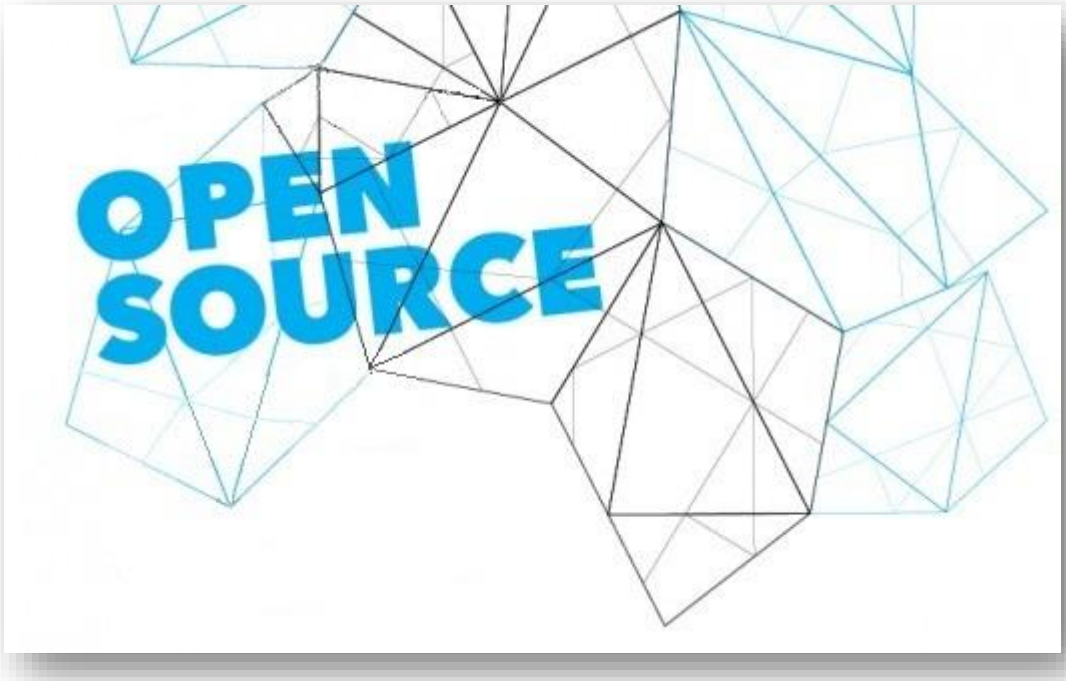
ΑΛΕΞΑΝΔΡΕΙΟ ΤΕΙ ΘΕΣΣΑΛΟΝΙΚΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΓΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΑΥΤΟΜΑΤΟΠΟΙΗΜΕΝΗ ΑΝΑΚΤΗΣΗ ΜΕΤΡΙΚΩΝ ΚΟΙΝΟΤΗΤΑΣ ΕΛΛΑΚ



Του φοιτητή:
Τσακιλτσίδη Σωκράτη
αρ. μητρώου 04/2647

Επιβλέπων καθηγητής:
Δρ. Αμπατζόγλου Απόστολος

ΠΕΡΙΛΗΨΗ

Αντικείμενο της παρούσας εργασίας είναι η δημιουργία ενός εργαλείου για την άντληση - συλλογή πληροφοριών και μετρικών από την σελίδα του *sourceforge.net* για έργα ανοιχτού λογισμικού. Σκοπός του συστήματος που σχεδιάστηκε είναι να απλοποιήσει τον τρόπο παρουσίασης των πληροφοριών και των μετρικών που προσφέρει το *sourceforge* και να κάνει αυτές τις πληροφορίες άμεσα διαθέσιμες στον χρήστη. Η ανάγκη δημιουργίας ενός τέτοιου συστήματος προκύπτει από το μέγεθος της πληροφορίας που υπάρχει στο διαδίκτυο και στη συγκεκριμένη περίπτωση στη σελίδα του *sourceforge.net,* η οποία κρίνεται αναγκαίο να φιλτραριστεί και να παρουσιαστεί στο χρήστη σε μία πιο δομημένη μορφή.

Το σύστημα δέχεται από τον χρήστη το όνομα οποιουδήποτε έργου ανοιχτού λογισμικού αναρτημένου στο *sourceforge.net* και αναλαμβάνει να αναλύσει – αναζητήσει – πλοηγηθεί και να συλλέξει συγκεκριμένες μετρικές, με σκοπό να τις παρουσιάσει σε μια δομημένη και εύχρηστη μορφή. Το σύστημα το οποίο δημιουργήθηκε είναι αυτόνομο γιατί δεν εξαρτάται από κάποιο εξωτερικό πρόγραμμα. Για να επιτευχθεί αυτός ο στόχος, δημιουργήθηκε ένα σύστημα το οποίο ανακτά και κάνει συλλογή της πληροφορίας-μετρικών μέσω του κώδικα html της ιστοσελίδας του έργου.

ABSTRACT

The subject of this project is to create a software that collects data and metrics of open source projects which are hosted on the *sourceforge.net* repository. The system is designed to simplify the presentation community metrics provided by sourceforge and make this information straightly available to the user. The creation of the system has been inspired by the need of filtering and presenting the chaotic amount of information provided by open source repositories, in our case sourceforge's page, in a structured format.

The input of the system is the name of any open source project, which was set by the user, and after analyzing - searching - navigating and collecting specific metrics, the output of the system is their presentation in a structured and manageable form. The provided system is autonomous because it does not depend on any external program. In order to achieve the above goal, we created a parser that recovers and collects the information-metrics through handling the html source code of the projects' website.

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω πολύ τον κ. Αμπατζόγλου Απόστολο που από την πρώτη στιγμή της ανάθεσης της πτυχιακής μου ήταν δίπλα μου, με στήριξε και με βοήθησε πολύ στην υλοποίηση της πτυχιακής.

Επίσης, ένα ευχαριστώ και σε όσους ήταν δίπλα μου όλο αυτό τον καιρό της εκπόνησης της πτυχιακής μου εργασίας, στηρίζοντας και δείχνοντας κατανόηση σε εμένα.

Τσακιλτσίδης Σωκράτης

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΥΡΕΤΗΡΙΟ ΣΧΗΜΑΤΩΝ

ΚΕΦΑΛΑΙΟ 1ο
ΕΙΣΑΓΩΓΗ

Στο έγγραφο αυτό θα γίνει η περιγραφή και η ανάλυση του συστήματος – εργαλείου το οποίο υλοποιήθηκε στα πλαίσια της πτυχιακή εργασίας. Το σύστημα αυτό δημιουργήθηκε με σκοπό να συλλέξει πληροφορίες για έργα ανοιχτού λογισμικού από τη σελίδα του *sourceforge.net*.

Η ανάγκη δημιουργίας αυτού του συστήματος προέκυψε από τα μεγάλα ποσά πληροφορίας που υπάρχουν στη σελίδα του *sourceforge.net* τα οποία πρέπει με κάποιο τρόπο να φιλτραριστούν και να παρουσιαστούν στο χρήστη σε μία πιο δομημένη μορφή. Το σύστημα λοιπόν αυτό συλλέγει τις πληροφορίες από το εκάστοτε έργο ανοιχτού λογισμικού που του έχει ορίσει ο χρήστης και τις παρουσιάζει σε αυτόν σε μια δομημένη και εύχρηστη μορφή. Το περιβάλλον διεπαφής του συστήματος είναι απλό και φιλικό προς το χρήστη στον τρόπο ορισμού του έργου για το οποίο θα γίνει η συλλογή πληροφοριών αλλά και στον τρόπο με τον οποίο εμφανίζει τα αποτελέσματα του.

Η ανάγκη ανάπτυξης συστημάτων για συλλογή πληροφοριών από σελίδες διαδικτύου προέκυψε από την ραγδαία ανάπτυξη του ίδιου του διαδικτύου. Η ανάπτυξη αυτή του διαδικτύου έχει δώσει πλέον στους χρήστες πρόσβαση σε χαοτικά μεγάλη έκταση πληροφορίας στην οποία αν δεν υπάρχει κάποια δομή και οργάνωση χάνει την αξία της.

Στα κεφάλαια που ακολουθούν θα γίνει μία αρχική μελέτη εννοιών, μία γενική επισκόπηση του έργου, μία βιβλιογραφική και ιστορική ανασκόπηση της εξέλιξης λογισμικού, μία λεπτομερής και τεχνική περιγραφή όλων των συστατικών του έργου και τέλος θα γίνει αναφορά σε σχέση με χαρακτηριστικά που θα μπορούσαν να υλοποιηθούν σε επόμενες γενιές του συστήματος αλλά και παραδοχές που χρειάστηκε να γίνουν κατά την υλοποίηση της σημερινής έκδοσης και πως αυτές θα μπορούσαν πιθανόν να αρθούν σε επόμενες εκδόσεις.

1.1 ΑΝΟΙΧΤΟ ΛΟΓΙΣΜΙΚΟ

Ο όρος "ανοιχτό λογισμικό" (free software) εισήχθη από τον Richard Stallman στις αρχές της δεκαετίας του ΄80 ενώ παράλληλα ίδρυσε και το Ίδρυμα Ελεύθερου Λογισμικού (free Software Foundation) το 1983 προκειμένου να βρεθεί ένας τρόπος διατήρησης της ελευθερίας των χρηστών να μελετούν, να

κατανοούν και να τροποποιούν το λογισμικό μέσα στα πλαίσια των κανόνων και του πνεύματος που διέπει τις ομάδες αυτές. Η δημιουργία του Ιδρύματος Ελεύθερου Λογισμικού και η προσπάθεια για την διάδοση αυτού συμπίπτει χρονικά με την περίοδο της αποσύνδεσης του υλικού από το λογισμικό και της προσπάθειας για εμπορευματοποίηση του δεύτερου.

Το 1998 παρουσιάστηκε μια νέα τάση στην παραγωγή λογισμικού, η διαδικασία ανάπτυξης κώδικα ανοιχτού λογισμικού (Open Source Software), την οποία μελέτησαν οι συγγραφείς Feller και Fitzgerald(2002), παρουσιάζοντας πολύ αξιόλογα και πετυχημένα έργα ανοιχτού λογισμικού όπως το λειτουργικό Linux, το πακέτο εργαλείων γραφείου Libre Office, ο φυλλομετρητής Mozilla Firefox καθώς και ο εξυπηρετητής Apache Server.

Το ανοιχτό λογισμικό αποτελεί στην πραγματικότητα μια κίνηση της παγκόσμιας κοινότητας των προγραμματιστών που, υποβοηθούμενη από το διαδίκτυο, θέτει σε αμφισβήτηση τις βάσεις του νομικού συστήματος της προστασίας περί πνευματικής ιδιοκτησίας για το λογισμικό. Το σύνηθες σχήμα της αποζημίωσης του δημιουργού για κάθε χρήση του έργου του, και της προστασίας του έργου του από κάθε επέμβαση τρίτου αναθεωρείται εκ βάθρων. Η κυρίαρχη ιδέα του ελεύθερου λογισμικού είναι ότι καθένας έχει πρόσβαση στο έργο και κανένας δεν έχει, εντέλει, δικαίωμα σε αυτό. Προέρχεται από την κοινότητα και απευθύνεται στην κοινότητα.

Με αυτές βέβαια τις παραδοχές το ανοιχτό λογισμικό κινείται κατ' ανάγκην εκτός των γνωστών νομικών συστημάτων. Η ύπαρξή του προϋποθέτει μια παραίτηση από σειρά δικαιωμάτων που απονέμει στον δημιουργό το ισχύον σύστημα προστασίας της πνευματικής ιδιοκτησίας. Η παραίτηση αυτή παρόλα αυτά υπόκειται σε κάποιους κανόνες που ρυθμίζονται από κάποιες ειδικές άδειες που δημιουργήθηκαν από το ίδρυμα ανοιχτού λογισμικού για το σκοπό αυτό.

Μία από αυτές τις άδειες είναι η GNU ή GPL (General Public Licence). Το λογισμικό που φέρει αυτή την άδεια δίνει τη δυνατότητα στους χρήστες του να μπορούν να το χρησιμοποιήσουν ,να το αλλάξουν, να χρησιμοποιήσουν κομμάτια του για την κατασκευή άλλων ανοιχτών λογισμικών αλλά τους υποχρεώνει και να διατηρούν τον ανοιχτό χαρακτήρα του έργου που θα προκύψει από όλες αυτές τις αλλαγές. Υπάρχουν πάρα πολλές τέτοιες άδειες που ορίζουν κανόνες διασφάλισης έργων ανοιχτού λογισμικού.

Αυτό δεν σημαίνει ότι προβλήματα δεν μπορεί να ανακύψουν. Ο εθελοντισμός και η καλή διάθεση της κοινότητας προγραμματιστών μπορεί να εξαλειφθούν, όταν το οικονομικό κέρδος εμφανιστεί. Τέτοια περίπτωση έχει ήδη συντρέξει. Γι' αυτές τις περιπτώσεις κρίσιμη είναι η άδεια χρήσης του λογισμικού και η εξέταση των γενικών όρων της παραίτησης του δημιουργού από τα περιουσιακά δικαιώματα επί του έργου του.

Το ελεύθερο λογισμικό παρουσιάζει ιδιαίτερο ενδιαφέρον τόσο ως κίνηση και ιδέα, όσο και ως μέθοδος, για μικρότερες αγορές λογισμικού, όπως η δική μας. Σκόπιμη κρίνεται επομένως η παρακολούθηση των σχετικών εξελίξεων, και η

ανάληψη δράσεων για την τόνωση του φαινομένου. Η εκπόνηση νομικού πλαισίου όμως θα πρέπει μάλλον να αναμείνει την οριστικοποίηση των μορφών του και την εγκαθίδρυσή του στην πράξη.

1.2 ΤΟ ΑΝΟΙΧΤΟ ΛΟΓΙΣΜΙΚΟ ΣΤΗΝ ΕΛΛΑΔΑ

Η Ελληνική Εταιρεία Λογισμικού Ανοιχτού Κώδικα, της οποίας πρόεδρος διατελεί σήμερα ο κ. Καρούνος, αριθμεί ως μέλη της περίπου 25 πανεπιστήμια, ΤΕΙ και ερευνητικά κέντρα, με στόχο τη δημιουργία μιας «βιβλιοθήκης προγραμμάτων» που μπορούν να συμβάλλον στην εκπαιδευτική διαδικασία. Βασικός της στόχος είναι να δέχεται προγράμματα που διατίθενται ελεύθερα και να τα διαχύει σε επιστημονικές κοινότητες ώστε να χρησιμοποιηθούν στην παραγωγική διαδικασία. Ταυτόχρονα, στόχος της εταιρείας είναι η επικοινωνία με κοινότητες ανοιχτού λογισμικού και ο συντονισμός τους προς κοινές κατευθύνσεις, κατά βάση εκπαιδευτικού-μαθησιακού περιεχομένου.

Η χρήση λογισμού ανοιχτού κώδικα από τις ελληνικές δημόσιες υπηρεσίες είναι κατά βάση θέμα πολιτικής βούλησης, όπως υποστηρίζεται από την Ελληνική Εταιρεία Λογισμικού Ανοιχτού Κώδικα. Από τη στιγμή που η χώρα μας είναι δεσμευμένη από τη Microsoft με μία σύμβαση σε υψηλότερο επίπεδο, αυτό έχει αναπόφευκτα παρενέργειες στα χαμηλότερα επίπεδα, δηλαδή σε όλα τα πιθανά σημεία που θα μπορούσε να χρησιμοποιηθεί λογισμικό ανοιχτού κώδικα, όπως υπουργεία, σχολεία κτλ.. Σύμφωνα με την Ελληνική Εταιρεία Λογισμικού Ανοιχτού Κώδικα, η Ελλάδα θα πρέπει αφενός να καταγγείλει τη σύμβαση, όπως έχουν κάνει άλλες ευρωπαϊκές χώρες, και αφετέρου να ορίσει πολιτικές υιοθέτησης του ανοιχτού λογισμικού σε όλους τους τομείς και πάνω απ' όλα στην εκπαίδευση και στις μικρομεσαίες επιχειρήσεις.

Σημαντικότερο όφελος από τη χρήση λογισμικού ανοιχτού κώδικα για μια οικονομία, δεν είναι κατά βάση το οικονομικό, αλλά κυρίως το αναπτυξιακό που εστιάζεται στη δημιουργία μιας ανταγωνιστικής αγοράς πληροφορικής υψηλού επιπέδου.

1.3 Το *sourceforge.net*

Το *sourceforge.net* (εικόνα 1) ανήκει σε μία κατηγορία συστημάτων που είναι γνωστά ως Αποθετήρια Ανοικτού Κώδικα. Σε αυτή τη κατηγορία ανήκουν και το Googlecode, το Bitbucket, το Github και πολλά άλλα.

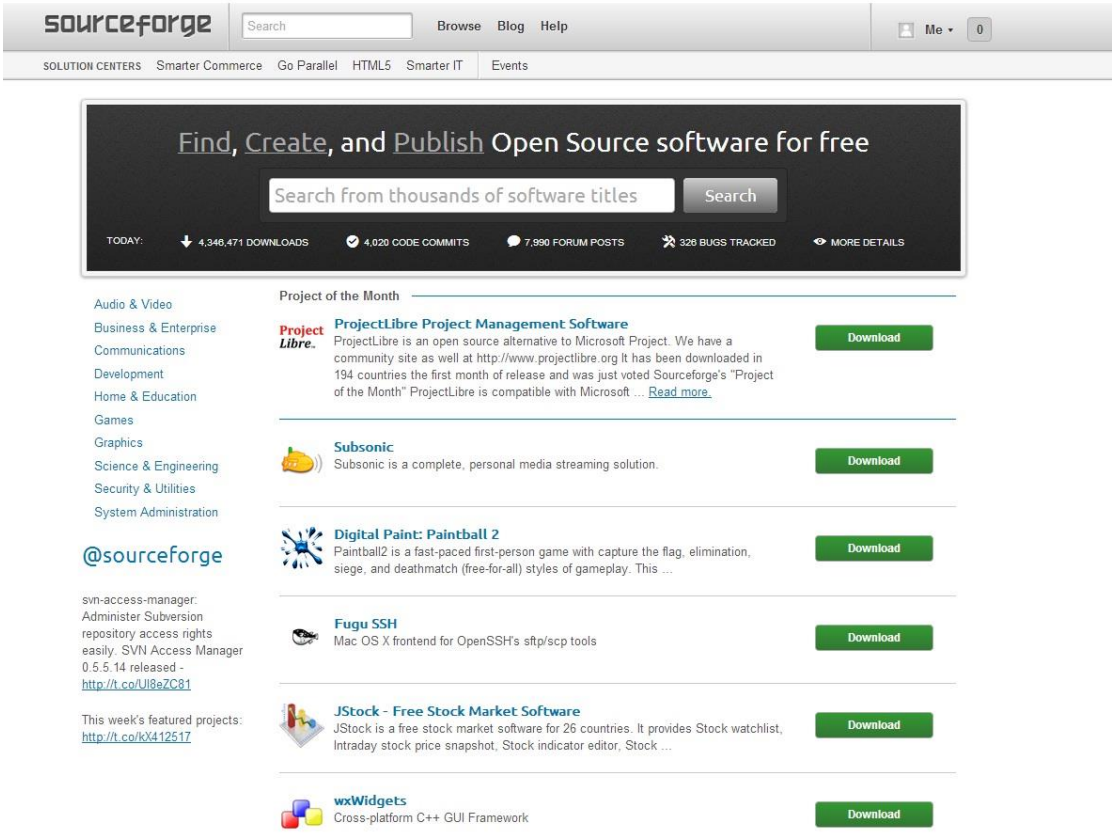
Το *sourceforge.net* ιδρύθηκε το Φεβρουάριο του 2009, από την εταιρεία Geeknet, και από τότε έχει φιλοξενήσει πάνω από 325.000 έργων ανοικτού λογισμικού. Δίνει τη δυνατότητα στους χρήστες του να ανεβάζουν δικά τους έργα

ανοιχτού λογισμικού και να συμμετέχουν σε έργα άλλων. Επίσης δίνει τη δυνατότητα σε όλους τους χρήστες αλλά και στους απλούς επισκέπτες να κατεβάζουν έργα αλλά και να έχουν πρόσβαση στο κώδικα τους. Στο *sourceforge.net* φιλοξενούνται μερικά από τα πιο δημοφιλή έργα ανοικτού λογισμικού στο κόσμο όπως το Mozilla Firefox, το Wampserver και το 7-zip. Στο site παρέχεται μία μηχανή αναζήτησης που δίνει τη δυνατότητα στο χρήστη να ψάξει για οποιαδήποτε κατηγορία λογισμικού τον ενδιαφέρει.

Επίσης ο χρήστης μπορεί να διαβάσει γενικές πληροφορίες σχετικές με το πρόγραμμα που τον ενδιαφέρει όπως:

- η λειτουργικότητα του έργου
- τα γενικά χαρακτηριστικά του
- την ημερομηνία της πρώτης έκδοσης του έργου
- κριτικές των χρηστών
- αναφορές σφαλμάτων

και ακόμη περισσότερες πληροφορίες που μπορεί να τον ενδιαφέρουν. Μία ακόμη δυνατότητα που έχει ο χρήστης είναι να παρακολουθεί την πορεία ενός έργου και να βλέπει κατά πόσο ένα έργο είναι ενεργό και πόση δουλειά έχει γίνει για τη βελτίωση του. Μία άλλη σημαντική λειτουργία που πρέπει να αναφέρουμε ενδεικτικά είναι ότι ο χρήστης έχει τη δυνατότητα να είναι ενεργό μέλος της κοινότητας του *sourceforge.net*, αναφέροντας και ο ίδιος σφάλματα, κάνοντας κριτικές αλλά μπορεί και να ζητήσει ή να προτείνει νέες δυνατότητες για επόμενες εκδόσεις του έργου. Φυσικά είναι ευκολονόητο ότι δεν γίνεται να αναφερθούν αναλυτικά όλες οι δυνατότητες που προσφέρει το site στο χρήστη. Παρόλα αυτά οι συγκεκριμένες λειτουργίες που αναφέρθηκαν είναι χρήσιμες για την παρουσίαση του συστήματος του οποίου παρουσιάζεται και αυτό θα φανεί πιο ξεκάθαρα στη συνέχεια του εγγράφου.



εικόνα 4. sourceforge.net

1.3 PARSING

Στη σημερινή εποχή, την εποχή του διαδικτύου και της πληροφορίας, το μόνο σίγουρο είναι ότι μπορούμε να έχουμε πληροφορίες για οτιδήποτε και αν μας ενδιαφέρει και αυτό μπορεί ο καθένας να το δοκιμάσει γράφοντας απλά μία λέξη που τον ενδιαφέρει στη φόρμα οποιασδήποτε μηχανής αναζήτησης. Ένα από τα ζητούμενα που προκύπτουν από όλο αυτό τον όγκο πληροφορίας είναι ο τρόπος με τον οποίο μπορούμε να συλλέξουμε και να χρησιμοποιήσουμε όλη αυτή την πληροφορία με συστηματικό τρόπο, έτσι ώστε να μπορούμε να φιλτράρουμε τις πληροφορίες που μας ενδιαφέρουν από τον υπόλοιπο όγκο που μας είναι αδιάφορος. Μία τέτοια ανάγκη προκύπτει πιο εξειδικευμένα στην αναζήτηση πληροφορίας σε έργα ανοιχτού λογισμικού. Πάνω σε αυτό το ζητούμενο σχεδιάστηκε και υλοποιήθηκε το σύστημα αυτοματοποιημένης ανάκτησης μετρικών. Το σύστημα αυτό σχεδιάστηκε και υλοποιήθηκε για να αντλεί πληροφορίες για έργα ανοικτού λογισμικού τα οποία είναι δημοσιευμένα στην σελίδα του *sourceforge.net* .

Στο διαδικτυακό τόπο του *sourceforge.net* υπάρχουν χιλιάδες έργα ανοικτού λογισμικού και για κάθε ένα από αυτά υπάρχει ένας αρκετά μεγάλος όγκος πληροφοριών που έχει να κάνει με τις διάφορες εκδόσεις αυτών, πληροφορίες σχετικές με την εξέλιξη του κάθε έργου αλλά και άλλες λεπτομέρειες και γενικά

χαρακτηριστικά για κάθε έργο στα οποία υπάρχει εκτενής αναφορά στη συνέχεια του εγγράφου. Από όλα αυτά τα έργα αλλά και από όλο αυτό τον όγκο πληροφορίας το σύστημα μας έχει τη δυνατότητα να απομονώνει τη πληροφορία που ενδιαφέρει πραγματικά τον χρήστη του και να αποκόπτει όλα τα άλλα περιττά στοιχεία. Με αυτό τον τρόπο ο χρήστης δεν χάνεται μέσα στην χαοτική πληροφορία που παρέχει το *sourceforge.net* αλλά ασχολείται μόνο με τα έργα και τις πληροφορίες που τον ενδιαφέρουν πραγματικά. Στο σημείο αυτό πρέπει να διευκρινιστεί ότι σε καμία περίπτωση το σύστημα δεν προορίζεται να αντικαταστήσει το ίδιο το *sourceforge.net*, το οποίο είναι ένα πάρα πολύ καλά δομημένο και πλήρες αποθετήριο ανοικτού κώδικα, ο μοναδικός ρόλος του είναι απλώς να διευκολύνει το χρήστη του να βρίσκει ευκολότερα τις πληροφορίες που τον απασχολούν. Επίσης άλλος ένας ρόλος που καλείται να παίξει είναι να κωδικοποιήσει την πληροφορία αυτή και να την προβάλει με συστηματικό τρόπο.

