Kieron Murphy

[20]The case for Java as a programming language A. Van Hoff Marimba Inc., Palo alto, CA, USA IEEE Computer Society DOI: 10.1109/4236.585172

[21] "JAVASOFT SHIPS JAVA 1.0". Retrieved 2018-05-13.

[22] St. Laurent, Andrew M. (2008). Understanding Open Source and Free Software Licensing. O'Reilly Media. p. 4. ISBN 9780596553951.

[23] "What is open source?". opensource.com. Retrieved 25 August 2017.

[24] Open Source Software:A Survey from 10,000 Feet Foundations and TrendsR© inTechnology, Information and Operations ManagementVol. 4, Nos. 3–4 (2010) 187–347c©2011 S. Androutsellis-Theotokis, D. Spinellis,M. Kechagia and G. GousiosDOI: 10.1561/0200000026

[25] The Rise of Open Source Licensing Mikko VälimäkiA Challenge to the Use of Intellectual Property in the SoĞ ware Industry PUBLISHED BYTURREPUBLISHING,A DIVISION OFTURRELEGALLTD. Aleksanterinkatu 17, 6th floor, Helsinki, FI-00100, Finland, http://pub.turre.com/

[26] Open Source Projects as Incubators of InnovationFrom Niche Phenomenon to Integral Part of the Software Industry Jan-Felix Schrape SOI Discussion Paper 2017-03 University of StuttgartInstitute for Social Sciences Department of Organizational Sociology and Innovation StudiesSeidenstr. 36D-70174 Stuttgart

[27] Understanding Free/OpenSource SoftwareDevelopment Processes Softw. Process Improve. Pract.2006;11: 95 – 105Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/spip.255

[28] Studying the evolution of PHP web applications Theodoros Amanatidis, Alexander Chatzigeorgiou∗ Department of Applied Informatics, University of Macedonia, Thessaloniki, Greece

[29]Fred Brooks, in his key book The Mythical Man-Month,[1] states that over 90% of the costs of a typical system arise in the maintenance phase, and that any successful piece of software will inevitably be maintained.

[30] Software Engineering — Software Life Cycle Processes — Maintenance ISO/IEC 14764:2006,

[31] Software Evolution: Past, Present and Future 1P.I. Okwu2 and I.N. Onyeje American Journal of Engineering Research (AJER) e-ISSN : 2320-0847 p-ISSN : 2320-0936 Volume-03, Issue-05

[32]Lehman, Meir M. (1980). "Programs, Life Cycles, and Laws of Software Evolution". Proc. IEEE. 68 (9): 1060–1076. doi:10.1109/proc.1980.11805.

[33]Lehman, M. M.; J. F. Ramil; P. D. Wernick; D. E. Perry; W. M. Turski (1997). "Metrics and laws of software evolution—the nineties view" (PDF). Proc. 4th International Software Metrics Symposium (METRICS '97). pp. 20–32. doi:10.1109/METRIC.1997.637156.

[34]Herraiz, Israel; Rodriguez, Daniel; Robles, Gregorio; Gonzalez-Barahona, Jesus M. (2013). "The evolution of the laws of software evolution". ACM Computing Surveys. 46 (2): 1–28. doi:10.1145/2543581.2543595. ISSN 0360-0300.

[35] On the Evolution of Lehman's Laws Michael W. Godfrey David R. Cheriton School of Computer Science University of Waterloo, CANADA Daniel M. German Department of Computer Science University of Victoria, CANADA

[36] Validating Software Evolution of Agile Projects Using Lehman Laws A Dissertation Submitted In Partial Fulfillment Of The Requirement For The Award Of The Degree Of Master Of Technology In Computer Science And Engineering Submitted BY Gurpreet Kour Roll No. 11203037

[37] Mens T. (2008) Introduction and Roadmap: History and Challenges of Software Evolution. In: Software Evolution. Springer, Berlin, Heidelberg Springer-Verlag Berlin Heidelberg 2008 ISBN 978-3-540-76439-7

[38] Alfred Holl Felix Paetzold Robert Breun Cooperative, cyclic-iterative knowledge gain in information systems anti-aging Georg Simon Ohm University of Applied Sciences Nuremberg, Germany 2011 ISBN 978-3-00-034697-2

[39]SOFTWARE EVOLUTION AND FEEDBACK Theory and Practice Edited by Nazim H. Madhavji University of Western Ontario, Canada Juan C. Fern ´andez-Ramil The Open University, UK and Dewayne E. Perry The University of Texas at Austin, USA

[40] Implications of Laws of Software Evolution on Continuing Successful Use of COTS Software M M LehmanDepartment of ComputingImperial CollegeLondon SW7 2BZ J F RamilDepartment of ComputingImperial CollegeLondon SW7 2BZ

[41] The evolution of the laws of software evolution. A discussion basedon a systematic literature review.ISRAEL HERRAIZ, Technical University of Madrid, SpainDANIEL

RODRIGUEZ, University of Alcala, Madrid, SpainGREGORIO ROBLES and JESUS M. GONZALEZ-BARAHONA, GSyC/Libresoft, UniversityRey Juan Carlos, Madrid, Spain

[42]Liguo Yu and Alok Mishra (2013) An Empirical Study of Lehman's Law on Software Quality Evolution in International Journal of Software and Informatics, 11/2013; 7(3):469-481.

