Αναγνώριση Προτύπων

Εργασία μαθήματος – Περιγραφή Προβλήματος

1. Εισαγωγή

Τα τελευταία χρόνια, η συνηθέστερη πρακτική ανάπτυξης λογισμικού περιλαμβάνει τη χρήση online αποθετηρίων (π.χ. GitHub / GitLab / BitBucket / AzureDevops), τα οποία έχουν πολλά έργα λογισμικού που χαρακτηρίζονται από την ελεύθερη διάθεσή τους σε όλους τους ενδιαφερόμενους προγραμματιστές και ονομάζονται "έργα ανοικτού λογισμικού" (open source software projects). Τα έργα αυτά, τα οποία χρήζουν ευρείας αποδοχής και χρησιμοποιούνται από εκατομμύρια έργα, αναπτύσσονται με έναν συνεργατικό τρόπο, όπου κάθε προγραμματιστής είναι ελεύθερος να συνδράμει με τον τρόπο και στον βαθμό που το επιθυμεί στην περαιτέρω ανάπτυξη του έργου. Λαμβάνοντας υπόψη το μέγεθος και την πολυπλοκότητα των ανοιχτών έργων λογισμικού είναι εμφανές ότι η ανάπτυξή τους περιλαμβάνει μια σειρά από προκλήσεις που εντοπίζονται κυρίως στη δημιουργία μιας δομημένης διαδικασίας ανάπτυξης λογισμικού όπου εκατοντάδες (ή και χιλιάδες) προγραμματιστές θα είναι σε θέση να οργανώσουν τη δουλειά τους για την από κοινού ανάπτυξη του έργου. Εξαιτίας των προκλήσεων αυτών, τα έργα ανοιχτού λογισμικού αποτελούν μια ιδιαίτερα χρήσιμη πηγή πληροφορίας που μπορεί να αξιοποιηθεί για τη βελτίωση της διαδικασίας ανάπτυξης λογισμικού χρησιμοποιώντας δεδομένα που εστιάζουν στους παρακάτω άξονες:

- **Contributors information**
- **Repositories information**
- **Commits information**
- **Issues information**
- **Comments information**
- **Events information**

2. Εργασία

2.1. Το Πρόβλημα

Η παρούσα εργασία αναφέρεται στην αξιοποίηση της ολοένα και αυξανόμενης πληροφορίας που εμπεριέχεται στα έργα ανοικτού λογισμικού, αλλά και στα αποθετήρια που αυτά χρησιμοποιούν. Μία από τις πιο δημοφιλείς πλατφόρμες φιλοξενίας κώδικα και ελέγχου εκδόσεων, το GitHub, παρέχει μία ελεύθερη διεπαφή προγραμματισμού εφαρμογών (Application Programming Interface – API[1]), μέσω του οποίου είναι δυνατή η προσπέλαση όλης της πληροφορίας που εμπεριέχεται στην πλατφόρμα και αφορά τα αποθετήρια ανοικτού λογισμικού.

Η επικοινωνία με το API μπορεί να γίνει εύκολα μέσω οποιασδήποτε γλώσσας προγραμματισμού (χαρακτηριστικά παραδείγματα εμπεριέχονται στον συνοδευτικό φάκελο "*library*" για την γλώσσα Python)

[1] https://docs.github.com/en/rest/guides/getting-started-with-the-rest-api

και τα αποτελέσματα να επεξεργαστούν και να χρησιμοποιηθούν για τη δημιουργία και την απάντηση των ερευνητικών σας ερωτημάτων.