1.4 ΑΠΑΙΤΗΣΕΙΣ ΣΥΣΤΗΜΑΤΟΣ

Οι απαιτήσεις του συστήματος διακρίνονται σε λειτουργικές και μη-λειτουργικές απαιτήσεις. Οι λειτουργικές (Functional Requirements) περιγράφουν το τι πρέπει να κάνει το σύστημα, δηλαδή καθορίζουν τις λειτουργίες και την συμπεριφορά του. Οι μη-λειτουργικές περιγράφουν ιδιότητες του συτήματος που συνήθως εκφράζονται βάσει χαρακτηριστικών της μορφής: απόδοση, χρηστικότητα, ασφάλεια νομιμότητα, κτλ.. Ένας ακόμα διαχωρισμός ο οποίος μπορεί να γίνει είναι ότι οι λειτουργικές απαιτήσεις εφαρμόζονται κατά το σχεδιαδμό του συτήματος ενώ οι μη-λειτουργικές απαιτήσεις εφαρμόζονται στην αρχιτεκτινική του συστήματος.

1.4.1 ΛΕΙΤΟΥΡΓΙΚΕΣ

Οι λειτουργικές απαιτήσεις του συστήματος είναι:

• Συλλογή του ονόματος του έργου
Το σύστημα ουσιαστικά ελέγχει αν το όνομα το οποίο έχει εισάγει ο χρήστης είναι έγκυρο και το εμφανίζει. Αν όχι εμφανίζει το κατάλληλο μήνυμα.

• Συλλογή των αρχείων του έργου
Το σύστημα συλλέγει όλα τα ονόματα των αρχείων τα οποία ανήκουν στο έργο και έχουν αναρτηθεί σε αυτό. Μία υπερκατηγορία των αρχείων είναι οι φάκελοι οι οποίοι περιέχουν αρχεία, αλλά συμπεριφέρονται και αντιμετωπίζονται από το σύστημα σαν αρχεία. Εδώ θα πρέπει να σημειωθεί ότι το σύστημα φτάνει μέχρι σε βάθος “3” από τον αρχικό φάκελο “Home” του κάθε έργου.

• Συλλογή του αριθμού λήψεων κάθε αρχείου
Το σύστημα συλλέγει για ένα συγκεκριμένο εύρος χρόνου, το οποίο είναι προκαθορισμένο, πόσες ήταν οι μηνιαίες λήψεις του κάθε αρχείου.

• Συλλογή των αναφορών tracker του έργου ανά μήνα
Το σύστημα συλλέγει για κάθε μήνα το πλήθος των αναφορών για κάθε μία από τις τέσσερις κατηγορίες tracker: σφάλματα (Bugs), αιτήματα προσθήκης νέων

χαρακτηριστικών (Feature Requests), ενημερώσεων (Patches) και αιτήματα υποστήριξης (Support Requests). Για κάθε κατηγορία γίνεται διαχωρισμός των αναφορών σε ανοιχτές, κλειστές, διαγραμμένες και εκκρεμείς.

• Απλή επικοινωνία με το χρήστη
Το σύστημα ζητάει απλά το όνομα του έργου που τον ενδιαφέρει και μόνο με αυτή την πληροφορία ολοκληρώνει όλη τη λειτουργικότητα του. Ο τρόπος παρουσίασης των αποτελεσμάτων γίνεται σε δύο ομάδες πινάκων. Η πρώτη ομάδα αποτελείται από ένα πίνακα ο οποίος παρουσιάζει τα ονόματα των αρχείων, τις ημερομηνίες και στην τις συνολικές λήψεις για κάθε ημερομηνία. Η δεύτερη ομάδα πινάκων αποτελείται από κανένα έως τέσσερις πίνακες στους οποίους παρουσιάζεται το πλήθος των αναφορών. Ένας πίνακας για κάθε τύπο αναφοράς όπως προαναφέρθηκαν (Bugs, Feature Requests, Patches, Support Requests).

• Ταυτοποίηση από το *sourceforge.net*
Το σύστημα θα πρέπει να εκτελεί όλες τις διεργασίες που έχουν να κάνουν με το *sourceforge* χωρίς να χρειάζεται να εισέλθει στη σελίδα ως ταυτοποιημένος χρήστης.

1.4.2 ΜΗ ΛΕΙΤΟΥΡΓΙΚΕΣ

• Γλώσσα προγραμματισμού
Το σύστημα θα πρέπει να υλοποιηθεί σε php 5.

• Βάση δεδομένων
Το σύστημα δεν χρησιμοποιεί κάποια βάση δεδομένων για να αποθηκεύσει τα αποτελέσματα, όλα τα αποτελέσματα εμφανίζονται στον χρήστη σε μια απλή σελίδα html.

• Λειτουργικό
Το σύστημα θα πρέπει να μπορεί να τρέξει σε Apache server οποιασδήποτε έκδοσης.

ΚΕΦΑΛΑΙΟ 2ο
ΕΞΕΛΙΞΗ ΛΟΓΙΣΜΙΚΟΥ

Σε αυτό το κεφάλαιο, παρουσιάζεται μια βιβλιογραφική αναφορά σχετικά με την εξέλιξη των έργων ανοιχτού λογισμικού, ως προς την κοινότητα που τα υποστηρίζει.

2.1 ΕΜΠΕΙΡΙΚΕΣ ΜΕΛΕΤΕΣ ΕΞΕΛΙΞΗΣ ΛΟΓΙΣΜΙΚΟΥ

Η εν λόγω έρευνα αποσκοπεί στην εμπειρική μελέτη της εξέλιξης του λογισμικού. Ως εκ τούτου, το πρώτο βήμα ήταν η ανασκόπηση της βιβλιογραφίας στην περιοχή εξέλιξης λογισμικού. Αρχικά παρατηρήθηκε, ότι υπάρχει μια σχετική έλλειψη εμπειρικών εργασιών στην περιοχή. Ωστόσο, παρουσιάζεται μια σειρά από άρθρα που δημοσιεύτηκαν πριν από την εργασία, τα οποία αναλύονται παρακάτω, με ιδιαίτερη έμφαση σε θέματα ερευνητικής μεθόδου, σύμφωνα με το στόχο αυτού του κεφαλαίου.

2.1.1 Belady and Lehman

Οι Lehman και Belady έχουν μελετήσει την πορεία ενός μεγάλου λειτουργικού συστήματος στο οποίο υπάρχουν συνεχείς διαδοχικές εκδόσεις. Θεωρούν ότι ο συνολικός αριθμός των υπομονάδων αυξάνεται γραμμικά σε σχέση με τον αριθμό έκδοσης, αλλά παράλληλα ότι ο αριθμός των υπομονάδων που επηρεάζονται αυξάνεται εκθετικά σε σχέση με τον αριθμό έκδοσης. Όλες οι διορθώσεις που γίνονται τείνουν να καταστρέψουν τη δομή και να διαταράξουν το σύστημα. Όλο και λιγότερες είναι οι προσπάθειες που γίνονται για την διόρθωση των αρχικών ελαττωμάτων σχεδιασμού και όλο και περισσότερα χρήματα ξοδεύονται για την διόρθωση ελαττωμάτων σε προηγούμενες διορθώσεις. Όσο περνάει ο καιρός, το σύστημα γίνεται όλο και λιγότερο καλά δομημένο.

Πιο συγκεκριμένα, στην εργασία τους, οι Lehman και Belady, στη δεκαετία του '70 εμπλέκουν 20 εκδόσεις του λειτουργικού OS/360, και ίσως αποτελεί τη πρώτη εμπειρική έρευνα επικεντρωμένη στην δυναμική συμπεριφορά ενός σχετικά μεγάλου και ώριμου (12 ετών) συστήματος. Οι συγγραφείς έκαναν μια σειρά παρατηρήσεων σχετικά με την αύξηση του μεγέθους και της πολυπλοκότητας του συστήματος, το οποίο τους οδήγησε να συμπεράνουν κάποιους νόμους της δυναμικής εξέλιξης του λογισμικού: συνεχής αλλαγή, αυξανόμενη πολυπλοκότητα, εξέλιξη προγράμματος και διατήρηση οργανωτικής σταθερότητας. Αυτοί οι νόμοι αναπτύχθηκαν περαιτέρω σε ένα έγγραφο που δημοσιεύθηκε το 1980. Σε αυτό το μοναδικό έργο, ο συγγραφέας (Lehman) αναλύει την έννοια των νόμων στο πλαίσιο των πραγματικών συστημάτων. Τα εμπειρικά στοιχεία που παρουσιάζονται

αφορούν ένα μικρό αριθμό από μεταγενέστερες εκδόσεις. Πρώτον, υποστηρίζεται, μέσω γραφικής παρουσίασης των δεδομένων, ότι υποστηρίζονται οι προαναφερθέντες νόμοι της εξέλιξης. Στη συνέχεια, μια λεπτομερής ανάλυση δείχνει πώς η γνώση που αποκτήθηκε μέσα από την ανάλυση των πρώτων 19 εκδόσεων του λογισμικού μπορούν να χρησιμοποιηθούν για το σχεδιασμό της 20ης έκδοσης. Το γεγονός αυτό δείχνει πώς μερικές αρχικές εκτιμήσεις ήταν πιθανό να είναι αισιόδοξες και κατά κάποιον τρόπο ανακριβής, μέχρι να διορθωθούν με την πρόσβαση στα ιστορικά δεδομένα, και είναι γίνουν χρήσιμες με την πρακτική εφαρμογή αυτού του τύπου της έρευνας.

2.1.2 Chong Hok Yuen

Η πρώτη συστηματική μελέτη των νόμων του Belady και του Lehman έγινε από έναν από τους μαθητές του Lehman, τον Chong Hok Yuen. Το έργο του παρουσιάστηκε σε μια σειρά τριών εμπειρικών μελετών, που δημοσιεύθηκαν το 1985, 1987 και το 1988. Αρχικά, μελετήθηκαν σε χρονική διαρκεί 19 μηνών δεδομένα για bugs (ελαττώματα) από ένα μεγάλο λειτουργικό σύστημα. Τα δεδομένα αναλύθηκαν σε τέσσερις διαφορετικούς μήνες στον 1ο, στον 7ο , στον 13ο και στον 19ο. Επτά εξαρτημένες μεταβλητές περιγράφονται, αλλά περιγράφονται μόνο δύο σύνολα αποτελεσμάτων, σχετικά με την τάξη προτεραιότητας (σοβαρότητας) και του χρόνου απόκρισης. Ο συγγραφέας αναφέρει μια σειρά από αποτελέσματα που αφορούν σε ανακάλυψη και διόρθωση ελαττωμάτων, συμπεριλαμβανομένων των παρατηρήσεων ότι οι ελλείψεις έχουν την τάση να ανακαλύπτονται σε γενεές, και τείνουν να συγκεντρώνονται λίγο μετά την ολοκλήρωση της έκδοσης. Ενδιαφέρον, παρουσιάζει ότι ο χρόνος που περνά δεν μπορεί να καθορίσει το πλήθος των ελαττωμάτων, όπως θα μπορούσε να αναμένεται, λόγω της αύξησης της πολυπλοκότητας του συστήματος. Ο συγγραφέας προτείνει ότι η έρευνά του μπορεί να χρησιμοποιηθεί για να ανιχνεύσει τη βελτίωση / επιδείνωση των καταστάσεων παρατηρώντας την τάση στο χρόνο απόκρισης ή στον αριθμό των εκκρεμών διορθώσεων και έτσι επιτρέπει στους διαχειριστές να αναλάβουν δράσεις για τη βελτίωση της διαδικασίας και της απόδοσης του συστήματος. Στη συνέχεια ο Yuen εκτελεί δοκιμές στους πέντε νόμους του Lehman σχετικά με τη δυναμική εξέλιξη. Επανεξετάζει τρία διαφορετικά συστήματα από τους Belady και Lehman, καθώς και διάφορα άλλα συστήματα, και κοιτάζει μια ποικιλία των εξαρτημένων μεταβλητών, συμπεριλαμβανομένου του αριθμού και του ποσοστού των μονάδων χειρισμού.

2.1.3 Tamai and Torimitsu

Οι Tetsuo Tamai και Yohsuke Torimitsu χρησιμοποίησαν μια έρευνα ερωτηματολογίου των ιαπωνικών οργανώσεων με σκοπό να εξετάσει την αντικατάσταση λογισμικού επιχειρηματικών εφαρμογών μέσα σε μια 5-ετή περίοδο. Αναφέρουν μια σειρά από περιγραφικά αποτελέσματα για 95 σημεία δεδομένων, συμπεριλαμβανομένου του γεγονότος ότι το λογισμικό μικρότερης κλίμακας τείνει να έχει μικρότερη διάρκεια ζωής, καθώς και ότι οι διοικητικές εφαρμογές τύπου

(π.χ., προσωπικό, λογιστική) τείνουν να έχουν μεγαλύτερη διάρκεια ζωής από ό, τι υποστηρίζουν επιχειρηματικές εφαρμογές τύπου (π.χ., υποστήριξη πωλήσεων, παραγωγής).

Ουσιαστικά δηλαδή υποστηρίζουν πως η διαδικασία εξέλιξης ενός συστήματος δεν τελειώνει με το "θάνατο" ενός μεμονωμένου συστήματος αλλά συνήθως συνεχίζει την εξέλιξη του από γενιά σε γενιά με το να αντικαθιστάται από νέα μεμονωμένα συστήματα.

2.1.4 Cook and Roesch

Οι Cook και Roesch εξέτασαν 10 προϊόντα με διάρκεια ζωής άνω των 18 μηνών σε πραγματικό χρόνο από μια γερμανική εταιρεία τηλεπικοινωνιών. Πρωταρχικός στόχος τους ήταν μια διερευνητική - παραγοντική ανάλυση των μετρικών λογισμικού για την πολυπλοκότητα. Οι συγγραφείς εξέτασαν την εξέλιξη του συστήματος στους 18 μήνες, και υποστήριξαν ότι βρήκαν ανταπόκριση στους νόμους της εξέλιξης λογισμικού, επικαλούμενοι τις συνεχείς αλλαγές, την αύξηση της πολυπλοκότητας, και το γεγονός ότι η συνολική αλλαγή δεν είναι ομοιόμορφη. Στο σύστημα αναπτύχθηκαν από 223 έως 359 λειτουργίες κατά τη διάρκεια της περιόδου που είχε μελετηθεί.

2.1.5 Gefen and Schneberger

Οι Gefen και Schneberger εξέτασαν δύο πρότυπα κατανομής των τροποποιήσεων συντήρησης λογισμικού για να προσδιοριστεί εάν η κατανομή συντήρησης λογισμικού είναι ομοιογενής. Αυτή η επιλογή έγινε γιατί η συντήρηση του λογισμικού είναι συνήθως και η πιο δαπανηρή διαδικασία η οποία σχετίζεται με την ανάπτυξη λογισμικού.

Οι συγγραφείς μελέτησαν Αναφορές Προβλημάτων Λογισμικού (Software Problem Reports ή SPRs) από ένα σύστημα 4ης γενιάς, συλλέγοντας μηνιαία στοιχεία. Αυτά τα SPRs χαρακτηρίζονται από τον τύπο τροποποίησης, καθώς και από τον αριθμό των νέων αιτήσεων - αναφορών. Επιπλέον, παρακολουθείται ο αριθμός των τροποποιήσεων που προκαλούνται από τις προηγούμενες τροποποιήσεις. Παρατήρησαν ότι το ποσοστό των τροποποιήσεων συντήρησης μειώνεται με την πάροδο του χρόνου στο σύνολό τους, αλλά όχι εάν θεωρηθούν ως μεμονωμένες φάσεις, τις οποίες χαρακτηρίζουν ως επέκταση, σταθεροποίηση και βελτίωση.

2.1.6 Victor Basili

Μια σχετικά πιο πρόσφατη μελέτη από τον Basili εξέτασε 25 εκδόσεις λογισμικού, του Goddard της NASA, συμπεριλαμβανομένων πάνω από 100 συστημάτων λογισμικού, συνολικού μεγέθους περίπου 4,5 εκατομμυρίων γραμμών κώδικα. Η εστίαση της μελέτης ήταν να χαρακτηρίσει τους τύπους των

δραστηριοτήτων συντήρησης και να εξετάσει τόσο τη συνολική προσπάθεια και τη κατανομή της προσπάθειας σε αυτά τα έργα συντήρησης. Η συλλογή δεδομένων διήρκεσε 18 μήνες. Η μελέτη εξέτασε τους τρεις τύπους συντήρησης (διορθωτική συντήρηση, προσαρμοστική και τελειοποιητική) και ένα σύνολο εργασιών συντήρησης για τον καθένα. Ο συγγραφέας βρήκε ότι η προσπάθεια διόρθωσης σφάλματος, συνήθως μικρού εύρους, απαιτεί σημαντική δραστηριότητα Ενώ σε αναβαθμίσεις απαιτείται αρκετά περισσότερος χρόνος για την επιθεώρηση και την πιστοποίηση. Η προσπάθεια για το σχεδιασμό και την κωδικοποίηση της μονάδα ελέγχου ήταν παρόμοια για τα δύο είδη. Στατιστική ανάλυση αυτών των διαφορών ήταν αναγκαστικά περιορισμένη.

2.1.7 Meir “Manny” Lehman

Πιο πρόσφατα, ο Lehman και οι συνεργάτες του έχουν ξεκινήσει μια νέα σειρά από έρευνες σχετικά με την εξέλιξη του λογισμικού. Σε πρώιμο άρθρο, όπου περιγράφουν μια εμπειρική ανάλυση των 8-χρονών ενός χρηματοπιστωτικού συστήματος συναλλαγών. Από το σύνολο των 100 γενεών (συμπεριλαμβανομένων των πολύ μικρών γενεών για συγκεκριμένους πελάτες), οι ερευνητές εξέτασαν 21 γενιές που εκτείνονται σε 5 χρόνια εξέλιξης. Για κάθε έκδοση σημειώθηκε το μέγεθος της όσον αφορά τον αριθμό των μονάδων και τον αριθμό των μονάδων που έχουν αλλάξει. Επιπλέον, διενεργήθηκε στατιστική μοντελοποίηση, με στόχο την ανάπτυξη ενός μοντέλου ανάπτυξης για το μέγεθος μονάδας ανά γενιά χρησιμοποιώντας ένα αντίστροφο τετράγωνο μοντέλο. Ένα από τα αποτελέσματα του μοντέλου αυτού ήταν ότι η παρατηρούμενη αύξηση του συστήματος εμφανίστηκε από την έκτη έκδοση. Είναι ενδιαφέρον, όπως στην περίπτωση προγενέστερης μελέτης, οι τελικές μερικές παρατηρήσεις θεωρούνται ακραίες τιμές στο γενικά ομαλό μοντέλο ανάπτυξης. Το άρθρο αυτό, επίσης, συνοψίζει και ενημερώσεις στους οκτώ νόμους της εξέλιξης του λογισμικού.

2.2 ΠΡΟΤΥΠΑ ΕΞΕΛΙΞΗΣ ΛΟΓΙΣΜΙΚΟΥ ΑΝΟΙΧΤΟΥ ΚΩΔΙΚΑ

Η ανάπτυξη του ελεύθερου λογισμικού και λογισμικού ανοιχτού κώδικα (ΕΛΛΑΚ) έχει εμφανιστεί και διαδοθεί σε όλο τον κόσμο της τεχνολογίας λογισμικού, κυρίως τα τελευταία χρόνια. Η ανάπτυξή του συμπίπτει με την εξάπλωση και τη χρήση του Internet και του World Wide Web. Υποστηρίζει την ευρεία πρόσβαση σε απομακρυσμένες πληροφορίες και την ικανότητα σε ομάδες ανθρώπων με κοινές ιδέες να μπορούν να επικοινωνούν μεταξύ τους.

2.2.1 Τύποι οντοτήτων για τη μελέτη της εξέλιξης ΕΛΛΑΚ

Οι τύποι των αντικειμένων που είναι κατάλληλοι για τη μελέτη εξέλιξης ενός λογισμικού έχουν εντοπιστεί στις μελέτες του Lehman και των συνεργατών του. Οι κύριοι τύποι των οντοτήτων λογισμικού είναι οι εκδόσεις, οι εφαρμογές, η διαδικασία ανάπτυξης και τα μοντέλα. Το καθένα από αυτά μπορεί να οριστεί ως εξής.

- ΕΛΛΑΚ Εκδόσεις

Γενικά, τα μεγάλα συστήματα συνεχίζουν να αυξάνονται με το χρόνο. Όπως αναφέρει ο 6ος νόμος του Lehman. Οι σταθερές και οι ασταθείς εκδόσεις των ΕΛΛΑΚ διανέμονται παγκοσμίως. Οι εκδόσεις alpha, beta, candidate, και οι σταθερές εκδόσεις είναι διαθέσιμες για τους χρήστες , ενώ οι ασταθείς εκδόσεις είναι για τους προγραμματιστές που συμβάλλουν ενεργά στις ενημερώσεις του λογισμικού. Οι πολλαπλές πλατφόρμες των εκδόσεων λογισμικού συγχρονίζονται και κατανέμονται την ίδια στιγμή, αν και μπορεί να διαφέρουν όταν προστίθενται νέες πλατφόρμες (παράλληλα). Η εξέλιξη τους γίνεται με μια μη παραδοσιακή διαδικασία μεταξύ των σταθερών εκδόσεων. Οι εκδόσεις επίσης μπορούν να ονομαστούν με σχήματα ιεραρχικής ονοματολογίας, μερικές φορές με τρία ή και τέσσερα επίπεδα ένθετης αρίθμησης για τις σταθερές και τις ασταθείς εκδόσεις που απευθύνονται σε διαφορετικό κοινό. Ωστόσο η πλειοψηφία των συστημάτων, κυρίως αυτών που είναι μικρού και μεσαίου μεγέθους, δεν συνεχίζουν να αυξάνονται ίσως επειδή το λογισμικό δεν χρησιμοποιείται εντατικά.

- ΕΛΛΑΚ Εφαρμογές

Τα προγράμματα ελευθέρου λογισμικού αναπτύσσονται από τα πρώτα ορίσματα που εισάγονται στην εφαρμογή ή τις απαιτούμενες αλλαγές που πρέπει να γίνουν, σε ένα υπάρχον σύστημα που διανέμεται και εγκαθίσταται ως ένα πρόγραμμα. Τα συστήματα μπορεί να είναι μικρού(<5,000 γραμμές κώδικα), μεσαίου(5,000 – 100,000), μεγάλου (100,000 – 1,000,000) ή πολύ μεγάλου μεγέθους(>1,000,000), όπου τα δύο τελευταία είναι λιγότερα σε αριθμό, αλλά και περισσότερο γνωστά. Τα περισσότερα μεγάλα προγράμματα ή συστήματα του ΕΛΛΑΚ μπορεί να υπάρχουν σε διαφορετικές εκδόσεις, που σχετίζονται όμως μεταξύ τους και προορίζονται για διαφορετικές πλατφόρμες εφαρμογών (π.χ. , MS Windows, Solaris, GNU/Linux, Mac OS X). Πολλά από αυτά δομούνται ως κατανεμημένα συστήματα, συστήματα που έχουν ρυθμιστεί για τη χρήση των scripts (π.χ. , χρησιμοποιώντας Perl, Python, Tcl), middleware ή ως modules, (π.χ. , Apache, Mozilla). Η μεγαλύτερη ποικιλία εφαρμογών ΕΛΛΑΚ έχει ερευνηθεί για τα πρότυπα εξέλιξης. Αυτά που εξετάστηκαν σε βάθος μέχρι στιγμής είναι ο πυρήνας του Linux, Debian Linux distributions3, Mono, Apache Web server, Mozilla Web browser, Berkeley DB, GNOME ένα σύστημα για την επιφάνεια εργασίας για τον τελικό χρήστη, PostgreSQL DBMS, καθώς και πολλά άλλα. Έχουν γίνει μελέτες για εφαρμογές ΕΛΛΑΚ για εκατοντάδες έως και πάνω από 40,000 συστήματα.

- Διαδικασία ανάπτυξης ΕΛΛΑΚ

Τα συστήματα ΕΛΛΑΚ αναπτύχθηκαν και συντηρήθηκαν ακολουθώντας κάποιες διαδικασίες ανάπτυξης. Ωστόσο δεν είναι ξεκάθαρο αν οι διαδικασίες αυτές, όπως αναφέρονται μπορούν να θωρηθούν ως μια μονολιθική διαδικασία, ή αν κάποιες συγκεκριμένες δραστηριότητες λογισμικού ακολουθούν σαφείς διαδικασίες ανάπτυξης που επίσης μπορούν να εξελιχθούν είτε ανεξάρτητα είτε

από κοινού. Μικρός αριθμός από πρόσφατες μελέτες παρατηρεί, περιγράφει και συγκρίνει τις διαδικασίες ανάπτυξης των ΕΛΛΑΚ με τις παραδοσιακές διαδικασίες έτσι ώστε να σημειωθούν οι διαφορές που εμφανίζονται στις δραστηριότητες και στην οργάνωση της διαδικασίας των ΕΛΛΑΚ. Επιπλέον οι δραστηριότητες αυτές έχουν τη δική τους διαδικασία που μπορεί να μην αντικατοπτρίζεται με αυτές που συμμετέχουν στην κυκλοφορία των συστημάτων κλειστού κώδικα. Επίσης οι δραστηριότητες που αφορούν τις διαδικασίες έκδοσης των έργων ανοικτού κώδικα μπορεί να ακολουθούν τη δική τους διαδικασία ανάπτυξης και μπορεί να είναι διαφορετική από αυτή που ακολουθούν τα συστήματα κλειστού λογισμικού.