[43] Applicability of Lehman Laws on Open Source Evolution: A Case study International Journal of Computer Applications (0975 – 8887) Volume 93 – No. 18, May 2014 Taranjeet Kaur M.Tech. Student, Nisha Ratti Assistant Professor, Parminder Kaur Assistant Professor Dept. of CSE & IT, RIEIT, Railmajra, Punjab, India

[44] Reuse Dimensions. Liesbeth Dusink ,Jan van Katwijk August 1995ACM SIGSOFT Software Engineering Notes 20:137-149 DOI: 10.1145/211782.211828

[45] M.D. McIlroy, Mass produced software components, Software Engineering: Report on a conference by the NATO Science Committee, Garmisch, Germany, 7th to 11th October 1968, Naur, P., Randell, B., Eds., NATO Scientific Affairs Division, Brussels, Belgium, 1969

[46] Structuring Software Reusability Metrics for Component-Based Software Development Danail Hristov, Oliver Hummel, Mahmudul Huq, Werner Janjic Software Engineering Group University of Mannheim Mannheim, Germany

[47] Software Reuse:From Library to Factory MartinL.Griss SoftwareTechnology Laboratory HPL-93-67 July,1993

[48] Structuring Software Reusability Metrics for Component-Based Software Development Danail Hristov, Oliver Hummel, Mahmudul Huq, Werner Janjic Software Engineering Group University of Mannheim Mannheim, Germany

[49] Code Reuse & Reusability of the Software Ms.Kirandeep Kaur, Ms.Rekha Rani, Ms.Jagdeep kaur (CSE, KCCEIT, Nawanshahrr /PTU, INDIA) The International Journal Of Engineering And Science (IJES) Volume 2 Issue 4| Pages 28-30, 2013 ISSN(e):2319 –1813 ISSN(p):2319 –1805

[50] Johannes Sametinger Softwar e Engineering with Reusable Components March 3, 1997 Springer- Verlag Berlin Heidelberg New York London Paris Tokyo Hong Kong Barcelona Budap est

[51] A PRAGMATIC APPROACH TO SOFTWARE REUSE B.JALENDER 1,Dr. A GOVARDHAN 2, Dr.P PREMCHAND 31 Asst Professor, Department of IT, VNRVJIET, Hyderabad, India-500090. 2 Principal, JNTU College of Engineering, Jagtial, Karimnagar, AP, India. 3 Professor, Department of CSE, Osmania University, Hyderabad, India.

[52] Diomidis Spinellis. ckjm — a tool for calculating Chidamber and Kemerer Java metrics. Technical report, Athens University of Economics and Business, Athens, Greece, 2006

[53] Diomidis Spinellis. Tool writing: A forgotten art? IEEE Software, 22(4):9–11, July/August 2005. (doi:10.1109/MS.2005.111).

[54] Shyam R. Chidamber and Chris F. Kemerer, C. F. (1994), A metrics suite for object-oriented design. IEEE Transaction on. Software Engineering. Vol 20 No.6,June 1994,Page No. 476–493.

[55] JavaScript:The World's Most Misunderstood Programming Language Douglas Crockford

[56] “What is MySQL?” MySQL 5.1 Reference Manual. Oracle. Retrieved 17 September 2012. The official way to pronounce “MySQL” is “My Ess Que Ell” (not “my sequel”)

[57] "Interview with Kai Seidler from the XAMPP project." MySQL AB.

[58] Apache Web server Complete Guide Dedoimedo

[59] Software Evolution Muhammad Asim, Studying Computer Sciences at Comsats Institute of Information Technology www.slideshare.net/asim741/software-evolution-58237445

[60] www.csetutor.com/advantages-and-disadvantages-of-object-oriented-programming-language/ Τελευταία ανάκτηση: 05/06/2018

[61] Σχεδιαστικά Πρότυπα, Απόστολος Ζάρρας, Λέκτορας Πανεπιστημίου Ιωαννίνων, www.cs.uoi.gr/~zarras/dpatts.htm Τελευταία ανάκτηση: 13/06/2018

[62] https://ellak.gr/2015/09/ti-ine-to-logismiko-aniktou-kodika-mia-isagogi/ Τελευταία ανάκτηση: 14/06/2018

[63] https://opensource.com/resources/what-open-source Τελευταία ανάκτηση: 14/06/2018

[64] https://www.cloudrepo.io/articles/what-is-a-maven-repository.html#what-is-a-maven-repository Τελευταία ανάκτηση: 20/06/2018

[65] https://maven.apache.org/guides/introduction/introduction-to-repositories.html Τελευταία ανάκτηση: 20/06/2018

[66] https://opensource.org Τελευταία ανάκτηση: 03/07/2018

[67] https://simple.wikipedia.org/wiki/Entity-relationship_model Τελευταία ανάκτηση:11/07/2018

[68] http://php.net/manual/en/history.php.phpΤελευταία ανάκτηση: 11/07/2018

[69] https://en.wikipedia.org/wiki/PHP Τελευταία ανάκτηση: 11/07/2018

[70] https://el.wikipedia.org/wiki/HTML Τελευταία ανάκτηση: 15/07/2018

[71] https://el.wikipedia.org/wiki/SQL Τελευταία ανάκτηση: 15/07/2018

[72] https://el.wikipedia.org/wiki/CSS Τελευταία ανάκτηση: 15/07/2018

[73] https://www.phpmyadmin.net/ Τελευταία ανάκτηση: 16/07/2018

[74] https://en.wikipedia.org/wiki/XAMPP Τελευταία ανάκτηση: 16/07/2018

[75] https://www.apachefriends.org/index.html Τελευταία ανάκτηση: 16/07/2018

[76] https://www.mysql.com/customers/ Τελευταία ανάκτηση: 18/07/2018

[77] https://el.wikipedia.org/wiki/MySQL Τελευταία ανάκτηση: 18/07/2018

[78] https://el.wikipedia.org/wiki/Microsoft_Excel Τελευταία ανάκτηση: 20/07/2018

[79] https://www.xlstat.com Τελευταία ανάκτηση: 20/07/2018