- Μοντέλα ΕΛΛΑΚ

Τα υπάρχοντα μοντέλα διαδικασιών ανάπτυξης λογισμικού δεν λαμβάνονται υπόψη στις δραστηριότητες ανάπτυξης των ΕΛΛΑΚ. Οι διαδικασίες έκδοσης και ομαδοποίησης μπορεί μερικές φορές να διαφέρουν ή να είναι ίδιες, εξαρτάται από τα συστήματα εφαρμογών λογισμικού και ανάπτυξης (π.χ. , Web site για ανοικτό λογισμικό). Η διαδικασία ανάπτυξης του ΕΛΛΑΚ διαφέρει ριζικά από τη διαδικασία ανάπτυξης του κλειστού λογισμικού έτσι όπως έχει μελετηθεί στους νόμους ανάπτυξης λογισμικού. Οι νόμοι αυτοί εφαρμόζονται, σε μεγάλο βαθμό για τον υπολογισμό της ανάπτυξης του ΕΛΛΑΚ.

2.2.2 Πρότυπα για την εξέλιξη του λογισμικού ανοικτού κώδικα

Εδώ παρουσιάζονται κάποια παραδείγματα από μελέτες όπου εξέτασαν συστήματα ΕΛΛΑΚ και επικεντρώνονται κυρίως για το πώς τα αποτελέσματά τους μπορούν να συγκριθούν με εκείνα του Lehman και των συνεργατών του.

Οι Godfrey και Tu παρουσίασαν μια έρευνα για την ανάπτυξη του Linux κατά το διάστημα 1994-1999, και βρήκαν ότι ο ρυθμός ανάπτυξής του είναι παραπάνω από γραμμικός, όπως εμφανίζεται στα σχήματα 1 έως 3. Ο Schach αναφέρει τα αποτελέσματα της έρευνας για τον Linux Kernel για τις 96 εκδόσεις και δείχνει ότι η μονάδα σύζευξης (ή διασύνδεσης) αυξάνεται με εκθετικό (superlinear) ρυθμό. Τα δεδομένα εμφανίζονται στο σχήμα 4.

Οι Koch and Schneider μελέτησαν το GNOME ένα σύστημα για την επιφάνεια εργασίας για τον τελικό χρήστη, και τα δεδομένα τους δείχνουν ότι το μέγεθος του πηγαίου κώδικα αυξάνεται κατά τη διάρκεια των εκδόσεων με υπέρ-γραμμικό τρόπο καθώς ο αριθμός των προγραμματιστών που συμβάλλουν στον κώδικα του GNOME αυξάνεται. Τα δεδομένα από την έρευνά τους απεικονίζονται στο σχήμα 5.

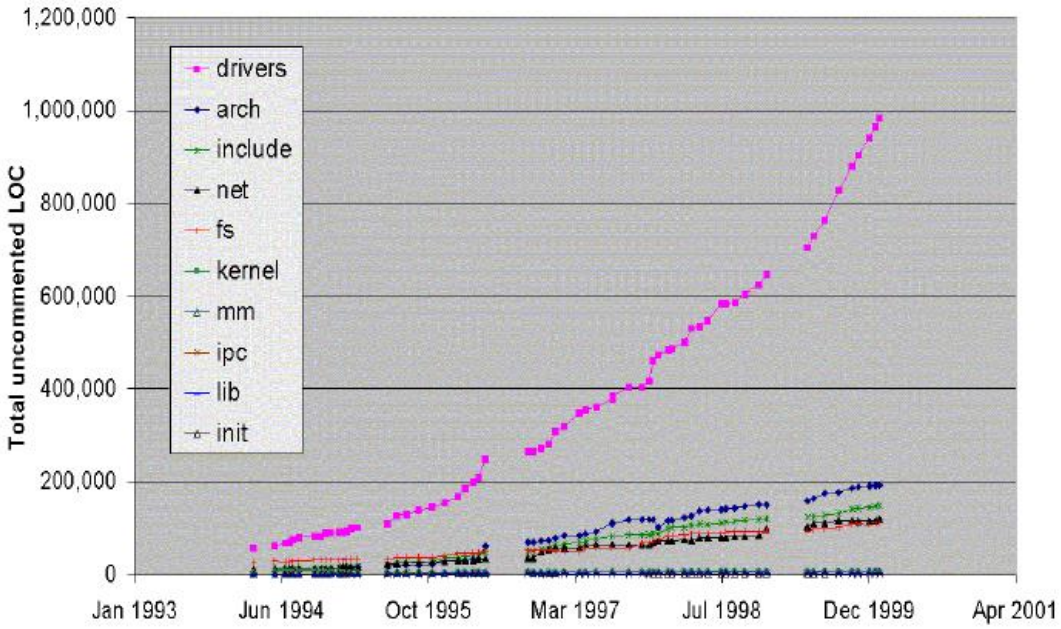


Figure 1. Data revealing the size and growth of major sub-systems in the Linux Kernel during 1994-1999 [Source: Godfrey and Tu 2000].

εικόνα 5 . ανάπτυξη του Linux 1994-1999

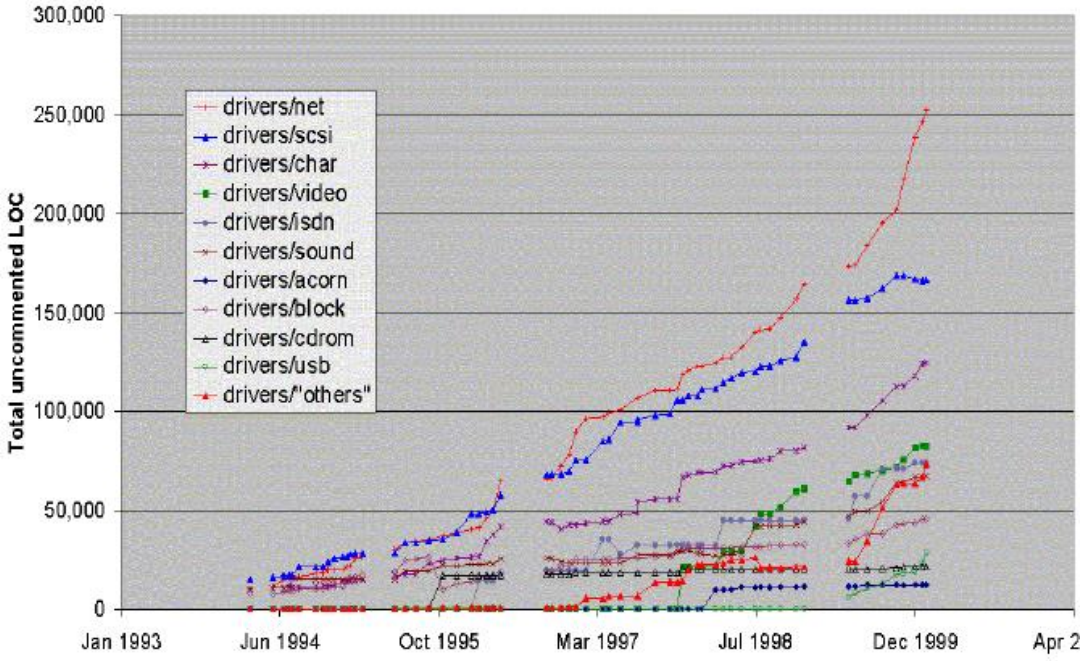


Figure 2. Data revealing the size and growth of device drivers in the Linux Kernel during 1994-1999 [Source: Godfrey and Tu 2000].

εικόνα 6 . ανάπτυξη του Linux 1994-1999

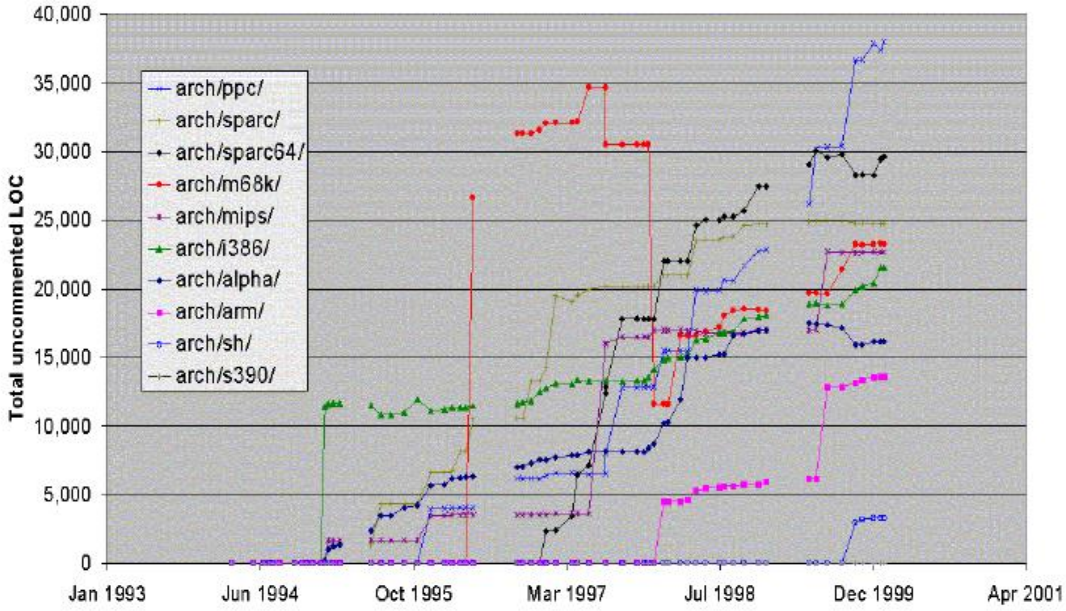


Figure 3. Data revealing the size and growth of the Linux Kernel for different computer platform architectures during 1994-1999 [Source: Godfrey and Tu 2000].

εικόνα 7 . ανάπτυξη του Linux 1994-1999

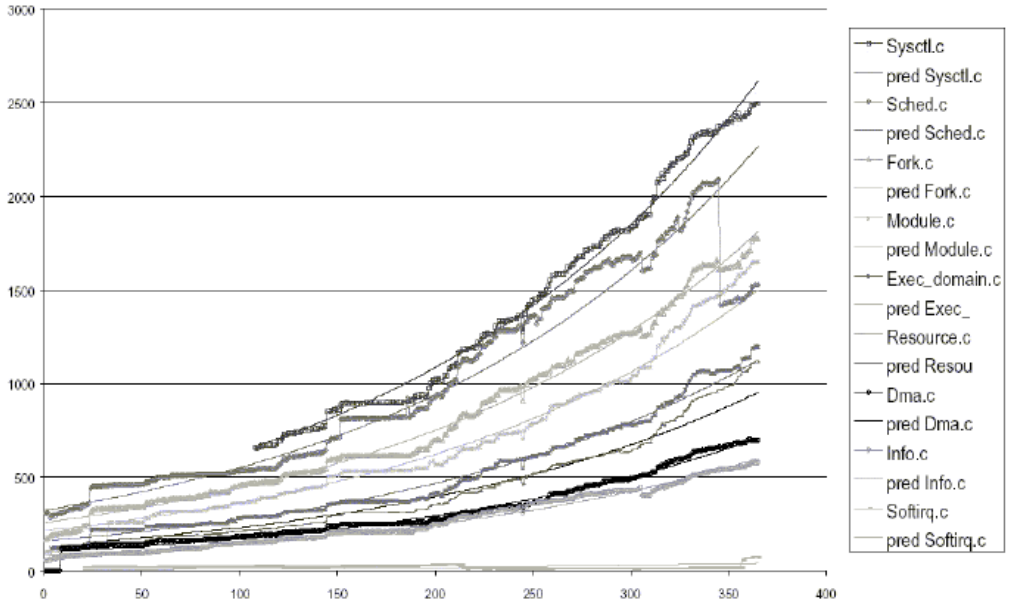


Figure 4. Measured (discrete points) versus predicted (smooth curves) of common coupling of source code modules in the Linux Kernel across releases [Source: Schach, Jin, *et al.*, 2002].

εικόνα 8 . αύξηση σύζευξης Linux

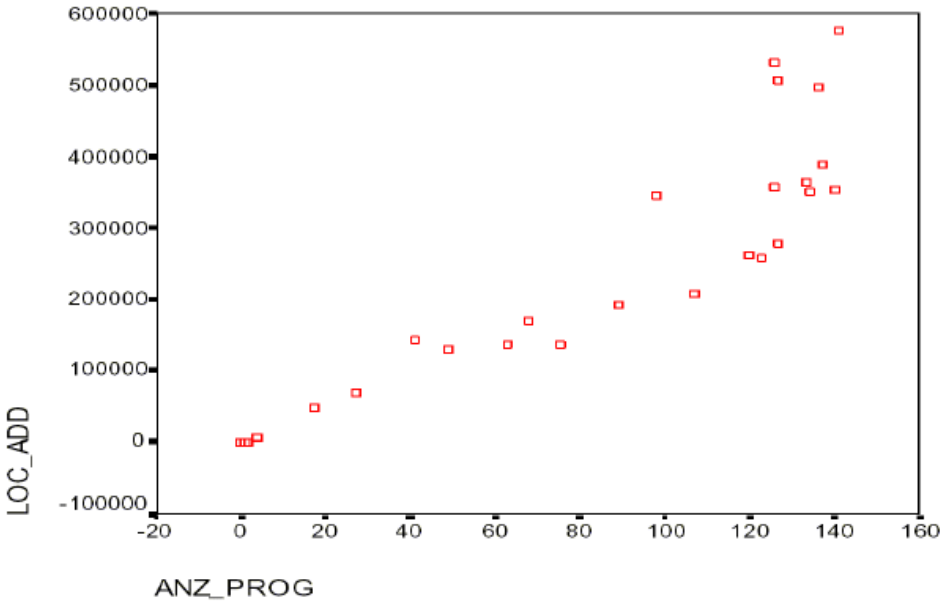


Figure 5. Growth of the lines of source code added as the number of software developers contributing code to the GNOME user interface grows
[Source: Koch and Schneider 2000].

εικόνα 9 . αύξηση κώδικα του GNOME

Οι Robles-Martinez, Gonzalez-Barahona, αναφέρουν στην μελέτη τους για το Mono (μια ΕΛΛΑΚ εφαρμογή των υπηρεσιών της Microsoft .NET, βιβλιοθήκες και διεπαφές), και οι μετρήσεις τους δείχνουν υπέρ γραμμικό ρυθμό ανάπτυξης στο μέγεθος του κώδικα και στον αριθμό των ενημερώσεων του κώδικα. Επίσης αναφέρουν ένα παρόμοιο πρότυπο ανάπτυξης στον αριθμό των ατόμων που συνεισφέρουν στον πηγαίο κώδικα του συστήματος Mono.

Σύμφωνα με τους Gonzalez - Barahona, Ortuno Perez στα μέσα του 2001, η διανομή του Debian GNU/Linux 2.2 έχει αυξηθεί σε περισσότερο από 55,000 γραμμές κώδικα, και από τότε έχει υπερβεί τις 100,000 στο Debian 3.0. Η O'Mahony παρουσιάζει κάποια δεδομένα από τη δική της μελέτη για το Debian Gnu/Linux που εκτείνεται από το 0.01 του 1993 ως το 3.0 στα τέλη του 2002, δείχνει ότι η το μεγέθους της διανομής αυξάνεται τα τελευταία 5 χρόνια. Τέλος, οι Gonzalez - Barahona, Lopez, and Robles παρέχουν δεδομένα για την ανάπτυξη κοινότητας του έργου Apache και του αριθμού των μονάδων, αποκαλύπτοντας και πάλι την άνω της γραμμικής ανάπτυξης στο διάστημα (1999-2004).

Σε αντίθεση με τους Godfrey and Tu που βρήκαν γραμμική ανάπτυξη στα συστήματα Fetchmail, X-Windows, and Gcc (the GNU compiler collection), και υπό-γραμμική ανάπτυξη στο Pine (email client), το οποίο διαφέρει από το προηγούμενο σύνολο συστημάτων.

Γιατί υπάρχει τόσο μεγάλος ρυθμός ανάπτυξης σε κάποια συστήματα ΕΛΛΑΚ (όπως είναι το Linux Kernel, Vim, GNOME, Mono, Debian GNU/Linux distribution, και το έργο Apache), ενώ σε άλλα όχι; Οι Godfrey and Tu στην περίπτωση του πυρήνα Linux αναφέρουν ότι:

a) Ένα μεγάλο μέρος του πηγαίου κώδικα σχετίζεται με προγράμματα οδήγησης συσκευών

b) μεγάλο μέρος του κώδικα προορίζεται για διαφορετικές πλατφόρμες, όπως φαίνεται στο σχήμα 3, και
c) η συνεισφορά στον κώδικα είναι ανοιχτή σε όποιον κάνει την απαιτούμενη προσπάθεια. Επίσης παρατηρούν ότι
d) τα configurations είναι συγκεκριμένα για τη πλατφόρμα hardware ή την αρχιτεκτονική και χρησιμοποιούν το 15% του συνολικού πηγαίου κώδικα.

Είναι πιθανό αυτές οι υποθέσεις να ισχύουν και για το GNOME, Vim, Mono και το Apache project, γιατί τα configurations του πηγαίου κώδικα είναι συγκεκριμένα σε διαφορετικά λειτουργικά συστήματα. (Linux, BSD, Windows, or Mac OS/X). Ωστόσο δεν είναι ξεκάθαρο γιατί ισχύει ή όχι για τα συστήματα Fetchmail, X-Windows, Gcc and Pine. Ίσως επειδή τα τελευταία συστήματα είναι μεγαλύτερα σε ηλικία και μπορεί να αναπτύσσονται σε ένα παλαιότερο τεχνολογικό σύστημα. Ενώ οι Cook, Ji, and Harrison στη μελέτη σύγκρισης για τα κλειστά συστήματα λογισμικού Logica FW, και το ΕΛΛΑΚ σύστημα DB system, βρήκαν ότι η ανάπτυξη τους κατά τη διάρκεια των εκδόσεων, δεν είναι ομοιόμορφα κατανεμημένη.

Οι Nakakoji, Yamamoto αναφέρουν συμπεράσματα από μια συγκριτική μελέτη περίπτωσης τεσσάρων συστημάτων ΕΛΛΑΚ, Linux Kernel, Postgres DBMS, GNUWingnut, και Juna 3D graphics library. Τα στοιχεία που προκύπτουν δείχνουν ότι τα συστήματα αυτά παρουσιάζουν διαφορετικά πρότυπα εξέλιξης όπως φαίνεται στο σχήμα 6. Δείχνει επίσης ότι είναι απαραίτητο να κατανοήσουμε τα πρότυπα ηλικίας και αρχιτεκτονικής των υποσυστημάτων και των μονάδων μεταξύ των εκδόσεων του λογισμικού, είτε σε ανοιχτά είτε σε κλειστά συστήματα λογισμικού, για να κατανοήσουμε καλύτερα αν το σύστημα εξελίσσεται. Αυτή η παρατήρηση προκύπτει από άλλες μελέτες.

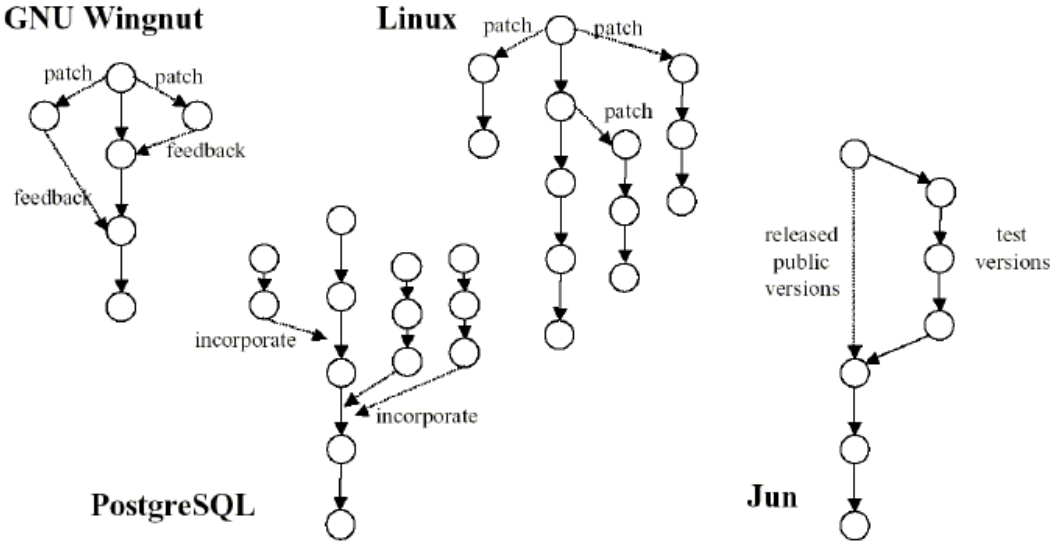


Figure 6. Patterns of software system evolution forking and joining across releases (nodes in each graph) for four different F/OSS systems
[Source: Nakakoji, Yamamoto, *et al.*, 2002]

εικόνα 10 . διαφορετικά πρότυπα εξέλιξης

Οι Hunt και Johnson αποκάλυψαν την κατανομή του Pareto στον βαθμό συμμετοχής των developers στα έργα του ΕΛΛΑΚ, από ένα δείγμα πληθυσμού >30,000 έργων που βρέθηκε στο *sourceforge.net*. Τα αποτελέσματά τους δείχνουν ότι η πλειοψηφία των έργων Ελεύθερου λογισμικού έχει μόνο ένα developer, ενώ μικρό ποσοστό έχει μεγαλύτερη ομάδα συμμετοχής.

Οι Madey, Freeh, και Tynan σε μια μελέτη παρόμοια με εκείνη του Hunt και Johnson, διαπίστωσαν ότι η a power law κατανομή χαρακτηρίζει το μέγεθος των έργων ΕΛΛΑΚ για πληθυσμό 40,000 έργων που άντλησαν από το *sourceforge.net*.

Οι Hars και Ou ανέφεραν μια παρόμοια τάση, και βρήκαν σύμφωνα με τη δική τους έρευνα ότι περισσότερο από το 60% των developers του ΕΛΛΑΚ συμμετέχουν σε άλλα 2-10 έργα ανάπτυξης.

Επίσης οι Capilupppi, Lago, and Morisio άντλησαν δεδομένα από το *sourceforge.net*, χρησιμοποίησαν ένα δείγμα από 400 έργα ΕΛΛΑΚ. Βρήκαν ότι η πλειοψηφία των συστημάτων στο δείγμα τους είναι μικρού ή μεσαίου μεγέθους, και μόνο ένα μικρό ποσοστό είναι μεγάλου μεγέθους. Μόνο τα μεγάλα συστήματα ΕΛΛΑΚ τείνουν να έχουν ομάδες ανάπτυξης με περισσότερους από έναν προγραμματιστές. Τα αποτελέσματα που προκύπτουν μπορούν επίσης να συγκριθούν με τη μελέτη των Tamai και Torimitsu, και καταλήγουν ότι τα μικρά συστήματα έχουν μικρότερη διάρκεια ζωής, σε σύγκριση με τα μεγαλύτερα ΕΛΛΑΚ συστήματα. Αυτό δείχνει ότι τα αποτελέσματα των ερευνών που αφορούν μεγάλα συστήματα, δεν είναι αντιπροσωπευτικά για την πλειοψηφία των έργων ΕΛΛΑΚ. Οι Di Penta, Neteler παρέχουν δεδομένα από μια μελέτη περίπτωσης που εστιάζει στην επανακωδικοποίηση μεγάλων εφαρμογών ΕΛΛΑΚ, το GRASS που είναι ένα γεωγραφικό σύστημα πληροφοριών, και λειτουργεί σε ένα μικρό υπολογιστή χειρός. Η προσπάθειά τους είχε ως στόχο τη μείωση των επαναλήψεων κώδικα, την διαγραφή των μη χρησιμοποιούμενων αρχείων καθώς και την αναδόμηση των βιβλιοθηκών και την αναδιοργάνωσή τους σε κοινές βιβλιοθήκες.

Οι τεχνικές αναβάθμισης συστημάτων λογισμικού ειδικά σε επίπεδο αρχιτεκτονικής δεν έχουν αναφερθεί στους νόμους εξέλιξης λογισμικού. Για παράδειγμα η ελαχιστοποίηση και η αναδόμηση μειώνει το μέγεθος μιας εφαρμογής, και οδηγεί στη βελτίωση της ποιότητας του κώδικα. Ο Scacchi αναφέρει τα αποτελέσματα από μια μελέτη για το έργο GNUenterprise, και βρίσκει ότι η εφαρμογή E-Commerce παρουσιάζει ανάπτυξη καθώς μεγαλώνει και συγχωνεύεται με άλλα ανεξάρτητα συστήματα ΕΛΛΑΚ, που κανένα από αυτά δεν σχεδιάστηκε για συγχώνευση. Χαρακτηρίζει αυτή την ασυνεχή ανάπτυξης της λειτουργικότητας και του μεγέθους ΕΛΛΑΚ συστημάτων ως architectural bricolage.

Οι Mockus, Fielding, Herbsleb παρουσίασαν μια συγκριτική μελέτη περίπτωσης του εξυπηρετητή (server) Apache (<100,000 γραμμές κώδικα) και Mozilla Web browser (2,000,000+ γραμμές κώδικα), φαίνεται ότι είναι ευκολότερο να διατηρηθεί η ποιότητα των χαρακτηριστικών του συστήματος για ΕΛΛΑΚ κατά τη διάρκεια των εκδόσεων, σε σύγκριση με τα εμπορικά συστήματα τηλεπικοινωνιών κλειστού κώδικα. Επίσης απέδειξαν ότι τα μεγάλα έργα ΕΛΛΑΚ πρέπει να έχουν 10-15 προγραμματιστές στην ομάδα ανάπτυξης για να διατηρηθεί η εξέλιξή τους. Αυτός ο δείκτης δείχνει το μέγεθος της ομάδας που πετυχαίνουν μεγάλο ρυθμό ανάπτυξης. Ωστόσο δεν είναι ξεκάθαρο εάν και πόσο καιρό μπορεί να διατηρηθεί η ανάπτυξη και γιατί ο αριθμός των developers μεταβάλλεται με το χρόνο.

Ο Scacchi και οι συνεργάτες του παρουσιάζουν τα αποτελέσματα που προκύπτουν από συγκριτικές μελέτες περίπτωσης των έργων ΕΛΛΑΚ, σε

διαφορετικές κοινότητες. Βρήκαν ότι οι απαιτήσεις και οι διαδικασίες έκδοσης διαφέρουν από τα αναμενόμενα χαρακτηριστικά στην τεχνολογία λογισμικού. Επίσης αναφέρουν ότι η εξέλιξη των ΕΛΛΑΚ εξαρτάται από την εξέλιξη της κοινότητας προγραμματιστών, την κοινότητα υποστήριξης λογισμικού.

Οι Nakakoji, Yamamoto αναφέρουν ότι τα τέσσερα συστήματα που ερευνούν, εξελίσσονται παράλληλα με τις κοινότητες των προγραμματιστών που τα διατηρούν. Τέλος οι Gonzalez - Barahona, Lopez, και Robles παρουσιάζουν λεπτομερές σύνολο δεδομένων που οραματίζεται την ανάπτυξη της κοινότητας των προγραμματιστών σε διάστημα 5 χρόνων που αντιστοιχεί στην αύξηση του αριθμού των μονάδων που ενσωματώνονται στο έργο Apache.

Οι Von Hippel and Katz αναφέρουν τα αποτελέσματα στις μελέτες τους και δείχνουν ότι κάποιοι τελικοί χρήστες στα έργα ΕΛΛΑΚ γίνονται προγραμματιστές, κι ότι οι περισσότεροι προγραμματιστές είναι τελικοί χρήστες του συστήματος που αναπτύσσεται, επιτρέποντας έτσι την ταυτόχρονη εξέλιξη του συστήματος και τη κοινότητα χρηστών-developer. Αυτό προκύπτει και από άλλες μελέτες. Τέλος, άλλοι όπως οι Like Hars και Ou, Madey, Freeh και Tynan βρήκαν ότι κάποιοι developers τους οποίους χαρακτηρίζουν ως developers – συνδετικούς κρίκους, συμμετέχουν σε πολλά έργα. Αυτοί οι developers δημιουργούν κοινωνικά δίκτυα και διασυνδέουν έργα ΕΛΛΑΚ για να μοιράζονται τον πηγαίο κώδικα.

Ωστόσο, σε όλες τις μελέτες ξεκινώντας από αυτή των Mockus, Fielding και Herbsleb, δεν υπάρχουν αντίστοιχες παρατηρήσεις ή νόμοι που να αναφέρονται για την εξέλιξη των συστημάτων κλειστού λογισμικού. Προφανώς αυτό δεν σημαίνει ότι πρότυπα δεν εμφανίζονται στην εξέλιξη του κλειστού λογισμικού. Αντίθετα, προκύπτει ότι άλλες μεταβλητές και πρότυπα που δεν υπολογίζονται στις προηγούμενες μελέτες, μπορούν να είναι σημαντικοί παράγοντες και να συμβάλλουν στην εξέλιξη του λογισμικού. Σε μια μελέτη που συγκρίνει ανοικτά και κλειστά συστήματα λογισμικού οι Paulson, Succi, και Eberlein βρήκαν ότι η συνολική εξέλιξη και των δύο συστημάτων είναι σύμφωνη με τους νόμους Εξέλιξης Λογισμικού, για τα συστήματα που εξέτασαν. Στην εξέλιξη των μεγάλων συστημάτων ΕΛΛΑΚ, είναι απαραίτητο η κοινότητα των χρηστών, προγραμματιστών, και χρηστών-προγραμματιστών να συνδέονται. Τα παλαιότερα ΕΛΛΑΚ συστήματα που μπορεί να εμφανίστηκαν πριν το ΕΛΛΑΚ γνωρίσει ευρεία αναγνώριση, μπορεί να έχουν χαμηλό ποσοστό συν-εξέλιξης.

ΚΕΦΑΛΑΙΟ 3^{Ο}
ΣΧΕΔΙΑΣΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

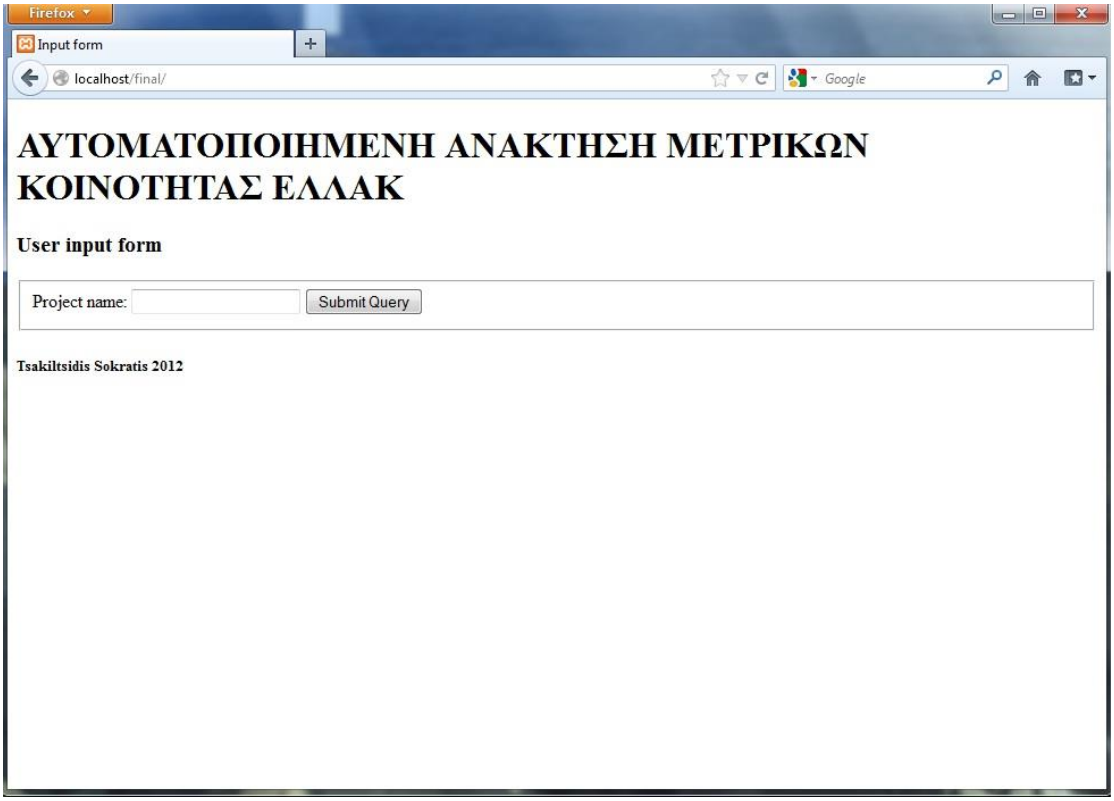
Στο κεφάλαιο αυτό αναλύεται η αρχιτεκτονική και η σχεδίαση του συστήματος καθώς και του κάθε υποσυστήματος που παίρνει μέρος στη διαδικασία της συλλογής πληροφοριών από το *sourceforge.net*.

3.1 ΣΥΝΟΛΙΚΗ ΕΠΙΣΚΟΠΗΣΗ ΕΡΓΟΥ

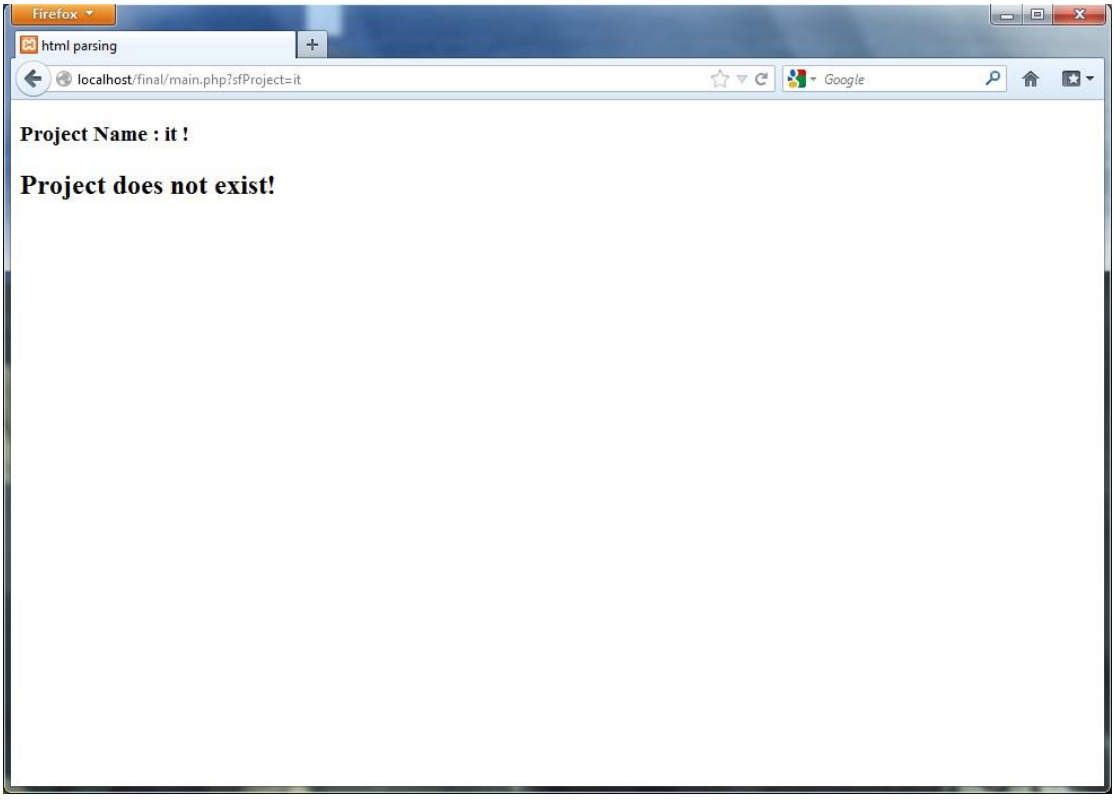
Το σύστημα αποτελείται βασικά από δύο υποσυστήματα, μία εφαρμογή που φροντίζει να παίρνει την πληροφορία των κινήσεων όσο αφορά τις λήψεις των αρχείων και από την εφαρμογή που φροντίζει να παίρνει την πληροφορία των κινήσεων όσο αφορά τις τέσσερις κατηγορίες αναφορών που γίνονται από τους χρήστες. Η λειτουργία των δύο τμημάτων του έργου με σκοπό την εκτέλεση της βασικής λειτουργίας είναι σχεδόν ανεξάρτητη. Παρ' όλα αυτά θα γίνει καλύτερα κατανοητή αν εξετάσουμε τον τρόπο με τον οποίο θα έτρεχε την εφαρμογή ο χρήστης.

Αρχικά λοιπόν ο χρήστης ανοίγει κάποιον περιηγητή πληκτρολογεί το url του συστήματος στη γραμμή διευθύνσεων και μεταβαίνει στη σελίδα του έργου (εικόνα 8). Εκεί ο χρήστης βλέπει μία απλή φόρμα στην οποία του ζητείτε να πληκτρολογήσει το όνομα του project. Αυτή η φόρμα και είναι ενσωματωμένη σε μια απλή σελίδα html η οποία έχει σαν στόχο να αποστείλει μέσω της φόρμας την επιλογή που έχει πληκτρολογήσει ο χρήστης. Με αυτό τον τρόπο ο χρήστης πυροδοτεί αρχικά τον έλεγχο για την ύπαρξη ή όχι του έργου στον ιστότοπο του *sourceforge.net*. Αν η πληροφορία λοιπόν υπάρχει στη βάση τότε εκτελούνται τα δύο υποσυστήματα τα οποία έχουν σαν στόχο να αντλήσουν την πληροφορία και να την προβάλουν στον χρήστη. Σε περίπτωση που η πληροφορία για το συγκεκριμένο έργο δεν υπάρχει στη βάση τότε το σύστημα εμφανίζει το κατάλληλο μήνυμα και τερματίζει την λειτουργία του (εικόνα 9).

Αμέσως μόλις γίνει από τον χρήστη μια έγκυρη εισαγωγή το σύστημα καλείται να εκτελέσει τις λειτουργίες που απαιτούνται για να εμφανίσει στο χρήστη τις πληροφορίες που του προσφέρονται από αυτό. Παρακάτω παρουσιάζονται στιγμιότυπα της λειτουργίας του συστήματος.



εικόνα 11 . Φόρμα εισαγωγής



εικόνα 12 . Μη έγκυρο όνομα έργου

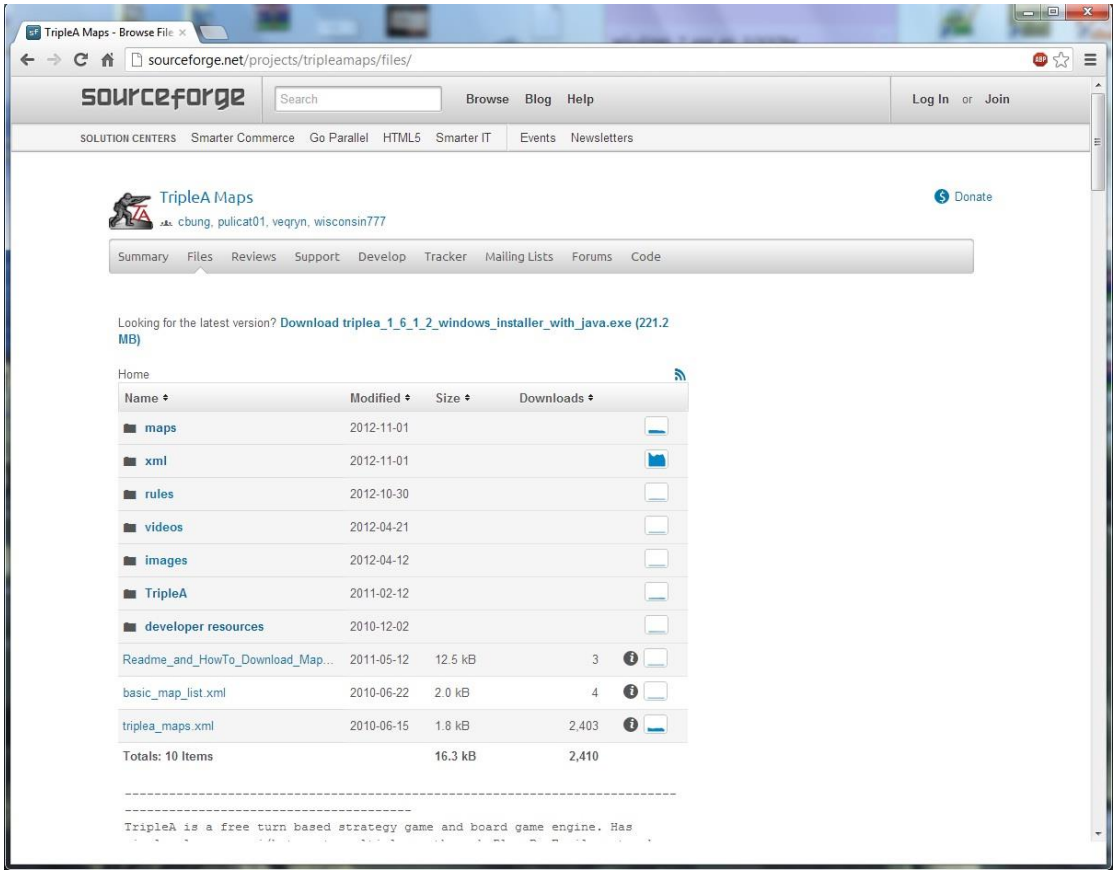
Στη συνέχεια του εγγράφου παρουσιάζεται αναλυτικότερα η δομή των δύο υποσυστημάτων καθώς και ο τρόπος με τον οποίο λειτουργούν και παρουσιάζουν τα αποτελέσματα τους.

3.2 ΜΕΤΡΙΚΕΣ ΛΗΨΕΩΝ

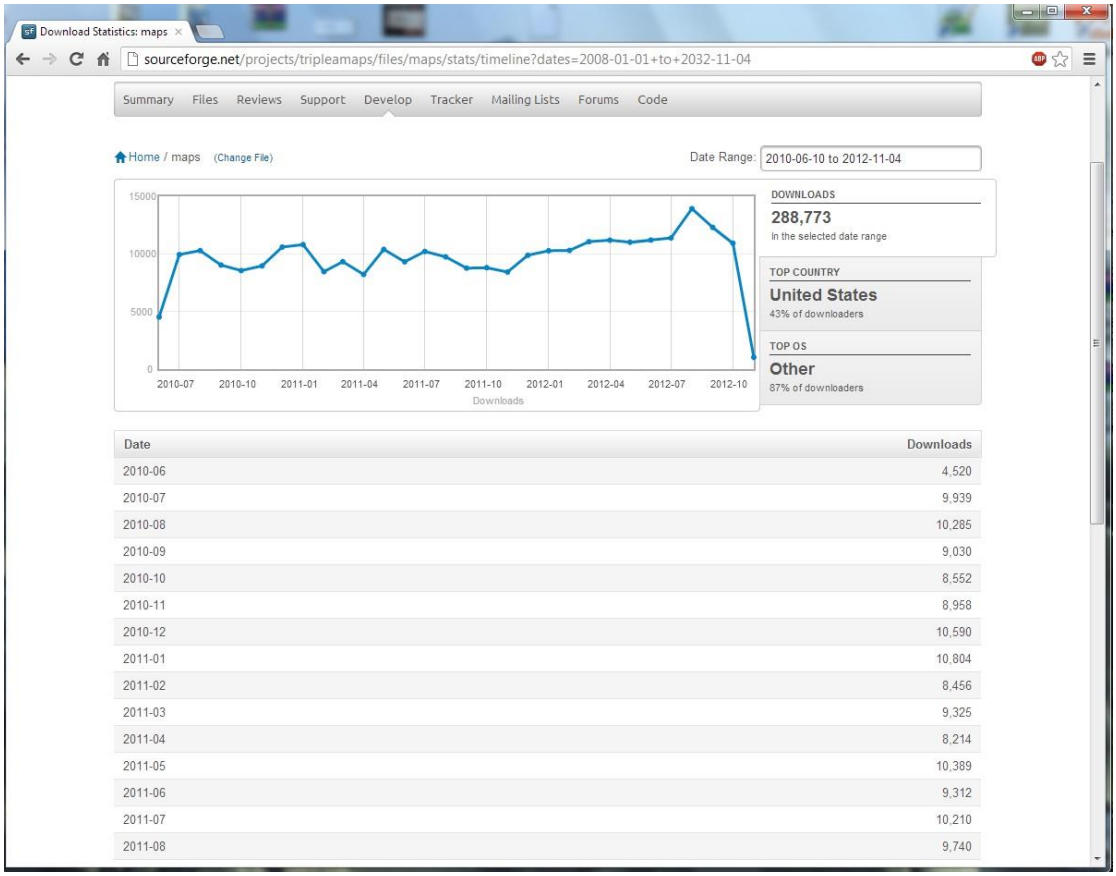
Το πρώτο υποσύστημα ασχολείται με την ανάκτηση της πληροφορίας των λήψεων κάθε αρχείου και φακέλου του έργου που έχει επιλεγεί. Σε αυτό το σημείο θα μελετήσουμε πιο συγκεκριμένα τα σημεία που εμπλέκονται στη λειτουργία αυτή αλλά και τον τρόπο με τον οποίο αντλούνται αυτές οι πληροφορίες.

Μόλις ο χρήστης εισάγει το όνομα του έργου για το οποίο θέλει να συλλέξει πληροφορίες, το σύστημα μέσω της φόρμας αποστέλλει την επιλογή του χρήστη στην κλάση *main* του συστήματος. Ή πιο σωστά η κλάση main αντλεί την είσοδο του χρήστη από την φόρμα της αρχικής html σελίδας.

Εκεί η είσοδος του χρήστη ενθυλακώνεται σε ένα url το οποίο κάνει τη χρήση του ευκολότερη. Έπειτα δημιουργείται, με παράμετρο αυτό το url, ένα αντικείμενο τύπου *project* το οποίο αναλαμβάνει αρχικά να ελέγξει αν το έργο υπάρχει στο *sourcefoge.net* και μετά να δημιουργήσει ένα πίνακα που θα περιλαμβάνει όλους τους φακέλους και τα αρχεία του έργου (εικόνα 10). Η μορφή στην οποία αποθηκεύονται αυτές οι πληροφορίες είναι επίσης ως url, με τις κατάλληλες προσθήκες έτσι ώστε η χρήση τους μετέπειτα να είναι ευκολότερη. Αφού έχουν συλλεχθεί τα ονόματα των αρχείων και των φακέλων απομένει για κάθε ένα από αυτά, μέσα στο εύρος των ημερομηνιών που είναι προκαθορισμένες από το σύστημα, να συλλεχτούν τα αντίστοιχα πλήθη των λήψεων, τα οποία παρουσιάζονται από το *sourceforge.net* ανά μήνα (εικόνα 11). Η αντιστοίχιση ημερομηνιών και πλήθους λήψεων γίνεται στην κλάση *downloads* ενώ η αντιστοίχιση αρχείων και ημερομηνιών γίνεται στην κλάση *files*. Τέλος σε αυτό το πρώτο υποσύστημα γίνεται και η εμφάνιση των αποτελεσμάτων του σε έναν απλό δυσδιάστατο πίνακα (εικόνα 14).



εικόνα 13 . φάκελοι και αρχεία έργου



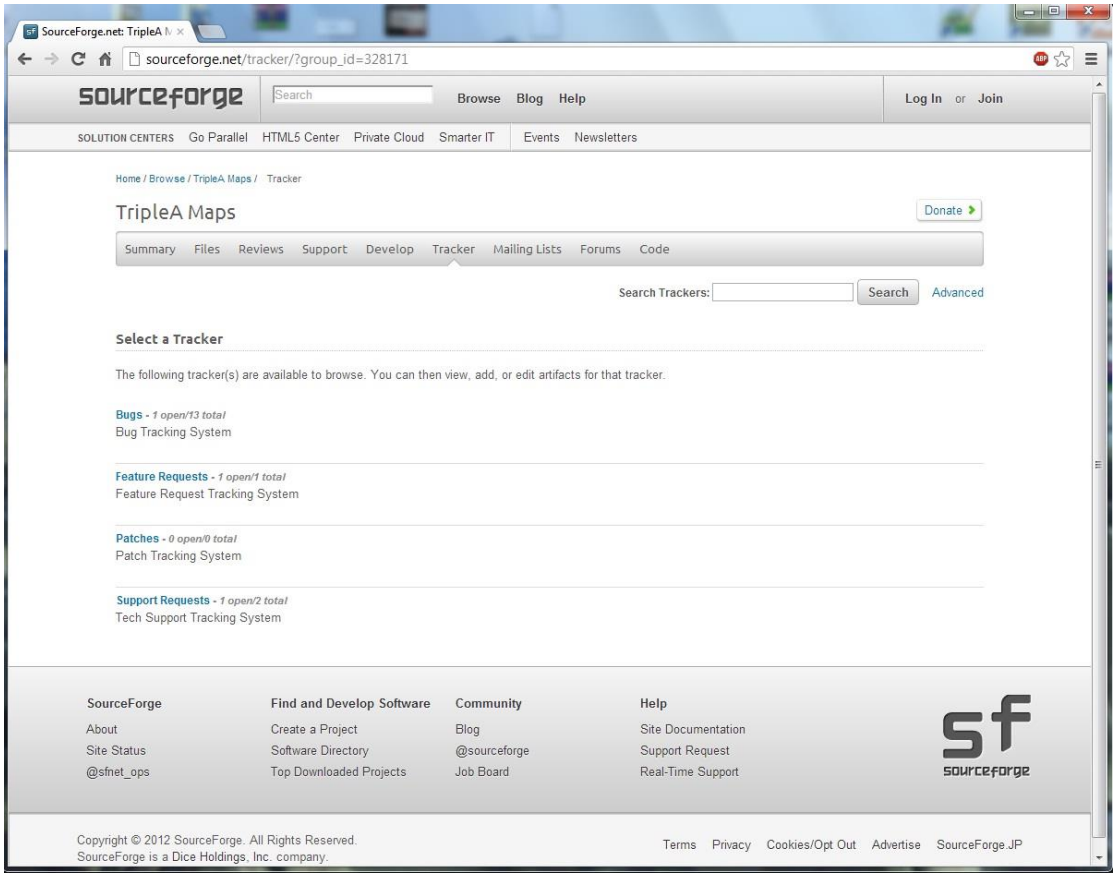
εικόνα 14 . μετρικές λήψεων (*sourceforge.net*)

3.3 ΜΕΤΡΙΚΕΣ ΑΝΑΦΟΡΩΝ

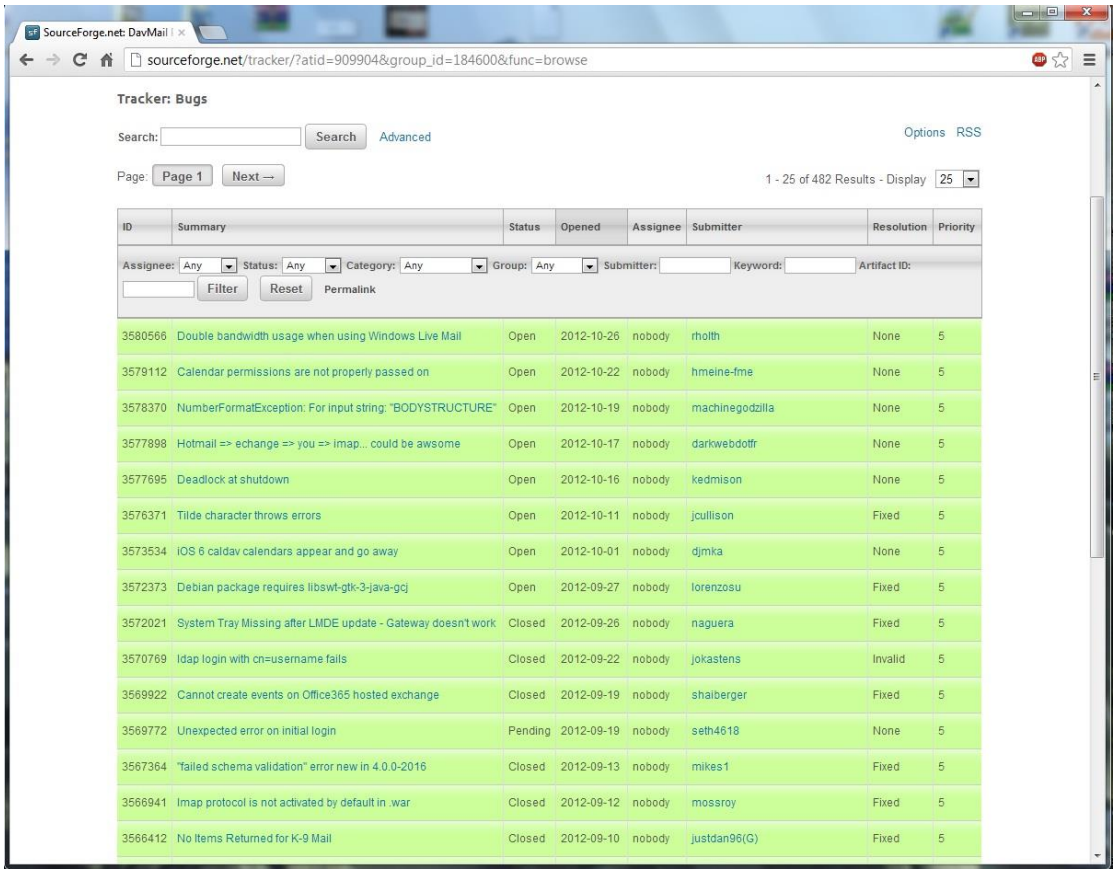
Το δεύτερο υποσύστημα ασχολείται με την ανάκτηση της πληροφορίας των αναφορών κάθε αρχείου και φακέλου του έργου που έχει επιλεγεί.

Μετά την ολοκλήρωση του πρώτου υποσυστήματος το οποίο ασχολείται με την άντληση των μετρικών λήψεων καλούνται από την κλάση main του συστήματος μία προς μία οι τέσσερις κλάσεις οι οποίες λειτουργούν ανεξάρτητα από το προηγούμενο κομμάτι και ταυτόχρονα ανεξάρτητα η καθεμία από την άλλη. Η σειρά με την οποία καλούνται είναι η εξής: Bugs, Feature Requests, Patches και Support Requests. Η σειρά αυτή είναι και η σειρά με την οποία παρουσιάζονται και στο *sourceforge.net* . Ο τρόπος λειτουργίας τους είναι παρόμοιος καθώς η πληροφορία για το κάθε ένα από αυτά παρουσιάζεται από το *sourceforge.net* σε ίδιο interface.

Βασιζόμενη και πάλι, η κάθε μία από τις τέσσερις κλάσεις, στο αρχικό url που έχει περαστεί ως παράμετρος από την κλάση main ξεκινάει την άντληση των απαιτούμενων πληροφοριών. Το url αρχικά διαμορφώνεται με στόχο να δείχνει στη σελίδα που παρουσιάζει ποιες αναφορές είναι διαθέσιμες για το έργο το οποίο έχει ο χρήστης εισάγει (εικόνα 12). Αυτό είναι και το σημείο όπου γίνεται ο έλεγχος αν το έργο έχει ή όχι αναφορές. Ένας ακόμα έλεγχος που γίνεται είναι αν υπάρχει η κάθε μια κατηγορία αναφοράς, γιατί υπάρχει πιθανότητα το έργο να μην διαθέτει αναφορές και για τις τέσσερις κατηγορίες. Αυτός ο έλεγχος γίνεται σε κάθε κλάση ξεχωριστά την στιγμή την οποία αυτή εκτελείται. Μετά το τέλος των ελέγχων και αν δεν έχει διακοπεί από αυτούς η λειτουργία του συστήματος, σε περίπτωση δηλαδή μη ύπαρξης αναφορών το σύστημα αναλύει το περιεχόμενο του διαμορφωμένου url που αναφέραμε και συλλέγει ζευγάρια κωδικών τα οποία θα οδηγήσουν το σύστημα στις επιθυμητές αναφορές. Το κάθε ζευγάρι κωδικών αποτελείται από δύο νούμερα, το πρώτο είναι το group_id και το δεύτερο το atid ή tracker id. Η κάθε μία από τις κατηγορίες Bugs, Feature Requests, Patches και Support Requests έχει ένα μοναδικό ζευγάρι κωδικών το οποίο αναφέρεται σε συγκεκριμένο έργο. Με αυτή την πληροφορία πλέον αποθηκευμένη στο σύστημα μπορεί να γίνει αντιστοίχηση, καταμέτρηση και διαχωρισμός των αναφορών που έχουν γίνει (εικόνα 13). Κάθε φορά ξεχωριστά για κάθε κατηγορία αναφοράς αλλά και ξεχωριστά για την κατάσταση της αναφοράς, δηλαδή ανοιχτή, κλειστή, εκκρεμής και διαγραμμένη, το σύστημα μετράει τις αναφορές που έχουν γίνει κατά περιόδους, διαχωρισμένες ανά μήνα. Το αποτέλεσμα είναι ένας πίνακας για κάθε κατηγορία αναφοράς ο οποίος δημιουργείται μέσα στην κλάση της κάθε μίας (εικόνα 15).



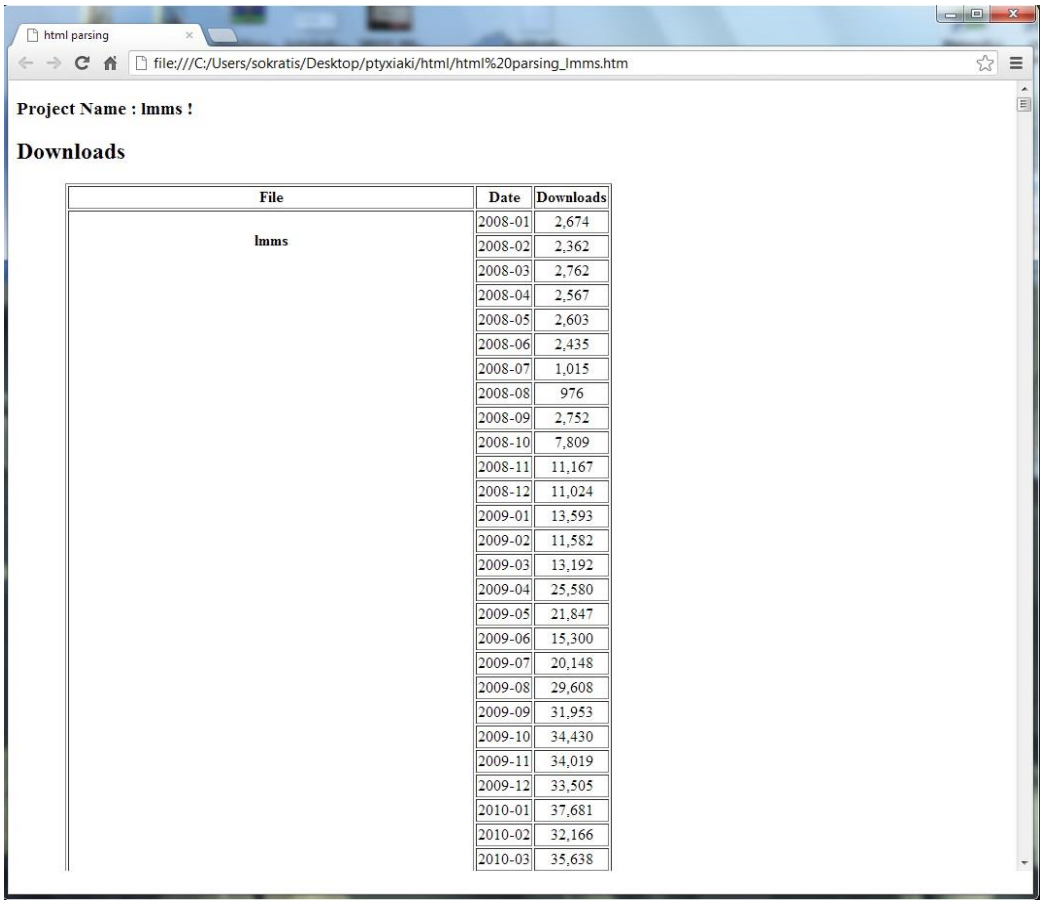
εικόνα 15 . κατηγορίες αναφορών



εικόνα 16 . αναφορές Bugs

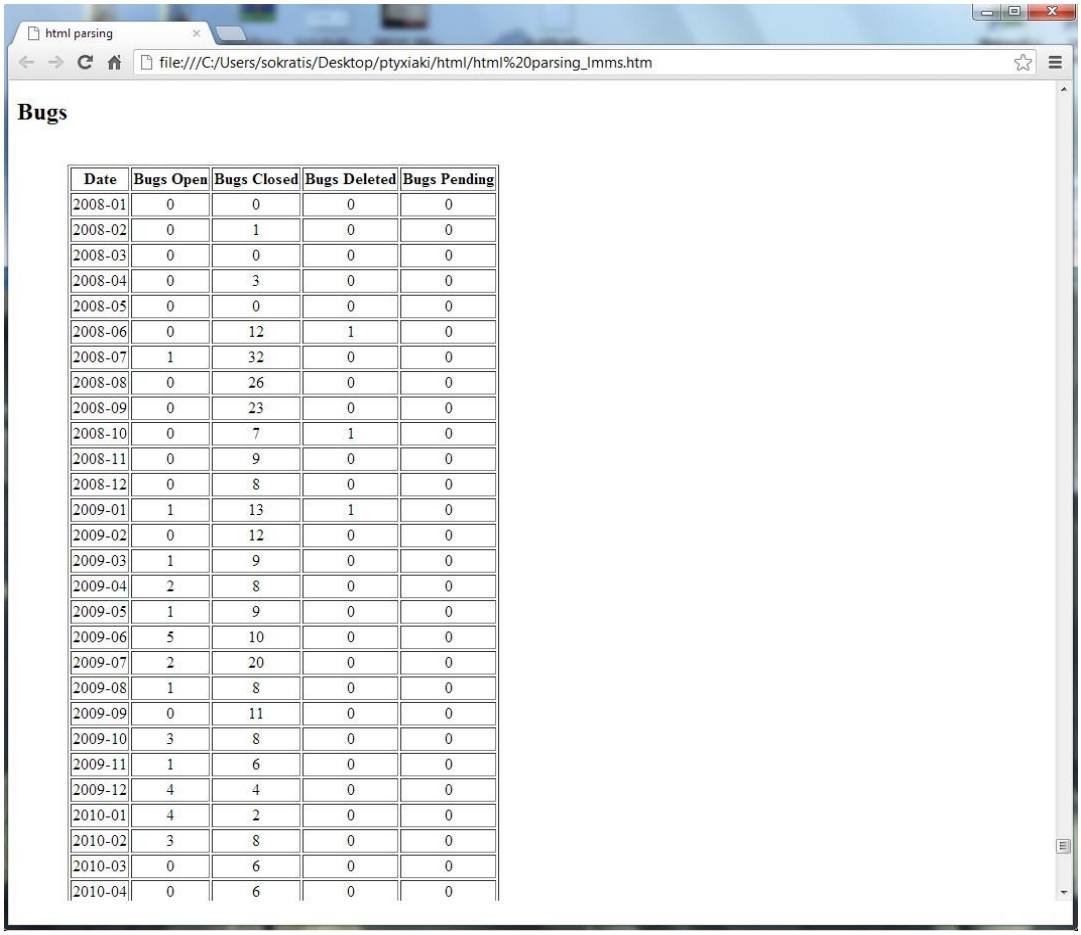
3.4 ΠΑΡΟΥΣΙΑΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ

Ο τρόπος με τον οποίο παρουσιάζονται στον χρήστη τα αποτελέσματα είναι όπως προαναφέρθηκε μέσω απλών πινάκων. Το *sourceforge.net* έχει την ιδιαιτερότητα να αποθηκεύει τα στοιχεία των μετρικών σε σύνολα τα οποία ταξινομούνται ανά μήνα. Για αυτό το λόγο και η παρουσίαση τους από το σύστημα γίνεται με αυτό τον τρόπο. Αναλυτικότερα, υπάρχει ολοκληρωμένο αρχείο αποτελεσμάτων στο τέλος του εγγράφου στο παράρτημα Β.



File	Date	Downloads
lmms	2008-01	2,674
	2008-02	2,362
	2008-03	2,762
	2008-04	2,567
	2008-05	2,603
	2008-06	2,435
	2008-07	1,015
	2008-08	976
	2008-09	2,752
	2008-10	7,809
	2008-11	11,167
	2008-12	11,024
	2009-01	13,593
	2009-02	11,582
	2009-03	13,192
	2009-04	25,580
	2009-05	21,847
	2009-06	15,300
	2009-07	20,148
	2009-08	29,608
	2009-09	31,953
	2009-10	34,430
	2009-11	34,019
	2009-12	33,505
	2010-01	37,681
	2010-02	32,166
	2010-03	35,638

εικόνα 17 . αποτελέσματα λήψεων



Bugs

Date	Bugs Open	Bugs Closed	Bugs Deleted	Bugs Pending
2008-01	0	0	0	0
2008-02	0	1	0	0
2008-03	0	0	0	0
2008-04	0	3	0	0
2008-05	0	0	0	0
2008-06	0	12	1	0
2008-07	1	32	0	0
2008-08	0	26	0	0
2008-09	0	23	0	0
2008-10	0	7	1	0
2008-11	0	9	0	0
2008-12	0	8	0	0
2009-01	1	13	1	0
2009-02	0	12	0	0
2009-03	1	9	0	0
2009-04	2	8	0	0
2009-05	1	9	0	0
2009-06	5	10	0	0
2009-07	2	20	0	0
2009-08	1	8	0	0
2009-09	0	11	0	0
2009-10	3	8	0	0
2009-11	1	6	0	0
2009-12	4	4	0	0
2010-01	4	2	0	0
2010-02	3	8	0	0
2010-03	0	6	0	0
2010-04	0	6	0	0

Εικόνα 18 . αποτελέσματα Bugs

ΚΕΦΑΛΑΙΟ 4Ο
ΥΛΟΠΟΙΗΣΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

Στο κεφάλαιο αυτό θα γίνει μία αναλυτική περιγραφή της υλοποίησης του συστήματος. Θα δοθεί μεγαλύτερη προσοχή σε σημαντικά σημεία του κώδικα που αντλεί πληροφορίες.

4.1 ΔΟΜΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

Το σύστημα έχει σαν αφετηρία μία σελίδα html, την index.html,από όπου ο χρήστης σε μία φόρμα εισάγει το όνομα του έργου από το οποίο θέλει να ανακτήσει τις πληροφορίες που του προσφέρει το σύστημα. Η σελίδα αυτή έχει σαν σκοπό την εύκολη χρήση της και το πέρασμα της παραμέτρου που έχει εισάγει ο χρήστης στην πρώτη κλάση του συστήματος.

4.1.1 ΚΥΡΙΑ ΚΛΑΣΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

Μόλις υποβληθεί η αρχική φόρμα από τον χρήστη, ουσιαστικά πυροδοτείται η έναρξη της λειτουργίας του συστήματος από την κύρια κλάση του, η οποία βρίσκεται στο αρχείο main.php και αντλεί από την αρχική φόρμα την είσοδο του χρήστη.

Η βασική λειτουργία αυτής της κλάσης είναι αρχικά η δημιουργία του url με την χρήση της παραμέτρου από την φόρμα και έπειτα η κλήση όλων των κλάσεων οι οποίες θα αναλάβουν την βαθύτερη ανάκτηση των μετρικών. Όλες οι κλάσεις οι οποίες καλούνται από εδώ παίρνουν σαν όρισμα το urlπου έχει δημιουργηθεί. Η σειρά με την οποία καλούνται από εδώ οι κλάσεις είναι η σειρά με την οποία παρουσιάζονται στη συνέχεια.

4.1.2 ΚΛΑΣΗ PROJECT

Η πρώτη κλάση η οποία καλείται είναι αυτή η οποία βρίσκεται στο αρχείο project.php. Η κλάση αυτή αφού κληθεί και αρχικοποιήσει τις μεταβλητές της, εκτελεί μία συνάρτηση που περιέχει, η οποία θα αποθηκεύσει σε ένα πίνακα σε μορφή url όλους τους φακέλους και τα αρχεία του έργου. Το url το οποίο αποθηκεύεται, δείχνει στη σελίδα του φακέλου ή αρχείου η οποία περιέχει τις αναφορές τις οποίες θέλουμε να αντλήσουμε.

Το όνομα της συνάρτησης είναι “listLinks” και δέχεται σαν παράμετρο το αρχικό url και μια μεταβλητή η οποία θα επιστραφεί στην κλάση main και βοηθάει στον έλεγχο για την ύπαρξη του έργου που έχει εισάγει ο χρήστης. Αυτό βέβαια σημαίνει πως ο κώδικας μέχρι αυτό το σημείο θα εκτελεστεί ακόμα και αν η είσοδος του χρήστη είναι λανθασμένη ή το έργο δεν υπάρχει στο *sourceforge.net* . Η

συνάρτηση listLinks έχοντας σαν βάση την θέση files του έργου που έχει εισάγει ο χρήστης ξεκινάει να αναλύει το html αρχείο ελέγχοντας αν τα πεδία-σύνδεσμοι τα οποία εμφανίζονται είναι φάκελοι ή αρχεία του έργου. Ο τρόπος με τον οποίο εισχωρεί σε βάθος είναι κάνοντας έναν απλό έλεγχο αν το πεδίο το οποίο συναντά κάθε φορά είναι φάκελος ή αρχείο. Σε περίπτωση που είναι φάκελος, η συνάρτηση αποθηκεύει στον πίνακα τον σύνδεσμο του φακέλου ο οποίος οδηγεί στις αναφορές λήψεων του και μετά κατευθύνεται μέσα στον φάκελο. Εάν συναντήσει αρχείο τότε αποθηκεύει πάλι στον πίνακα τον σύνδεσμο του αρχείου που οδηγεί στις αναφορές λήψεων και συνεχίζει την αναζήτηση πεδίων. Με αυτό τον τρόπο επιτυγχάνεται η αποθήκευση όλων των συνδέσμων, φακέλων και αρχείων, του έργου που οδηγούν στις αναφορές λήψεων του. Αξίζει να σημειωθεί πως αυτό είναι και το σημείο στο οποίο ορίζεται το εύρος των ημερομηνιών στο οποίο θα λάβουμε τις αναφορές. Αυτό γίνεται προσθέτοντας σε κάθε νέο πεδίο του πίνακα, δηλαδή σε κάθε σύνδεσμο, τον τρόπο με τον οποίο το *sourceforge.net* καθορίζει το εύρος των ημερομηνιών για τις οποίες ενδιαφέρεται ο χρήστης. Για παράδειγμα, στον σύνδεσμο ο οποίος οδηγεί στις αναφορές λήψεων του φακέλου “User Manual” του έργου “lmms” ο οποίος είναι ο :

`sourceforge.net/projects/lmms/files/UserManual/stats/timeline`

προστίθεται στο τέλος του το εύρος ημερομηνιών, το οποίο είναι ένα σταθερό String :

`/?dates=2008-01-01+to+2020-01-01`

το σύνολο που προκύπτει θα μας δώσει τις αναφορές ανά μήνα από τον Ιανουάριο του 2008, ή τον πρώτο μήνα του έργου αν αυτός είναι μεταγενέστερος, μέχρι και τον τρέχοντα μήνα. Ο λόγος για τον οποίο εισάγουμε το έτος 2020 είναι ακριβώς για να το αγνοήσει το *sourceforge.net* αλλά παράλληλα να μας κατευθύνει μόνο του στην σημερινή ημερομηνία. Αυτή είναι μια ιδιομορφία το *sourceforge.net* η οποία στο συγκεκριμένο σημείο μας βοηθάει, γλιτώνοντας μας από περιττό έλεγχο ημερομηνιών.

Μετά το τέλος της λειτουργίας της listLinks και αφού έχει επιστρέψει τον πίνακα με τους συνδέσμους, η κλάση project για κάθε ένα από τα στοιχεία του πίνακα δημιουργεί και ένα αντικείμενο τύπου file, περνώντας σαν παράμετρο, ένα κάθε φορά, τα στοιχεία του πίνακα. Τέλος στην κλάση project υπάρχει και η συνάρτηση printDownloads η οποία καλείται από την main με σκοπό την εμφάνιση στον χρήστη του πίνακα με τις αναφορές λήψεων.

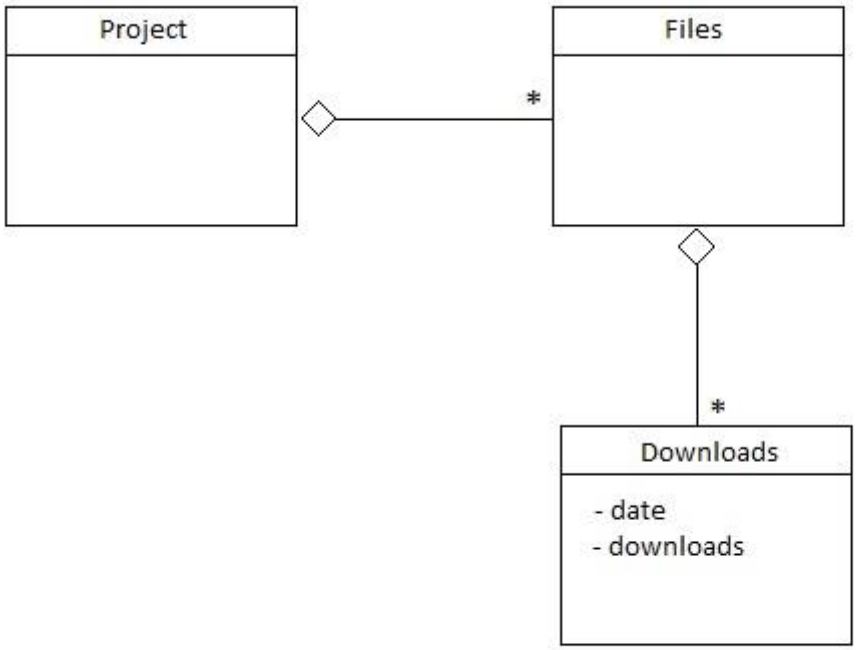
4.1.2.1 ΚΛΑΣΗ FILE

Η κλάση file, η οποία περιέχεται στο αρχείο file.php, μόλις κληθεί από την κλάση project κατά την δημιουργία ενός τέτοιου αντικειμένου, αρχικοποιεί τις μεταβλητές της και δέχεται σαν παράμετρο όπως είπαμε έναν σύνδεσμο ο οποίος δείχνει την σελίδα html που περιέχει τα στοιχεία τα οποία θέλουμε να αντλήσουμε. Ο σκοπός της κλάσης είναι με βάση αυτό το σύνδεσμο να αναλύσει αυτό το αρχείο html και να αποθηκεύσει αντιστοιχημένα την ημερομηνία (μήνα και έτος), μαζί με τον αριθμό λήψεων για τον συγκεκριμένο μήνα. Μόλις ένα τέτοιο ζευγάρι συμπληρωθεί, δημιουργείται ένα αντικείμενο τύπου downloads με παράμετρο αυτό το ζευγάρι τιμών.

Η κλάση αυτή περιλαμβάνει επίσης δύο συναρτήσεις get, οι οποίες επιστρέφουν το όνομα του αρχείου “getName” και το πλήθος των ζευγαριών για το συγκεκριμένο αρχείο ή φάκελο “getNumberOfDates”. Τέλος στην κλάση file υπάρχει και η συνάρτηση printDownloads η οποία καλείται από την project με σκοπό την εμφάνιση στον χρήστη του πίνακα με τις αναφορές λήψεων.

4.1.2.1 ΚΛΑΣΗ DOWNLOADS

Η μοναδική λειτουργία της κλάσης downloads η οποία περιέχεται στο αρχείο downloads.php είναι η αντιστοίχηση των μεταβλητών που της αποστέλλονται κατά τη δημιουργία ενός αντικειμένου downloads στην κλάση file. Επίσης υπάρχει και η συνάρτηση printDownloads η οποία καλείται από την file με σκοπό την εμφάνιση στον χρήστη του πίνακα με τις αναφορές λήψεων.



εικόνα 19 . διάγραμμα κλάσεων του υποσυστήματος αναφορών λήψεων

4.1.3 TRACKER

Αφού ολοκληρωθεί η εμφάνιση του πίνακα με τις αναφορές λήψεων, καλούνται από την κλάση main μία προς μία, οι τέσσερις κλάσεις οι οποίες αποσκοπούν στην αλίευση των αναφορών των χρηστών για σφάλματα - Bugs, αιτήματα προσθήκης νέων χαρακτηριστικών - Feature Requests, ενημερώσεων - Patches και αιτήματα υποστήριξης - Support Requests. Επειδή το *sourceforge.net*

παρουσιάζει τις πληροφορίες για αυτές τις κατηγορίες σε διαφορετικές σελίδες, όπως προαναφέρθηκε, αλλά με τον ίδιο τρόπο, η λειτουργία των τεσσάρων κλάσεων είναι όμοια. Γι αυτό το λόγο θα περιγραφεί η λειτουργία και η δομή μόνο μίας από αυτές.

Μόλις λοιπόν κληθεί η κλάση δημιουργώντας ένα αντικείμενο του τύπου της από την κλάση main, η κλάση αρχικοποιεί τις μεταβλητές της. Η έναρξη των λειτουργιών της ξεκινάει και πάλι με βάση το αρχικό url. Από την σελίδα αυτού του url γίνεται μία αναζήτηση στο html αρχείο για να αντληθεί ένας νέος σύνδεσμος ο οποίος δείχνει στο μενού πλοήγησης ανάμεσα στις τέσσερις κατηγορίες αναφορών (εικόνα 12). Από αυτό το νέο σύνδεσμο και το html αρχείο του, αντλούνται και αποθηκεύονται σε ένα πίνακα τα ζευγάρια κωδικών group_id και atid ή trackerid, τα οποία με τη σειρά τους θα μπορέσουν να μας οδηγήσουν στη κάθε κατηγορία αναφορών ξεχωριστά κάθε φορά.

Στην συνέχεια και αφού υπάρχει πλέον διαθέσιμη η πηγή από την οποία θα αντλήσουμε τις πληροφορίες μας, οδηγούμαστε εκεί κάνοντας παράλληλα και τον έλεγχο αν υπάρχει η κατηγορία αναφορών την οποία ερευνούμε. Αξίζει να σημειωθεί ότι σε ένα έργο δεν είναι απαραίτητο να υπάρχουν και οι τέσσερις κατηγορίες αναφορών. Σε αυτό το σημείο υπάρχει διαθέσιμη η πληροφορία σε ένα σύνολο σελίδων, στις οποίες παρουσιάζονται όλες οι αναφορές με την ημερομηνία υποβολής τους και την κατάσταση στην οποία βρίσκονται. Αυτό που κάνει το σύστημα εδώ είναι να δημιουργεί τέσσερις πίνακες, έναν για κάθε κατάσταση αναφοράς. Για παράδειγμα στην κλάσηbugs δημιουργούνται οι πίνακες:

```
$openBugs = array();
$closedBugs = array();
$deletedBugs = array();
$pendingBugs = array();
```

Τα πεδία των πινάκων αυτών θα περιέχουν ημερομηνίες στις οποίες έγινε καταχώρηση η τροποποίηση κάποιας αναφοράς στην κοινότητα του *sourceforge.net*. Το σύστημα ελέγχει όλες τις καταχωρήσεις που έχουν γίνει μεταβαίνοντας σε όλες τις σελίδες που υπάρχουν για την συγκεκριμένη κατηγορία αναφορών. Ο τρόπος με τον οποίο γίνεται αυτό είναι αυξάνοντας την μεταβλητή “offset” στο url της θέσης από όπου ξεκινάει η αναζήτηση. Το σύστημα σταματάει την αύξηση του όταν πλέον μετά τον κατάλληλο έλεγχο αναγνωρίζει ότι δεν υπάρχει επόμενη σελίδα.

http://sourceforge.net/tracker/...&**offset=25**&group_id=105168&atid=640434...

Έχοντας πλέον τους τέσσερις πίνακες με τις ημερομηνίες στις οποίες έχουν διαπιστωθεί αναφορές το σύστημα κάνει μία ταξινόμηση τους σε εύρος μηνός. Ακριβώς όπως παρουσιάζονται οι αναφορές λήψεων.

Τέλος, σε κάθε κατηγορία ξεχωριστά γίνεται και η εμφάνιση του πίνακα σε μορφή παρόμοια του πίνακα που έχει ήδη εμφανιστεί από τις αναφορές λήψεων (εικόνα 15). Στη θέση του πίνακα, θα εμφανιστεί και το κατάλληλο μήνυμα σε περίπτωση που η συγκεκριμένη κατηγορία δεν έχει εγγραφές για το έργο το οποίο αναλύεται.

4.2 ΠΑΡΑΔΟΧΕΣ ΓΙΑ ΤΗΝ ΥΛΟΠΟΙΗΣΗ ΤΟΥ ΕΡΓΟΥ

Αξίζει να σημειωθεί ότι οι μετρικές αναφορών δεν είναι πάντα απαραίτητο να συμβαδίζουν με την έκδοση στην οποία αντιστοιχούν ημερολογιακά. Αυτό συμβαίνει γιατί ο χρήστης της κοινότητας δεν έχει την δυνατότητα κατά τη διάρκεια της δήλωσης μιας αναφοράς σφάλματος να προσδιορίσει για ποιά έκδοση του έργου την κάνει. Αυτό που γίνεται είναι απλά η καταγραφή της αναφοράς με την ημερομηνία στην οποία αυτή καταχωρείται.

Ο κύριος τρόπος συλλογής δεδομένων και πλοήγησης στο δέντρο του κάθε έργου είναι ο προσδιορισμός των στοιχείων που μας ενδιαφέρουν με ποιο σταθερό σημείο του html εγγράφου συνδέονται. Αυτή είναι και η γενικότερη ιδέα λειτουργίας του συστήματος.

Πλήρης παρουσίαση του κώδικα του συστήματος υπάρχει στο τέλος του εγγράφου, στο παράρτημα Α.

ΚΕΦΑΛΑΙΟ 5Ο
ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΕΠΕΚΤΑΣΙΜΟΤΗΤΑ

5.1 ΣΥΜΠΕΡΑΣΜΑΤΑ

Κατά την υλοποίηση της πτυχιακής εργασίας έγινε αρκετή προσπάθεια για να εκπληρώσει τους αρχικούς στόχους της, οι οποίοι ήταν να αποσπά κάποιες από τις πληροφορίες για έργα ανοιχτού λογισμικού. Το σύστημα όπως και κάθε σύστημα που γεννιέται έχει φυσικά πολλά περιθώρια βελτιστοποίησης σε τομείς που θα αναφερθούν αλλά πιθανότατα και σε άλλα σημεία.

Αυτό που αξίζει να τονιστεί, και μπορεί να βοηθήσει το σύστημα να βελτιωθεί εύκολα και γρήγορα, είναι η αυτονομία του. Δεν εξαρτάται από κανένα εξωτερικό υπάρχον πρόγραμμα και λειτουργεί αυτόνομα.

5.1.1 ΕΞΕΛΙΞΗ ΤΟΥ ΕΡΓΟΥ “CHICKEN”

Αποτελέσματα και συμπεράσματα μπορούμε να αντλήσουμε από τα γραφήματα που παρουσιάζονται παρακάτω και μας δίνουν την εξέλιξη του έργου “chicken” του οποίου τα αποτελέσματα παρουσιάζονται και στο παράρτημα Β στο τέλος αυτού του εγγράφου.

5.2 USER INTERFACE

Ένας από τους τομείς στους οποίους μπορεί να βελτιωθεί το σύστημα είναι στη διάδραση του με το χρήστη. Δηλαδή, παρόλο που το σύστημα φαίνεται να δουλεύει χωρίς ιδιαίτερες απαιτήσεις από το χρήστη έχει αδύνατα σημεία. Ένα από αυτά είναι ότι ο χρήστης για να θέσει σε λειτουργία τον σύστημα θα πρέπει να του δώσει επακριβώς το όνομα του έργου που τον ενδιαφέρει χωρίς το παραμικρό συντακτικό λάθος. Θα ήταν καλό σε επόμενες εκδόσεις η λειτουργία αυτή να γίνει αρκετά πιο ευέλικτη και να μπορεί το σύστημα να βρίσκει ευκολότερα τα project που ενδιαφέρουν το χρήστη του χωρίς ο δεύτερος να πρέπει να γνωρίζει επακριβώς το όνομα του έργου που τον ενδιαφέρει. Η αυτόματη συμπλήρωση του ονόματος και η υποστήριξη αναζήτησης θα ήταν δύο μόνο από τις προτεινόμενες λύσιες. Επίσης ένα ακόμη κομμάτι του interface που θα μπορούσε να βελτιωθεί είναι ο ορισμός συγκεκριμένων στοιχείων από αυτά που προσφέρει το τρέχον σύστημα. Για παράδειγμα η ανάγκη από τον χρήστη να ανακτήσει μόνο αναφορές λήψεων και όχι αναφορές tracker ή ο προσδιορισμός συγκεκριμένου εύρους ημερομηνιών για το οποίο θέλει να λάβει αποτελέσματα. Αυτή η βελτίωση θα βοηθούσε σημαντικά και στη μείωση του χρόνου εκτέλεσης του συτήματος μιάς και η τρέχουσα έκδοση του

καλύπτει μεγάλο εύρος ημερομηνιών και άρα επεξεργάζεται μεγάλο όγκο δεδομένων.

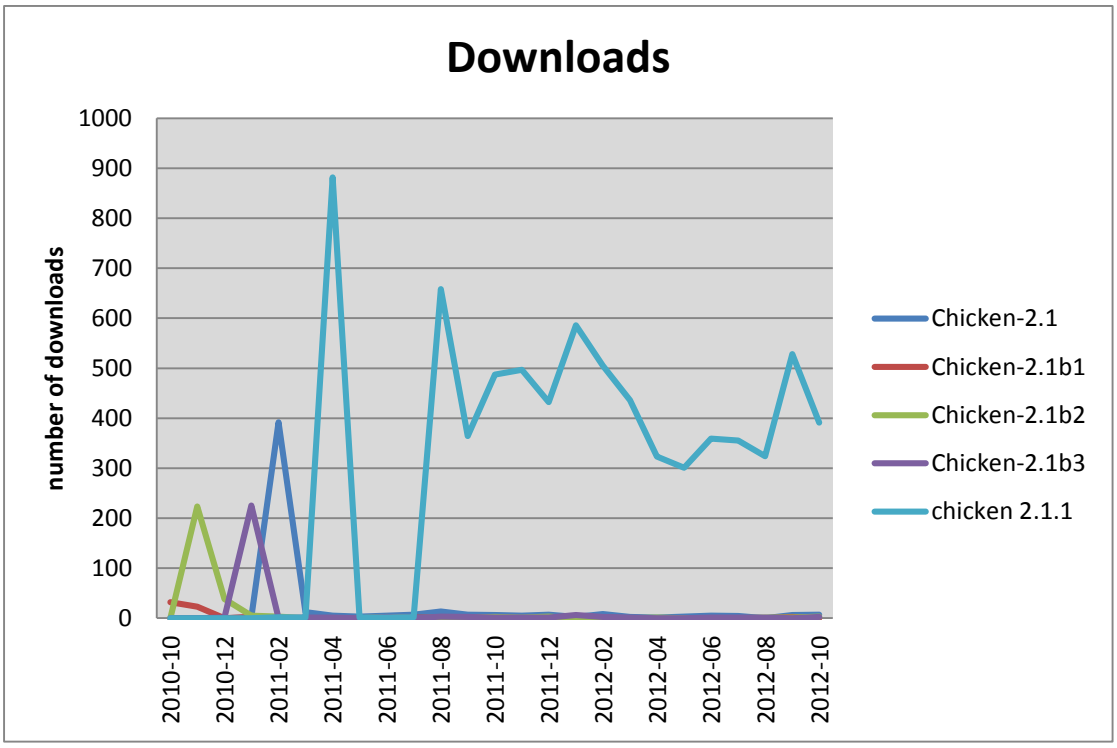
Ένα ακόμα σημείο στο οποίο υστερεί αυτή η πρώτη προσπάθεια υλοποίησης του συστήματος είναι το εύρος των ημερομηνιών στις αναφορές των τεσσάρων κατηγοριών Bugs, Feature Requests, Patches και Support Requests. Η ημερομηνία έναρξης ανάκτησης μετρικών του συστήματος είναι η 01-01-2008 αλλά παρόλο που για τις αναφορές λήψεων το σύστημα αυτόματα προσαρμόζει τις ημερομηνίες σύμφωνα με την ημερομηνία "γέννησης" του έργου στο *sourceforge.net,* δηλαδή αν η ημερομηνία ανεβάσματος του πρώτου αρχείου ενός έργου είναι η 05-06-2010, τα αποτελέσματα των αναφορών λήψεων θα ξεκινήσουν από αυτή την ημερομηνία. Αντίθετα, τα αποτελέσματα αναφορών tracker ξεκινούν πάντα από την ημερομηνία αρχικοποίησης του συστήματος, δηλαδή την 01-01-2008. Αυτό δεν σημαίνει βέβαια ότι υπάρχει κάποιο λάθος στα αποτελέσματα ή ότι παρουσιάζονται ψευδή στοιχεία, απλά οι πίνακες αποτελεσμάτων ξεκινούν από αυτή την ημερομηνία με μηδενικές τιμές. Ένα τέτοιο παράδειγμα είναι και το αποτέλεσμα εκτέλεσης του συστήματος για το έργο "chicken", το οποίο παρουσιάζεται στο παράρτημα Β στο τέλος του εγγράφου αυτού.

5.3 PARSING

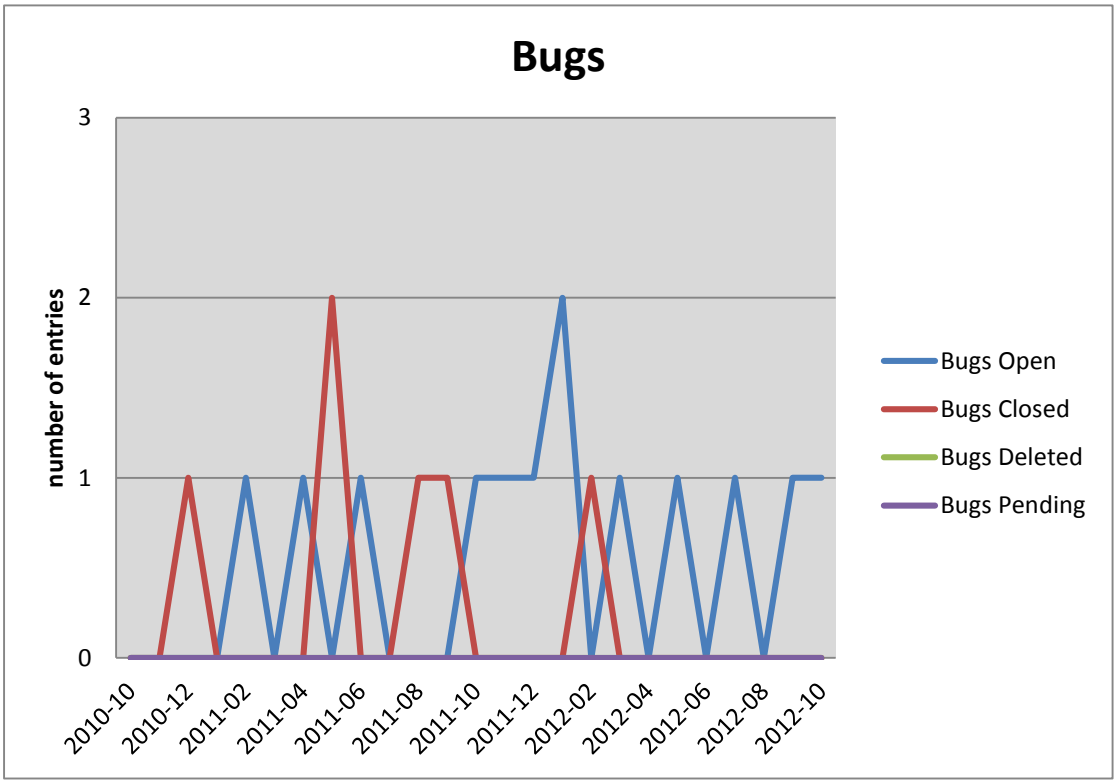
Στο θέμα της συλλογής πληροφοριών υπάρχουν αρκετά περιθώρια βελτίωσης του συστήματος που αφορούν τόσο στην βελτίωση της ταχύτητας της διαδικασίας όσο και στον όγκο των δεδομένων που αποσπώνται από τη σελίδα του *sourceforge.net*. Για τη βελτίωση της ταχύτητας θα ήταν χρήσιμο να παραλληλοποιηθεί η διαδικασία έτσι ώστε κομμάτια του έργου που γίνονται σειριακά και το ένα δεν επηρεάζει το άλλο θα μπορούσαν να γίνονται παράλληλα και να βελτιώσουν κατά πολύ το χρόνο εκτέλεσης της εφαρμογής. Αυτό θα μπορούσε να επιτευχθεί με χρήση παράλληλου προγραμματισμού (νήματα) είτε ακόμη και με τη χρήση βιβλιοθηκών που υποστηρίζουν κατανεμημένο τρόπο λειτουργίας των εφαρμογών. Όσον αφορά τον όγκο της πληροφορίας που συλλέγει το σύστημα θα μπορούσε να γίνει εύκολα μία αύξηση του.

5.4 ΑΝΑΔΡΟΜΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ

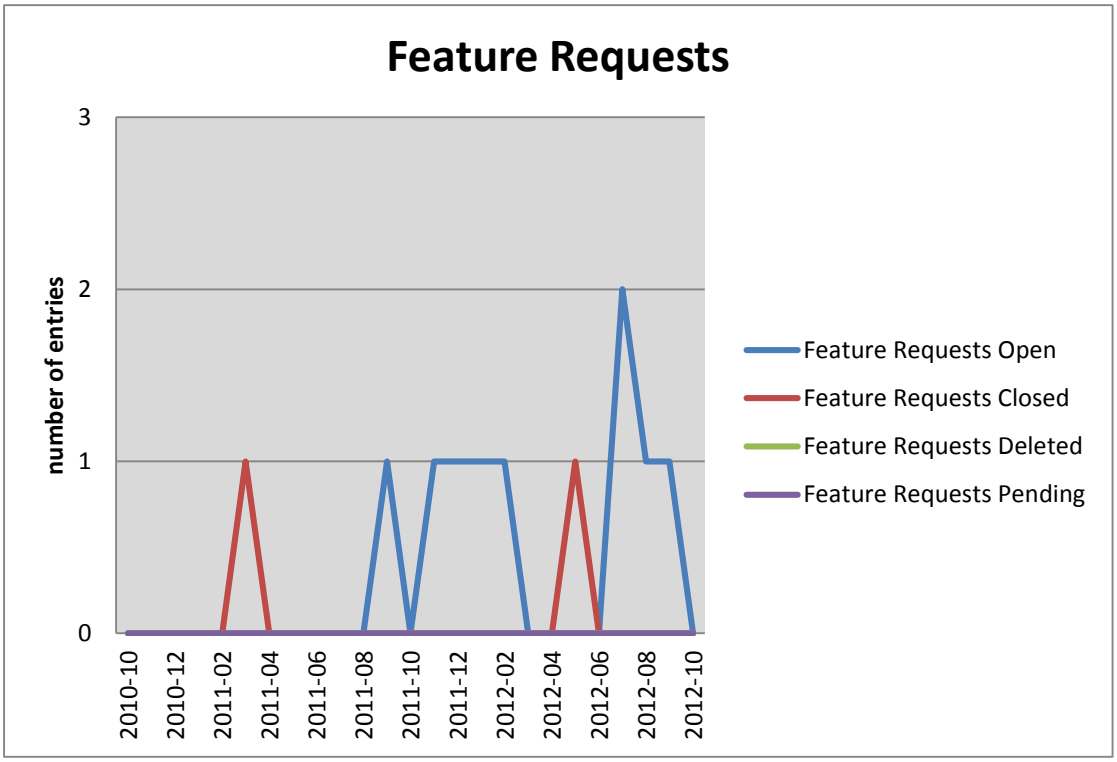
Η άντληση των πληροφοριών γίνεται με επιτυχία αλλά παρόλα αυτά υπάρχουν σημεία στον κώδικα όπου θα μπορούσαν να συρρικνωθούν κάνοντας έτσι τα αρχεία κώδικα πιο κατανοητά και έτσι επεκτάσιμα σε επόμενες εκδόσεις. Αυτό θα μπορούσε να επιτευχθεί χρησιμοποιώντας αναδρομικές διαδικασίες-συναρτήσεις.



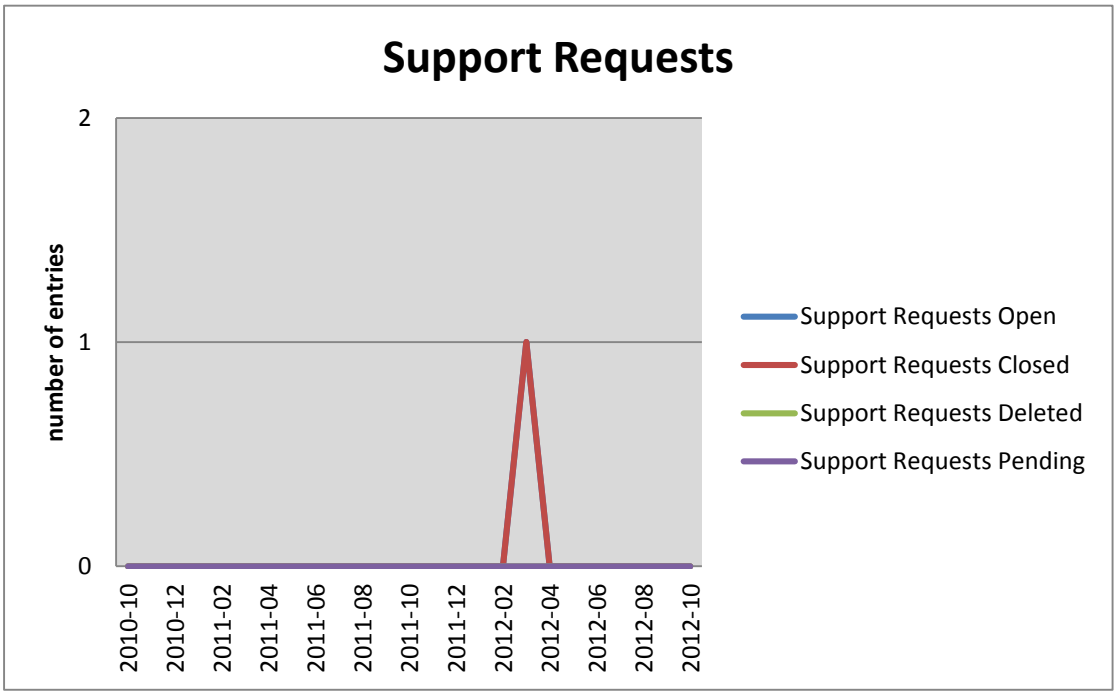
εικόνα 17 . διάγραμμα λήψεων του έργου "chicken"



εικόνα 20 . διάγραμμα Bugs του έργου "chicken"



εικόνα 21 . διάγραμμα Feature Requests του έργου "chicken"



εικόνα 22 . διάγραμμα Support Requests του έργου "chicken"

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. http://www.php.net/
2. http://www.w3schools.com/php/
3. http://pear.php.net/
4. L. A. Belady and M. M. Lehman, “A model of large program development”, Computing Science, Imperial College of Science and Technology, London, England, 1976
5. Tetsuo Tamai, Yoshuke Torimitsu, “Software lifetime and its evolution process over generations”, Proc. Conference on software maintenance – 1992, Orlando, Florida, November 1992
6. D. Gefen, S. L. Schneberger, “The NonHomogenous Maintenance Periods : A Case Study”, venue of software modifications, International Conference on Software Maintenance 1996
7. Victor R. Basili, “The SEL Adapts to Meet Changing Times”, Computer Science Department/ Institute for Advanced Computer Studies, University of Mairyland

ΕΡΓΑΛΕΙΑ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΑΝ

Notepad++	*για συγγραφή κώδικα*
Xampp for Windows	*για προσομοίωση server*
Adobe Dreamweaver	*για κάποια σημεία html editing*
Google Chrome και Mozilla Firefox	*για εκτέλεση του κώδικα και έλεγχο*

ΠΑΡΑΡΤΗΜΑ **Α**
ΚΩΔΙΚΑΣ

main.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<htmllang = "EN"dir = "ltr"xmlns =
"http://www.w3.org/1999/xhtml">
  <head>
      <title> html parsing </title>
      </head>

  <body>

     <?php
      @require_once("project.php");
      @require_once("patches.php");
      @require_once("supportReq.php");
      @require_once("bugs.php");
      @require_once("featureReq.php");

      set_time_limit(7200);

      $sfp = filter_input(INPUT_GET,"sfProject");

      //dimiourgia tou url pou deixnei sti thesi files tou ergou pou
      //exei eisagei o xristis
      $url = "http://www.sourceforge.net/projects/$sfp/files/";

      //ektypwsi tou url stin html selida eksodou
      print"<h3> Project Name : $sfp ! </h3>";

      //dimiourgia antikeimenou typou Project, kai ektelesi tou
      //prwtou ypoprogrammatos tou systimatos

      $proj = new Project($url, $exists);
      if($exists == 0) {
       //emfanisi tou pinaka anaforwn lipsewn
       $proj -> printDownloads();
       //ektelesi tou deyterou ypoprogrammatos tou systimatos
       $b = new bugs($url);
       $feat = newfeatureReq($url);
       $pat = new patches($url);
       $support = newsupportReq($url);
       } else {
        echo "<h2> Project does not exist! </h2>";
       }

      ?>

  </body>
  </html>
```


project.php

```
<?php

@require_once("file.php");

ini_set("display_errors",0);

class Project{
 private$p_url;
 private$files = array();

 publicfunction listLinks($url, &$exists){

  //echo $url;
  //elegxos tou url gia tyxon lathos eisodo tou xristi
  if($lines = file($url)){
   $exists = 0;
  } else{
   $exists = 1;
  }

  /*
  $exists = 0;
  foreach( $lines as $line_num => $line ) {
   $line = htmlspecialchars( $line );
   $found = strpos($line, "Whoops, we can't find that
    page.");
   if ($found>0 || $found === 0) {
    $exists = 1;
    break;
   }
  } */

  if($exists == 0){

   $tempArray = array();
   $listArray = array();
   $lc = 0;

   //gemisma tou pinaka listArray me ta url apo olous tous
   //fakelous kai ta arxeia pou deixnoun stis anafores lipsewn
   //tous

   foreach($linesas$line_num => $line){

    $line = htmlspecialchars($line);
    //"kleidi" to opoio anazitoume stin hmtl
    $found = strpos($line, "/stats/timeline");

    if($found != 0){

     $tempArray=str_word_count($line,1,' /-
     _=<>.%0123456789');

     //$type = strcmp($tempArray[8], "status
     //folder");
     //elegxos an stin grammi pou entopistike einai arxeio i
     //fakelos
```


```
if($tempArray[8]=="status folder"){
  $str="http://www.sourceforge.net".$tempArray[14]."/?da
  tes=2008-01-01+to+2020-01- 01";
  //ayto einai kai to simeio sto opoio rythmizoume to
  //xroniko eyros gia to opoio tha parousiastoun ta
  //apotelesmata

  $listArray[$lc] = $str;
  $lc++;
  $url_2 = substr($str,0,-47);

  //echo "url: " . $url_2 .  "<br>";
  $lines_2 = file($url_2);

  foreach($lines_2as$line_num => $line){

    $line=htmlspecialchars($line);
    $found=strpos($line,"/stats/timeline");

    if($found != 0){
      $tempArray=str_word_count($line,1,'/_=<>.%0123456789');

      //$type = strcmp($tempArray[8], "status folder");

      if($tempArray[8] == "status folder"){
              $str="http://www.sourceforge.net".$tempArray[1
              4]."/?dates=2008-01-01+to+2020-01-01";
              $listArray[$lc] = $str;
              $lc++;
              $url_3 = substr($str,0,-47);
              //echo "url: " . $url_3 . "<br>";

       $lines_3 = file($url_3);

       foreach($lines_3as$line_num=>$line){
         $line = htmlspecialchars($line);
         $found = strpos($line, "/stats/timeline");

         if($found!=0){
           $tempArray=str_word_count($line, 1,
           '/_=<>.%0123456789');
           //$type = strcmp($tempArray[8], "status folder");

           if($tempArray[8] == "status folder"){
             $str="http://www.sourceforge.net".$temp
             Array[14]."/?dates=2008-01-  01+to+2020-01-01";
             $listArray[$lc] = $str;
             $lc++;
             $url_4 = substr($str,0,-47);

             //echo "url: " . $url_4 . "<br>";

             $lines_4 = file($url_4);

             foreach($lines_3as$line_num => $line){
                $line = htmlspecialchars($line);
                $found=strpos($line, "/stats/timel
                 ine");

                if($found!=0){
                  $tempArray=str_word_count($line,1,' /-
                  _=<>.%0123456789');
```


```
                    $str="http://www.sourceforge.net".$tempArray[
                    14]."/?dates=2008-01-01+to+2020-01-01";
                    $listArray[$lc] = $str;
                    $lc++;
                  }
                }
              }
              elseif($tempArray[8] == "status file"){
                //echo "1<br>";
                //telika to simeio to opoio entopisame einai
                //arxeio kai oxi fakelos
                $str="http://www.sourceforge.net".$tempArray[14]
                ."/?dates=2008-01- 01+to+2020-01-01";
                $listArray[$lc] = $str;
                $lc++;
              }
            }
          }
        }
        elseif($tempArray[8] == "status file"){
          //echo "2<br>";
          $str="http://www.sourceforge.net".$tempArra
          y[14]."/?dates=2008-01-01+to+2020-01-01";
          $listArray[$lc] = $str;
          $lc++;
        }
      }
    }
    elseif($tempArray[8] == "status file"){
      //echo "3<br>";
      $str="http://www.sourceforge.net".$tempArray[14]."/?date
      s=2008-01-01+to+2020-01-01";
      $listArray[$lc] = $str;
      $lc++;
    }
  }
 }
 //epistrofi tis gematis listas
 Return $listArray;
 }
}

publicfunction __construct($url1, &$exists){

 $this->p_url = $url1;
 //ektelesi tis synartisis listLinks
 $listArray = $this->listLinks($this->p_url, &$exists);

 //aplos elegxos tis metablitis exists I opoia mas deixnei
 //an to project yparxei i oxi
 if($exists == 0){

  for($i=0; $i<count($listArray); $i++){
   //dimiourgia enos antikeimenou typou File me kathe link pou
   //exei dimiourgithei stin listLinks
   $f = new File($listArray[$i]);
   $this->files[count($this->files)] = $f;
  }
 }
 //call tracker here
 //$this->tracker(url1);
```


```
    }

    publicfunctionprintDownloads(){
      echo"<h2> Downloads </h2>";
      echo"<table border='1' style='margin-left:50px'>";
      echo"<tr><th>File</th><th>Date</th><th>Downloads</th></tr>" ;
      for($j=0; $j<count($this->files); $j++){
        $parts = explode("/stats",$this->files[$j]->getName());
        $parts = explode("/files/",$parts[0]);
        echo "<tr>";
        echo"<tdvalign='top'align='center'rowspan='".($this>files[$j
        ]->getNumberOfDates()- 1)."'><b><br>".$parts[1]."</b></td>";
        $this->files[$j]->printDownloads();
        echo "</tr>";
      }
      echo "</table>";
    }
  };
?>
```

file.php

```
<?php

  @require_once("downloads.php");

  class File{
    private $downloads = array();
    private $name;

    publicfunction getNumberOfDates(){
      returncount($this->downloads);
    }

    publicfunction __construct($s){

      $this->name = $s;

      $lines = file($s);

      foreach($linesas$line_num => $line){

        $found3 = strpos($line,"<td>");
        $found4 = strpos($line,"< headers=\"files_downloads_h\"");
        //diaxwrismos imerominias kai arithmou downloads

        if($found3! == false){
          $tempArray = str_word_count($line,1,'/-_=<>.%0123456789');
          $dt = $tempArray[0];
          $dt = str_replace("</td>", "", $dt);
          $dt = str_replace("<td>", "", $dt);
        }

        if($found4 !== false){
          $tempArray = str_word_count($line,1,'/-_"=.,%0123456789');
          if($dt==''){
            $dl = $tempArray[2];
            $d = new Downloads($dt,$dl);
```


```
            $this->downloads[count($this->downloads)] = $d;
          }
          else{
            $dl = $tempArray[3];
            $d = new Downloads($dt,$dl);
            $this->downloads[count($this->downloads)] = $d;
          }
        }
      }
    }

    publicfunction getName(){
      return $this->name;
    }

    publicfunction printDownloads(){
       for($j=0; $j<(count($this->downloads)-1); $j++){
        $this->downloads[$j]->printDownloads($j);
       }
    }
  };
?>
```

downloads.php

```
<?php
  class Downloads {
    private $date;
    private $downloads;

    publicfunction __construct($dt, $dl){
      $this->date = $dt;
      $this->downloads = $dl;
    }

    public function printDownloads($j){
      if($j == 0){
        echo "<td>".$this->date."</td>";
        echo "<td align='center'>".$this->downloads ."</td>";
      }else{
        echo "<tr>";
        echo "<td>".$this->date."</td>";
        echo "<td align='center'>".$this->downloads ."</td>";
        echo "</tr>";
      }
    }
  };
?>
```


bugs.php

```php
<?php

 class bugs{
  private $p_url;
  private $files = array();

  public function __construct($url1){

    $this->p_url = $url1;
    $flag = 0;
    $lines = file($url1);

    foreach($linesas$line_num => $line){
     $line = htmlspecialchars($line);
     $found = strpos($line, "menu_tracker");
     if($flag == 1){
      $tempArray=str_word_count($line,1,' /-_=<>.%0123456789');
      $new_url="http://www.sourceforge.net/tracker/?".$tempArray[
      5];
      //echo "<br>" . "new url : " . $new_url . "<br><br>";
      $flag++;
     }
     if($flag == 0 && $found != 0){
      //echo "<br> Tracker found for this project! <br>";
      $flag++;
     }
    }

    if($flag == 0){
     echo"<h2><br>Bug Tracker not found for this
     project!</h2><br>";
    }
    else{
     $lines = file($new_url);
     $flag = 0;
     $counter = 0;
     $myArray = array();
     foreach($linesas$line_num => $line){
      $line = htmlspecialchars($line);
      $found = strpos($line, "func=browse");
      if($flag == 1){
       $tempArray=str_word_count($line,1,' /-
       _=<>&;?.%0123456789');
       $type1 = substr($tempArray[1], 33, -4);
       $flag = 0;
       $myArray[$counter][3] = $type1;
       $counter++;
      }
      if($found != 0){
       $tempArray=str_word_count($line,1,'
       /_=<>&;?.%0123456789');
       $link1 = substr($tempArray[0], 61, -10);
       $tempArray=str_word_count($link1,1,' /-
       _<>&;?.%0123456789');
       $atid = substr($tempArray[1], 0, -17);
       $gid = substr($tempArray[2], 0, -13);
       $link1="http://www.sourceforge.net/tracker/?limit=100&ati
```


```
    d=".$atid."&group_id=".$gid."&func=browse";
    $flag = 1;
    $myArray[$counter] = array();
    $myArray[$counter][0] = $link1;
    $myArray[$counter][1] = $gid;
    $myArray[$counter][2] = $atid;
  }
}

//print_r($myArray);

$openBugs = array();
$closedBugs = array();
$deletedBugs = array();
$pendingBugs = array();

$myNewArray = array();
$o = 0;
$c = 0;
$d = 0;
$p = 0;
$flag = 100;
$status = 'N';
$offset = 0;
$i = 0;
$type = -1;

for($i=0; $i<count($myArray); $i++){
  if($myArray[$i][3] == "Bugs"){
    $type = $i;
  }
}
if($type != -1){
  do{

    $flag2 = 0;
    $myNewArray[$i]="http://www.sourceforge.net/tracker?words
    =tracker_browse&sort=open_date&sortdir=desc&offset=".$off
    set."&atid=".$myArray[$type][2]."&group_id=".$myArray[$ty
    pe][1]."&func=browse";

    //echo $myNewArray[$i]. "<br>";

    $lines = file($myNewArray[$i]);

    foreach($lines as $line_num => $line){
      $line = htmlspecialchars($line);
      $foundN = strpos($line, "Next");
      if($foundN != 0){
        $flag2 = 1;
      }
      $foundO = strpos($line, "Open");
      $foundC = strpos($line, "Closed");
      $foundD = strpos($line, "Deleted");
      $foundP = strpos($line, "Pending");
      if($flag <= 4){
        $flag++;
        if($flag == 4 && $status == 'O'){
          $openBugs[$o] = substr($line, 10, -4);
          $o++;
          $status = 'N';
        }
```


```
      elseif($flag == 4 && $status == 'C'){
        $closedBugs[$c] = substr($line, 10, -4);
        $c++;
        $status = 'N';
      }
      elseif($flag == 4 && $status == 'D'){
        $deletedBugs[$d] = substr($line, 10, -4);
        $d++;
        $status = 'N';
      }
      elseif($flag == 4 && $status == 'P'){
        $pendingBugs[$p] = substr($line, 10, -4);
        $p++;
        $status = 'N';
      }
    }
    if($foundO == 10){
      $flag = 0;
      $status = 'O';
    }
    elseif($foundC == 10){
      $flag = 0;
      $status = 'C';
    }
    elseif($foundD == 10){
      $flag = 0;
      $status = 'D';
    }
    elseif($foundP == 10){
      $flag = 0;
      $status='P';
    }
  }

  $offset = $offset+25;
  $i++;

}while($flag2 == 1);

$tracker = array();

$year = 2008;
$month = 01;
$i = 0;

while(!($year == 2012 && $month == 11)){
  if($month < 10){
    $tracker[$i] = $year . "-0" . $month;
  }
  else{
    $tracker[$i] = $year . "-" . $month;
  }

  if($month <= 11){
    $month++;
  }
  else{
    $month = 01;
    $year++;
  }
  $i++;
}
```


```
    $i = 0;
    $open = array();
    $closed = array();
    $deleted = array();
    $pending = array();

    foreach($tracker as $date){

      $open[$i] = 0;
      $closed[$i] = 0;
      $deleted[$i] = 0;
      $pending[$i] = 0;

      foreach($openBugs as $ob){
        if($date == $ob){
          $open[$i]++;
        }
      }
      foreach($closedBugs as $cb){
        if($date == $cb){
          $closed[$i]++;
        }
      }
      foreach($deletedBugs as $db){
        if($date == $db){
          $deleted[$i]++;
        }
      }
      foreach($pendingBugs as $pb){
        if($date == $pb){
          $pending[$i]++;
        }
      }
      $i++;
    }

    echo "<br><h2>Bugs</h2><br>";
    echo "<table style='margin-left:50px' border='1'>";
    echo "<tr>";
    echo "<th>Date</th>";
    echo "<th>Bugs Open</th>";
    echo "<th>Bugs Closed</th>";
    echo "<th>Bugs Deleted</th>";
    echo "<th>Bugs Pending</th>";
    echo "</tr>";
    for($i=0; $i<count($tracker); $i++){
      echo "<tr>";
      echo "<td align='center'>" . $tracker[$i] . "</td>";
      echo "<td align='center'>" . $open[$i] . "</td>";
      echo "<td align='center'>" . $closed[$i] . "</td>";
      echo "<td align='center'>" . $deleted[$i] . "</td>";
      echo "<td align='center'>" . $pending[$i] . "</td>";
      echo "</tr>";
    }
    echo "</table>";
  }
  else{
    echo "<h2><br>Bug Tracker not found for this
    project!</h2><br>";
  }
}
```


```
    }
  };

?>
```


featureRequests.php

```
<?php

 class featureReq{
  private $p_url;
  private $files = array();

  publicfunction __construct($url1){

    $this->p_url = $url1;
    $flag = 0;
    $lines = file($url1);

    foreach($lines as $line_num => $line){
     $line = htmlspecialchars($line);
     $found = strpos($line, "menu_tracker");
     if($flag == 1){
      $tempArray=str_word_count($line,1,' /-_=<>.%0123456789');
      $new_url="http://www.sourceforge.net/tracker/?".$tempAr
      ray[5];
      //echo "<br>" . "new url : " . $new_url . "<br><br>";
      $flag++;
     }
     if($flag == 0 && $found != 0){
      //echo "<br> Tracker found for this project! <br>";
      $flag++;
     }
    }

    if($flag == 0){
     echo"<h2> <br> Feature Request Tracker not found for this
     project! </h2> <br>";
    }
    else{
     $lines = file($new_url);
     $flag = 0;
     $counter = 0;
     $myArray = array();
     foreach($lines as $line_num => $line){
      $line = htmlspecialchars($line);
      $found = strpos($line, "func=browse");
      if($flag == 1){
       $tempArray = str_word_count($line,1,' /-
       _=<>&;?.%0123456789');
       $type1 = substr($tempArray[1], 33, -4);
       $flag = 0;
       $myArray[$counter][3] = $type1;
       $counter++;
      }
      if($found != 0){
       $tempArray = str_word_count($line,1,' /-
       _=<>&;?.%0123456789');
       $link1 = substr($tempArray[0], 61, -10);
       $tempArray = str_word_count($link1, 1, ' /-
       _<>&;?.%0123456789');
       $atid = substr($tempArray[1], 0, -17);
       $gid = substr($tempArray[2], 0, -13);
```


```
    $link1 = "http://www.sourceforge.net/tracker/?limit=10
    0&atid=".$atid."&group_id=".$gid."&func=browse";
    $flag = 1;
    $myArray[$counter] = array();
    $myArray[$counter][0] = $link1;
    $myArray[$counter][1] = $gid;
    $myArray[$counter][2] = $atid;
  }
}

//print_r($myArray);

$openFR = array();
$closedFR = array();
$deletedFR = array();
$pendingFR = array();

$myNewArray = array();
$o = 0;
$c = 0;
$d = 0;
$p = 0;
$flag = 100;
$status = 'N';
$offset = 0;
$i = 0;
$type = -1;

for($i=0; $i<count($myArray); $i++){
  if($myArray[$i][3] == "Feature Requests"){
    $type = $i;
  }
}

if($type != -1){
  do{
    $flag2 = 0;
    $myNewArray[$i] = "http://www.sourceforge.net/tracker/
    ?words=tracker_browse&sort=open_date&sortdir=desc&of
    fset=".$offset."&atid=".$myArray[$type][2]."&group_i
    d=".$myArray[$type][1]."&func=browse";
    //echo $myNewArray[$i]. "<br>";
    $lines = file($myNewArray[$i]);

    foreach($linesas$line_num=>$line){
      $line = htmlspecialchars($line);
      $foundN = strpos($line, "Next");
      $slen = strlen($line);
      if($foundN != 0 && $slen > 200){
        $flag2 = 1;
      }
      $foundO = strpos($line, "Open");
      $foundC = strpos($line, "Closed");
      $foundD = strpos($line, "Deleted");
      $foundP = strpos($line, "Pending");
      if($flag <= 4){
        $flag++;
        if($flag == 4 && $status == 'O'){
          $openFR[$o] = substr($line, 10, -4);
          $o++;
          $status = 'N';
        }
```


```
      elseif($flag == 4 && $status == 'C'){
        $closedFR[$c] = substr($line, 10, -4);
        $c++;
        $status = 'N';
      }
      elseif($flag == 4 && $status == 'D'){
        $deletedFR[$d] = substr($line, 10, -4);
        $d++;
        $status = 'N';
      }
      elseif($flag == 4 && $status == 'P'){
        $pendingFR[$p] = substr($line, 10, -4);
        $p++;
        $status = 'N';
      }
    }
    if($foundO == 10){
      $flag = 0;
      $status = 'O';
    }
    elseif($foundC == 10){
      $flag = 0;
      $status = 'C';
    }
    elseif($foundD == 10){
      $flag = 0;
      $status = 'D';
    }
    elseif($foundP == 10){
      $flag = 0;
      $status = 'P';
    }
  }

  $offset = $offset+25;
  $i++;

}while($flag2 == 1);

$tracker = array();

$year = 2008;
$month = 01;
$i = 0;

while(!($year == 2012 && $month == 11)){
  if($month < 10){
    $tracker[$i] = $year . "-0" . $month;
  }
  else{
    $tracker[$i] = $year . "-" . $month;
  }

  if($month <= 11){
    $month++;
  }
  else{
    $month = 01;
    $year++;
  }
  $i++;
}
```


```
    $i = 0;
    $open = array();
    $closed = array();
    $deleted = array();
    $pending = array();

    foreach($tracker as $date){

      $open[$i] = 0;
      $closed[$i] = 0;
      $deleted[$i] = 0;
      $pending[$i] = 0;

      foreach($openFR as $ob){
        if($date == $ob){
          $open[$i]++;
        }
      }
      foreach($closedFR as $cb){
        if($date == $cb){
          $closed[$i]++;
        }
      }
      foreach($deletedFR as $db){
        if($date == $db){
          $deleted[$i]++;
        }
      }
      foreach($pendingFR as $pb){
        if($date == $pb){
          $pending[$i]++;
        }
      }
      $i++;
    }

    echo "<br><h2> Feature Requests</h2><br>";
    echo "<table style='margin-left:50px' border='1'>";
    echo "<tr>";
    echo "<th>Date</th>";
    echo "<th>Feature Requests Open</th>";
    echo "<th>Feature Requests Closed</th>";
    echo "<th>Feature Requests Deleted</th>";
    echo "<th>Feature Requests Pending</th>";
    echo "</tr>";
    for($i=0; $i<count($tracker); $i++){
      echo "<tr>";
      echo "<td align='center'>" . $tracker[$i] . "</td>";
      echo "<td align='center'>" . $open[$i] . "</td>";
      echo "<td align='center'>" . $closed[$i] . "</td>";
      echo "<td align='center'>" . $deleted[$i] . "</td>";
      echo "<td align='center'>" . $pending[$i] . "</td>";
      echo "</tr>";
    }
    echo "</table>";
  }
  else{
    echo "<h2><br>Feature Request Tracker not found for this
    project!</h2><br>";
  }
}
```


```
    }
  };

?>
```

patches.php

```
<?php

 class patches{
  private $p_url;
  private $files = array();

  publicfunction __construct($url1){

    $this->p_url = $url1;
    $flag = 0;
    $lines = file($url1);

    foreach($linesas$line_num => $line){
     $line = htmlspecialchars($line);
     $found = strpos($line, "menu_tracker");
     if($flag == 1){
      $tempArray = str_word_count($line,1,' /-_=<>.%0123456789');
      $new_url = "http://www.sourceforge.net/tracker/?".$tempAr
      ra y[5];
      //echo "<br>" . "new url : " . $new_url . "<br><br>";
      $flag++;
     }
     if($flag == 0 && $found != 0){
      //echo "<br> Tracker found for this project! <br>";
      $flag++;
     }
    }

    if($flag == 0){
     echo "<h2><br>Patches Tracker not found for this
     project!</h2><br>";
    }
    else{
     $lines = file($new_url);
     $flag = 0;
     $counter = 0;
     $myArray = array();
     foreach($lines as $line_num => $line){
      $line = htmlspecialchars($line);
      $found = strpos($line, "func=browse");
      if($flag == 1){
       $tempArray = str_word_count($line,1,' /-
       _=<>&;?.%0123456789');
       $type1 = substr($tempArray[1], 33, -4);
       $flag = 0;
       $myArray[$counter][3] = $type1;
       $counter++;
      }
      if($found != 0){
       $tempArray = str_word_count($line, 1, ' /-
       _=<>&;?.%0123456789');
       $link1 = substr($tempArray[0], 61, -10);
       $tempArray = str_word_count($link1, 1, ' /-
       _<>&;?.%0123456789');
       $atid = substr($tempArray[1], 0, -17);
       $gid = substr($tempArray[2], 0, -13);
```


```
    $link1 = "http://www.sourceforge.net/tracker/?limit=10
    0&atid=".$atid."&group_id=".$gid."&func=browse";
    $flag = 1;
    $myArray[$counter] = array();
    $myArray[$counter][0] = $link1;
    $myArray[$counter][1] = $gid;
    $myArray[$counter][2] = $atid;
  }
}

//print_r($myArray);

$openPatches = array();
$closedPatches = array();
$deletedPatches = array();
$pendingPatches = array();

$myNewArray = array();
$o = 0;
$c = 0;
$d = 0;
$p = 0;
$flag = 100;
$status = 'N';
$offset = 0;
$i = 0;
$type = -1;

for($i=0; $i<count($myArray); $i++){
  if($myArray[$i][3] == "Patches"){
    $type = $i;
  }
}

if($type != -1){
  do{
    $flag2 = 0;
    $myNewArray[$i] = "http://www.sourceforge.net/tracker/
    ?words=tracker_browse&sort=open_date&sortdir=desc&of
    fset=".$offset."&atid=".$myArray[$type][2]."&group_i
    d=".$myArray[$type][1]."&func=browse";
    //echo $myNewArray[$i]. "<br>";
    $lines = file($myNewArray[$i]);

    foreach($linesas$line_num => $line){
      $line = htmlspecialchars($line);
      $foundN = strpos($line, "Next");
      if($foundN != 0){
        $flag2 = 1;
      }
      $foundO = strpos($line, "Open");
      $foundC = strpos($line, "Closed");
      $foundD = strpos($line, "Deleted");
      $foundP = strpos($line, "Pending");
      if($flag <= 4){
        $flag++;
        if($flag == 4 && $status == 'O'){
          $openPatches[$o] = substr($line, 10, -4);
          $o++;
          $status = 'N';
        }
        elseif($flag == 4 && $status == 'C'){
```


```
            $closedPatches[$c] = substr($line, 10, -4);
            $c++;
            $status = 'N';
          }
          elseif($flag == 4 && $status == 'D'){
            $deletedPatches[$d] = substr($line, 10, -4);
            $d++;
            $status = 'N';
          }
          elseif($flag == 4 && $status == 'P'){
            $pendingPatches[$p] = substr($line, 10, -4);
            $p++;
            $status = 'N';
          }
        }
        if($foundO == 10){
          $flag = 0;
          $status = 'O';
        }
        elseif($foundC == 10){
          $flag = 0;
          $status = 'C';
        }
        elseif($foundD == 10){
          $flag = 0;
          $status = 'D';
        }
        elseif($foundP == 10){
          $flag = 0;
          $status = 'P';
        }
      }

      $offset = $offset+25;
      $i++;

    }while($flag2 == 1);

    $tracker = array();

    $year = 2008;
    $month = 01;
    $i = 0;

    while(!($year == 2012 && $month == 11)){
      if($month < 10){
        $tracker[$i] = $year . "-0" . $month;
      }
      else{
        $tracker[$i] = $year . "-" . $month;
      }

      if($month <= 11){
        $month++;
      }
      else{
        $month = 01;
        $year++;
      }
      $i++;
    }
```


```
    $i = 0;
    $open = array();
    $closed = array();
    $deleted = array();
    $pending = array();

    foreach($tracker as $date){

      $open[$i] = 0;
      $closed[$i] = 0;
      $deleted[$i] = 0;
      $pending[$i] = 0;

      foreach($openPatches as $ob){
        if($date == $ob){
          $open[$i]++;
        }
      }
      foreach($closedPatches as $cb){
        if($date == $cb){
          $closed[$i]++;
        }
      }
      foreach($deletedPatches as $db){
        if($date == $db){
          $deleted[$i]++;
        }
      }
      foreach($pendingPatches as $pb){
        if($date == $pb){
          $pending[$i]++;
        }
      }
      $i++;
    }

    echo "<br><h2>Patches</h2><br>";
    echo "<table style='margin-left:50px' border='1'>";
    echo "<tr>";
    echo "<th>Date</th>";
    echo "<th>Patches Open</th>";
    echo "<th>Patches Closed</th>";
    echo "<th>Patches Deleted</th>";
    echo "<th>Patches Pending</th>";
    echo "</tr>";
    for($i=0; $i<count($tracker); $i++){
      echo "<tr>";
      echo "<td align='center'>" . $tracker[$i] . "</td>";
      echo "<td align='center'>" . $open[$i] . "</td>";
      echo "<td align='center'>" . $closed[$i] . "</td>";
      echo "<td align='center'>" . $deleted[$i] . "</td>";
      echo "<td align='center'>" . $pending[$i] . "</td>";
      echo "</tr>";
    }
    echo "</table>";
  }
  else{
    echo "<h2><br>Patches Tracker not found for this
    project!</h2><br>";
  }
 }
}
```


```
    };
?>
```


supportRequests.php

```
<?php

 class supportReq{
  private $p_url;
  private $files = array();

  publicfunction __construct($url1){

    $this->p_url = $url1;
    $flag = 0;
    $lines = file($url1);

    foreach($lines as $line_num => $line){
     $line = htmlspecialchars($line);
     $found = strpos($line, "menu_tracker");
     if($flag == 1){
      $tempArray = str_word_count($line,1,' /-_=<>.%0123456789');

      $new_url = "http://www.sourceforge.net/tracker/?".$tempAr
      ray[5];
      //echo "<br>" . "new url : " . $new_url . "<br><br>";
      $flag++;
     }
     if($flag == 0 && $found != 0){

      //echo "<br>Tracker found for this project!<br>";
      $flag++;
     }
    }

    if($flag == 0){
     echo "<h2><br> Support Requests Tracker not found for this
     project! </h2><br>";
    }
    else{
     $lines = file($new_url);
     $flag = 0;
     $counter = 0;
     $myArray = array();
     foreach($lines as $line_num => $line){
      $line = htmlspecialchars($line);
      $found = strpos($line, "func=browse");
      if($flag == 1){
       $tempArray = str_word_count($line,1,' /-
       _=<>&;?.%0123456789');
       $type1 = substr($tempArray[1], 33, -4);
       $flag = 0;
       $myArray[$counter][3] = $type1;
       $counter++;
      }
      if($found != 0){
       $tempArray = str_word_count($line, 1, ' /-
       _=<>&;?.%0123456789');
       $link1 = substr($tempArray[0], 61, -10);
       $tempArray = str_word_count($link1, 1, ' /-
       _<>&;?.%0123456789');
       $atid = substr($tempArray[1], 0, -17);
```


```
    $gid = substr($tempArray[2], 0, -13);
    $link1="http://www.sourceforge.net/tracker/?limit=100&ati
    d=".$atid."&group_id=".$gid."&func=browse";
    $flag = 1;
    $myArray[$counter] = array();
    $myArray[$counter][0] = $link1;
    $myArray[$counter][1] = $gid;
    $myArray[$counter][2] = $atid;
  }
}

//print_r($myArray);

$openSR = array();
$closedSR = array();
$deletedSR = array();
$pendingSR = array();

$myNewArray = array();
$o = 0;
$c = 0;
$d = 0;
$p = 0;
$flag = 100;
$status = 'N';
$offset = 0;
$i = 0;
$type = -1;

for($i=0; $i<count($myArray); $i++){
  if($myArray[$i][3] == "Support Requests"){
    $type = $i;
  }
}

if($type != -1){
  do{
    $flag2 = 0;
    $myNewArray[$i] = "http://www.sourceforge.net/tracker/
    ?words=tracker_browse&sort=open_date&sortdir=desc&of
    fset=".$offset."&atid=".$myArray[$type][2]."&group_i
    d=".$myArray[$type][1]."&func=browse";
    //echo $myNewArray[$i]. "<br>";
    $lines = file($myNewArray[$i]);

    foreach($lines as $line_num => $line){
    $line = htmlspecialchars($line);
    $foundN = strpos($line, "Next");
    if($foundN != 0){
       $flag2 = 1;
    }
    $foundO = strpos($line, "Open");
    $foundC = strpos($line, "Closed");
    $foundD = strpos($line, "Deleted");
    $foundP = strpos($line, "Pending");
    if($flag <= 4){
      $flag++;
      if($flag == 4 && $status == 'O'){
        $openSR[$o] = substr($line, 10, -4);
        $o++;
        $status = 'N';
      }
```


```
      elseif($flag == 4 && $status == 'C'){
        $closedSR[$c] = substr($line, 10, -4);
        $c++;
        $status = 'N';
      }
      elseif($flag == 4 && $status == 'D'){
        $deletedSR[$d] = substr($line, 10, -4);
        $d++;
        $status = 'N';
      }
      elseif($flag == 4 && $status == 'P'){
        $pendingSR[$p] = substr($line, 10, -4);
        $p++;
        $status = 'N';
      }
    }
    if($foundO == 10){
      $flag = 0;
      $status = 'O';
    }
    elseif($foundC == 10){
      $flag = 0;
      $status = 'C';
    }
    elseif($foundD == 10){
      $flag = 0;
      $status = 'D';
    }
    elseif($foundP == 10){
      $flag = 0;
      $status = 'P';
    }
  }

  $offset = $offset + 25;
  $i++;

}while($flag2 == 1);

$tracker = array();

$year = 2008;
$month = 01;
$i = 0;

while(!($year == 2012 && $month == 11)){
  if($month < 10){
    $tracker[$i] = $year . "-0" . $month;
  }
  else{
    $tracker[$i] = $year . "-" . $month;
  }

  if($month <= 11){
    $month++;
  }
  else{
    $month = 01;
    $year++;

  }
  $i++;
```


```
  }

  $i = 0;
  $open = array();
  $closed = array();
  $deleted = array();
  $pending = array();

  foreach($tracker as $date){

    $open[$i] = 0;
    $closed[$i] = 0;
    $deleted[$i] = 0;
    $pending[$i] = 0;

    foreach($openSR as $ob){
      if($date == $ob){
        $open[$i]++;
      }
    }
    foreach($closedSR as $cb){
      if($date == $cb){
        $closed[$i]++;
      }
    }
    foreach($deletedSR as $db){
      if($date == $db){
        $deleted[$i]++;
      }
    }
    foreach($pendingSR as $pb){
      if($date == $pb){
        $pending[$i]++;
      }
    }
    $i++;
  }

  echo "<br><h2>Support Requests</h2><br>";
  echo "<table style='margin-left:50px' border='1'>";
  echo "<tr>";
  echo "<th>Date</th>";
  echo "<th>Support Requests Open</th>";
  echo "<th>Support Requests Closed</th>";
  echo "<th>Support Requests Deleted</th>";
  echo "<th>Support Requests Pending</th>";
  echo "</tr>";
  for($i=0; $i<count($tracker); $i++){
    echo "<tr>";
    echo "<td align='center'>" . $tracker[$i] . "</td>";
    echo "<td align='center'>" . $open[$i] . "</td>";
    echo "<td align='center'>" . $closed[$i] . "</td>";
    echo "<td align='center'>" . $deleted[$i] . "</td>";
    echo "<td align='center'>" . $pending[$i] . "</td>";
    echo "</tr>";
  }
  echo "</table>";
}
else{
  echo "<h2><br> Support Requests Tracker not found for
`` this project! </h2><br>";
}
```


```
        }
      }
    };

?>
```


ΠΑΡΑΡΤΗΜΑ **Β**
ΑΠΟΤΕΛΕΣΜΑΤΑ “ΠΛΗΡΟΥΣ” ΕΡΓΟΥ

Παρουσιάζεται η μορφή στην οποία ο χρήστης λαμβάνει τα αποτελέσματα από το σύστημα. Ο τρόπος παρουσιάσης τους είναι βέβαια σε μία html σελίδα μέσω κάποιου browser. Το συγκεκριμένο έργο είναι ένα πλήρες έργο όπως αναφέρεται και στον τίτλο του παραρτήματος, δηλαδή είναι ένα έργο το οποίο περιλαμβάνει εκτός από της λήψεις (downloads) και όλες τις κατηγορίες tracker (Bugs, Feature Requests, Patches και Support Requests). Σε άλλα έργα που δεν διαθέτουν αναφορές tracker, καμία ή κάποιες από αυτές, ο τρόπος εμφάνισης είναι ακριβώς ίδιος μόνο που στη θέση του κάθε πίνακα από τους τέσσερις εμφανίζεται το κατάλληλο μήνυμα μή-διάθεσης αναφορών.

Project Name : chicken !

Downloads

File	Date	Downloads
Chicken-2.1.1	2010-10	0
	2010-11	0
	2010-12	0
	2011-01	0
	2011-02	1,518
	2011-03	1,227
	2011-04	882
	2011-05	1,070
	2011-06	1,113
	2011-07	1,612
	2011-08	658
	2011-09	364
	2011-10	487
	2011-11	497
	2011-12	432
	2012-01	586
	2012-02	505
	2012-03	436
	2012-04	323
	2012-05	301
	2012-06	359

	2012-07	355
	2012-08	324
	2012-09	528
	2012-10	391
Chicken-2.1.1/README.txt	2010-10	0
	2010-11	0
	2010-12	0
	2011-01	0
	2011-02	16
	2011-03	20
	2011-04	16
	2011-05	18
	2011-06	13
	2011-07	22
	2011-08	18
	2011-09	19
	2011-10	20
	2011-11	15
	2011-12	30
	2012-01	37
	2012-02	21
	2012-03	17
	2012-04	5
	2012-05	12
	2012-06	6
	2012-07	8
	2012-08	2
	2012-09	24
	2012-10	19
Chicken-2.1.1/Chicken-2.1.1.dmg	2010-10	0
	2010-11	0
	2010-12	0
	2011-01	0
	2011-02	1,502
	2011-03	1,207
	2011-04	866
	2011-05	1,052
	2011-06	1,100
	2011-07	1,590

	2011-08	640
	2011-09	345
	2011-10	467
	2011-11	482
	2011-12	402
	2012-01	549
	2012-02	484
	2012-03	419
	2012-04	318
	2012-05	289
	2012-06	353
	2012-07	347
	2012-08	322
	2012-09	504
	2012-10	372
Chicken-2.1	2010-10	0
	2010-11	0
	2010-12	0
	2011-01	1,817
	2011-02	392
	2011-03	12
	2011-04	5
	2011-05	3
	2011-06	5
	2011-07	7
	2011-08	13
	2011-09	7
	2011-10	6
	2011-11	5
	2011-12	7
	2012-01	2
	2012-02	8
	2012-03	2
	2012-04	1
	2012-05	3
	2012-06	5
	2012-07	4
	2012-08	0
	2012-09	6

	2012-10	7
Chicken-2.1/README.txt	2010-10	0
	2010-11	0
	2010-12	0
	2011-01	10
	2011-02	5
	2011-03	0
	2011-04	1
	2011-05	0
	2011-06	1
	2011-07	0
	2011-08	1
	2011-09	0
	2011-10	0
	2011-11	0
	2011-12	0
	2012-01	0
	2012-02	0
	2012-03	1
	2012-04	1
	2012-05	0
	2012-06	0
	2012-07	0
	2012-08	0
	2012-09	0
	2012-10	1
Chicken-2.1/Chicken-2.1.dmg	2010-10	0
	2010-11	0
	2010-12	0
	2011-01	1,807
	2011-02	387
	2011-03	12
	2011-04	4
	2011-05	3
	2011-06	4
	2011-07	7
	2011-08	12
	2011-09	7
	2011-10	6

	2011-11	5
	2011-12	7
	2012-01	2
	2012-02	8
	2012-03	1
	2012-04	0
	2012-05	3
	2012-06	5
	2012-07	4
	2012-08	0
	2012-09	6
	2012-10	6
Chicken-2.1b2	2010-10	0
	2010-11	223
	2010-12	38
	2011-01	5
	2011-02	3
	2011-03	0
	2011-04	0
	2011-05	0
	2011-06	2
	2011-07	0
	2011-08	1
	2011-09	1
	2011-10	2
	2011-11	1
	2011-12	3
	2012-01	0
	2012-02	0
	2012-03	0
	2012-04	1
	2012-05	0
	2012-06	0
	2012-07	0
	2012-08	1
	2012-09	1
	2012-10	3
Chicken-2.1b2/README.txt	2010-10	0
	2010-11	0

	2010-12	0
	2011-01	3
	2011-02	0
	2011-03	0
	2011-04	0
	2011-05	0
	2011-06	1
	2011-07	0
	2011-08	0
	2011-09	0
	2011-10	0
	2011-11	0
	2011-12	2
	2012-01	0
	2012-02	0
	2012-03	0
	2012-04	1
	2012-05	0
	2012-06	0
	2012-07	0
	2012-08	0
	2012-09	0
	2012-10	2
Chicken-2.1b2/Chicken-2.1b2.dmg	2010-10	0
	2010-11	223
	2010-12	38
	2011-01	2
	2011-02	3
	2011-03	0
	2011-04	0
	2011-05	0
	2011-06	1
	2011-07	0
	2011-08	1
	2011-09	1
	2011-10	2
	2011-11	1
	2011-12	1
	2012-01	0

	2012-02	0
	2012-03	0
	2012-04	0
	2012-05	0
	2012-06	0
	2012-07	0
	2012-08	1
	2012-09	1
	2012-10	1
Chicken-2.1b3	2010-10	0
	2010-11	0
	2010-12	1,051
	2011-01	225
	2011-02	0
	2011-03	1
	2011-04	1
	2011-05	0
	2011-06	2
	2011-07	0
	2011-08	3
	2011-09	2
	2011-10	1
	2011-11	1
	2011-12	1
	2012-01	6
	2012-02	2
	2012-03	1
	2012-04	0
	2012-05	0
	2012-06	1
	2012-07	1
	2012-08	1
	2012-09	0
	2012-10	2
Chicken-2.1b3/README.txt	2010-10	0
	2010-11	0
	2010-12	1
	2011-01	3
	2011-02	0

	2011-03	0
	2011-04	0
	2011-05	0
	2011-06	1
	2011-07	0
	2011-08	0
	2011-09	0
	2011-10	0
	2011-11	0
	2011-12	1
	2012-01	0
	2012-02	1
	2012-03	0
	2012-04	0
	2012-05	0
	2012-06	0
	2012-07	0
	2012-08	0
	2012-09	0
	2012-10	0
Chicken-2.1b3/Chicken-2.1b3.dmg	2010-10	0
	2010-11	0
	2010-12	1,050
	2011-01	222
	2011-02	0
	2011-03	1
	2011-04	1
	2011-05	0
	2011-06	1
	2011-07	0
	2011-08	3
	2011-09	2
	2011-10	1
	2011-11	1
	2011-12	0
	2012-01	6
	2012-02	1
	2012-03	1
	2012-04	0

	2012-05	0
	2012-06	1
	2012-07	1
	2012-08	1
	2012-09	0
	2012-10	2
Chicken-2.1b1	2010-10	32
	2010-11	23
	2010-12	0
	2011-01	2
	2011-02	0
	2011-03	0
	2011-04	0
	2011-05	0
	2011-06	1
	2011-07	1
	2011-08	1
	2011-09	1
	2011-10	1
	2011-11	0
	2011-12	0
	2012-01	0
	2012-02	1
	2012-03	0
	2012-04	0
	2012-05	0
	2012-06	0
	2012-07	0
	2012-08	0
	2012-09	2
	2012-10	0
Chicken-2.1b1/Chicken-2.1b1.dmg	2010-10	32
	2010-11	23
	2010-12	0
	2011-01	2
	2011-02	0
	2011-03	0
	2011-04	0
	2011-05	0

	2011-06	1
	2011-07	1
	2011-08	1
	2011-09	1
	2011-10	1
	2011-11	0
	2011-12	0
	2012-01	0
	2012-02	1
	2012-03	0
	2012-04	0
	2012-05	0
	2012-06	0
	2012-07	0
	2012-08	0
	2012-09	2
	2012-10	0
Chicken-2.2b2.dmg	2010-10	0
	2010-11	0
	2010-12	0
	2011-01	0
	2011-02	0
	2011-03	0
	2011-04	0
	2011-05	0
	2011-06	0
	2011-07	0
	2011-08	0
	2011-09	0
	2011-10	0
	2011-11	2,465
	2011-12	3,525
	2012-01	3,993
	2012-02	3,070
	2012-03	2,263
	2012-04	1,990
	2012-05	1,611
	2012-06	1,931
	2012-07	1,929

	2012-08	2,499
	2012-09	6,243
	2012-10	5,515
Chicken-2.2b1.dmg	2010-10	0
	2010-11	0
	2010-12	0
	2011-01	0
	2011-02	0
	2011-03	0
	2011-04	0
	2011-05	0
	2011-06	0
	2011-07	0
	2011-08	2,300
	2011-09	1,745
	2011-10	1,965
	2011-11	1,929
	2011-12	163
	2012-01	248
	2012-02	420
	2012-03	133
	2012-04	7
	2012-05	19
	2012-06	10
	2012-07	14
	2012-08	31
	2012-09	11
	2012-10	9
README.txt	2010-10	0
	2010-11	0
	2010-12	75
	2011-01	26
	2011-02	16
	2011-03	15
	2011-04	12
	2011-05	2
	2011-06	13
	2011-07	13
	2011-08	25

	2011-09	33
	2011-10	34
	2011-11	67
	2011-12	41
	2012-01	70
	2012-02	40
	2012-03	32
	2012-04	24
	2012-05	17
	2012-06	15
	2012-07	13
	2012-08	15
	2012-09	37
	2012-10	49

Bugs

Date	Bugs Open	Bugs Closed	Bugs Deleted	Bugs Pending
2008-01	0	0	0	0
2008-02	0	0	0	0
2008-03	0	0	0	0
2008-04	0	0	0	0
2008-05	0	0	0	0
2008-06	0	0	0	0
2008-07	0	0	0	0
2008-08	0	0	0	0
2008-09	0	0	0	0
2008-10	0	0	0	0
2008-11	0	0	0	0
2008-12	0	0	0	0
2009-01	0	0	0	0
2009-02	0	0	0	0
2009-03	0	0	0	0
2009-04	0	0	0	0
2009-05	0	0	0	0
2009-06	0	0	0	0
2009-07	0	0	0	0
2009-08	0	0	0	0

2009-09	0	0	0	0
2009-10	0	0	0	0
2009-11	0	0	0	0
2009-12	0	0	0	0
2010-01	0	0	0	0
2010-02	0	0	0	0
2010-03	0	0	0	0
2010-04	0	0	0	0
2010-05	0	0	0	0
2010-06	0	0	0	0
2010-07	0	0	0	0
2010-08	0	0	0	0
2010-09	0	0	0	0
2010-10	0	0	0	0
2010-11	0	0	0	0
2010-12	0	1	0	0
2011-01	0	0	0	0
2011-02	1	0	0	0
2011-03	0	0	0	0
2011-04	1	0	0	0
2011-05	0	2	0	0
2011-06	1	0	0	0
2011-07	0	0	0	0
2011-08	0	1	0	0
2011-09	0	1	0	0
2011-10	1	0	0	0
2011-11	1	0	0	0
2011-12	1	0	0	0
2012-01	2	0	0	0
2012-02	0	1	0	0
2012-03	1	0	0	0
2012-04	0	0	0	0
2012-05	1	0	0	0
2012-06	0	0	0	0
2012-07	1	0	0	0
2012-08	0	0	0	0
2012-09	1	0	0	0
2012-10	1	0	0	0

Feature Requests

Date	Feature Requests Open	Feature Requests Closed	Feature Requests Deleted	Feature Requests Pending
2008-01	0	0	0	0
2008-02	0	0	0	0
2008-03	0	0	0	0
2008-04	0	0	0	0
2008-05	0	0	0	0
2008-06	0	0	0	0
2008-07	0	0	0	0
2008-08	0	0	0	0
2008-09	0	0	0	0
2008-10	0	0	0	0
2008-11	0	0	0	0
2008-12	0	0	0	0
2009-01	0	0	0	0
2009-02	0	0	0	0
2009-03	0	0	0	0
2009-04	0	0	0	0
2009-05	0	0	0	0
2009-06	0	0	0	0
2009-07	0	0	0	0
2009-08	0	0	0	0
2009-09	0	0	0	0
2009-10	0	0	0	0
2009-11	0	0	0	0
2009-12	0	0	0	0
2010-01	0	0	0	0
2010-02	0	0	0	0
2010-03	0	0	0	0
2010-04	0	0	0	0
2010-05	0	0	0	0
2010-06	0	0	0	0
2010-07	0	0	0	0
2010-08	0	0	0	0
2010-09	0	0	0	0
2010-10	0	0	0	0
2010-11	0	0	0	0

2010-12	0	0	0	0
2011-01	0	0	0	0
2011-02	0	0	0	0
2011-03	0	1	0	0
2011-04	0	0	0	0
2011-05	0	0	0	0
2011-06	0	0	0	0
2011-07	0	0	0	0
2011-08	0	0	0	0
2011-09	1	0	0	0
2011-10	0	0	0	0
2011-11	1	0	0	0
2011-12	1	0	0	0
2012-01	1	0	0	0
2012-02	1	0	0	0
2012-03	0	0	0	0
2012-04	0	0	0	0
2012-05	0	1	0	0
2012-06	0	0	0	0
2012-07	2	0	0	0
2012-08	1	0	0	0
2012-09	1	0	0	0
2012-10	0	0	0	0

Patches

Date	Patches Open	Patches Closed	Patches Deleted	Patches Pending
2008-01	0	0	0	0
2008-02	0	0	0	0
2008-03	0	0	0	0
2008-04	0	0	0	0
2008-05	0	0	0	0
2008-06	0	0	0	0
2008-07	0	0	0	0
2008-08	0	0	0	0
2008-09	0	0	0	0
2008-10	0	0	0	0
2008-11	0	0	0	0

2008-12	0	0	0	0
2009-01	0	0	0	0
2009-02	0	0	0	0
2009-03	0	0	0	0
2009-04	0	0	0	0
2009-05	0	0	0	0
2009-06	0	0	0	0
2009-07	0	0	0	0
2009-08	0	0	0	0
2009-09	0	0	0	0
2009-10	0	0	0	0
2009-11	0	0	0	0
2009-12	0	0	0	0
2010-01	0	0	0	0
2010-02	0	0	0	0
2010-03	0	0	0	0
2010-04	0	0	0	0
2010-05	0	0	0	0
2010-06	0	0	0	0
2010-07	0	0	0	0
2010-08	0	0	0	0
2010-09	0	0	0	0
2010-10	0	0	0	0
2010-11	0	0	0	0
2010-12	0	0	0	0
2011-01	0	0	0	0
2011-02	0	0	0	0
2011-03	0	0	0	0
2011-04	0	0	0	0
2011-05	0	0	0	0
2011-06	0	0	0	0
2011-07	0	0	0	0
2011-08	0	0	0	0
2011-09	0	0	0	0
2011-10	0	0	0	0
2011-11	0	0	0	0
2011-12	0	0	0	0
2012-01	0	0	0	0
2012-02	0	0	0	0

2012-03	0	0	0	0
2012-04	0	0	0	0
2012-05	0	0	0	0
2012-06	0	0	0	0
2012-07	0	0	0	0
2012-08	0	0	0	0
2012-09	0	0	0	0
2012-10	0	0	0	0

Support Requests

Date	Support Requests Open	Support Requests Closed	Support Requests Deleted	Support Requests Pending
2008-01	0	0	0	0
2008-02	0	0	0	0
2008-03	0	0	0	0
2008-04	0	0	0	0
2008-05	0	0	0	0
2008-06	0	0	0	0
2008-07	0	0	0	0
2008-08	0	0	0	0
2008-09	0	0	0	0
2008-10	0	0	0	0
2008-11	0	0	0	0
2008-12	0	0	0	0
2009-01	0	0	0	0
2009-02	0	0	0	0
2009-03	0	0	0	0
2009-04	0	0	0	0
2009-05	0	0	0	0
2009-06	0	0	0	0
2009-07	0	0	0	0
2009-08	0	0	0	0
2009-09	0	0	0	0
2009-10	0	0	0	0
2009-11	0	0	0	0
2009-12	0	0	0	0

2010-01	0	0	0	0
2010-02	0	0	0	0
2010-03	0	0	0	0
2010-04	0	0	0	0
2010-05	0	0	0	0
2010-06	0	0	0	0
2010-07	0	0	0	0
2010-08	0	0	0	0
2010-09	0	0	0	0
2010-10	0	0	0	0
2010-11	0	0	0	0
2010-12	0	0	0	0
2011-01	0	0	0	0
2011-02	0	0	0	0
2011-03	0	0	0	0
2011-04	0	0	0	0
2011-05	0	0	0	0
2011-06	0	0	0	0
2011-07	0	0	0	0
2011-08	0	0	0	0
2011-09	0	0	0	0
2011-10	0	0	0	0
2011-11	0	0	0	0
2011-12	0	0	0	0
2012-01	0	0	0	0
2012-02	0	0	0	0
2012-03	1	1	0	0
2012-04	0	0	0	0
2012-05	0	0	0	0
2012-06	0	0	0	0
2012-07	0	0	0	0
2012-08	0	0	0	0
2012-09	0	0	0	0
2012-10	0	0	0	0