Πιο συγκεκριμένα, το βασικό ζητούμενο του προβλήματος προς επίλυση αποτελεί η διατύπωση και η ενασχόληση με ένα ή περισσότερα ερευνητικά ερωτήματα, αναφορικά με τις πληροφορίες που εμπεριέχονται σε αποθετήρια ανοικτού λογισμικού και αφορούν τους προγραμματιστές που εμπλέκονται με αυτά, αλλά και τα ίδια τα χαρακτηριστικά των αποθετηρίων. Χρήσιμες πηγές πληροφορίας αποτελούν χαρακτηριστικά των αποθετηρίων όπως η εμπλοκή των προγραμματιστών σε ένα αποθετήριο (contributors), η εμπλοκή των χρηστών σε θέματα που απασχολούν το αποθετήριο (issues), ο αριθμός των σχολίων σε κάθε issue του αποθετηρίου (comments), ο αριθμός των αλλαγών που έχουν πραγματοποιηθεί (commits), οι θεματικές ενότητες και οι γλώσσες προγραμματισμού που καλύπτει το αποθετήριο (topics - languages) κ.ο.κ. Μερικά ενδεικτικά ερευνητικά ερωτήματα δίνονται παρακάτω (μπορείτε να χρησιμοποιήσετε αυτά ή να διατυπώσετε τα δικά σας):

- **Εύρεση του ρόλου που κατέχει ένας χρήστης εντός του αποθετηρίου**

Ένας χρήστης μπορεί να συνεισφέρει στην ανάπτυξη ενός αποθετηρίου με πολλούς και διαφορετικούς τρόπους. Η πιο γνωστή κατηγοριοποίηση των χρηστών ενός αποθετηρίου περιλαμβάνει τρεις διαφορετικές ομάδες: τους **Dev** χρήστες, οι οποίοι συνεισφέρουν κυρίως στην ανάπτυξη του κώδικα του αποθετηρίου, τους **Ops** χρήστες, οι οποίοι συνεισφέρουν κατά βάση στον έλεγχο, αποσφαλμάτωση, εντοπισμό σφαλμάτων και περαιτέρω προώθησης του έργου και, τέλος, στους **DevOps**, που συνεισφέρουν και στις δύο παραπάνω κατηγορίες.

Ένα χρήσιμο ερευνητικό ερώτημα αποτελεί η εξακρίβωση του ρόλου στον οποίο ανήκει κάθε χρήστης του αποθετηρίου. Για τον σκοπό αυτό, μπορούν να χρησιμοποιηθούν χρήσιμες πληροφορίες που βρίσκονται εντός του αποθετηρίου, όπως ο αριθμός των commits που έχει πραγματοποιήσει ο χρήστης, ο αριθμός των issues που έχει ανοίξει/κλείσει/σχολιάσει ο χρήστης, το πλήθος των σχολίων του χρήστη κ.ο.κ.

- **Αξιολόγηση της δυσκολίας αντιμετώπισης των θεμάτων ενός αποθετηρίου**

Τα issues κάθε αποθετηρίου αποτελούν θέματα που έχουν εντοπιστεί στο έργο και πρέπει να επιλυθούν. Τα θέματα αυτά μπορεί να αφορούν κακή λειτουργία του κώδικα (errors/bugs), νέες προσθήκες στον κώδικα και τη λειτουργικότητα του έργου, γενικά θέματα προς συζήτηση, απορίες χρηστών κ.ο.κ. Οι χρήστες που είναι υπεύθυνοι για το συγκεκριμένο έργο οφείλουν να επιλύουν γρήγορα και σωστά τα θέματα που εμφανίζονται, έτσι ώστε το αποθετήριο να διατηρείται σε μια ιδανική κατάσταση και οι χρήστες του να είναι ικανοποιημένοι.

Στο παραπάνω πλαίσιο, είναι ιδιαίτερα σημαντική η αξιολόγηση της δυσκολίας που εμφανίζει κάθε νέο issue ως προς την επίλυσή του, καθώς μπορεί να βοηθήσει αισθητά τους υπεύθυνους του έργου στο να προτεραιοποιήσουν τα θέματα που πρέπει να επιλύσουν, αλλά και να δώσουν την ανάλογη προσοχή. Στο ερώτημα αυτό, πρέπει να δοθεί ιδιαίτερη έμφαση σε χαρακτηριστικά όπως η χρονική διάρκεια που απαιτήθηκε για να κλείσει το issue, ο αριθμός των σχολίων που εμπλέκονται σε αυτό, ο αριθμός των commits που αναφέρονται στο issue κ.ο.κ.

Πίνακας 1 Παράδειγμα εύρεσης βαθμού δυσκολίας με βάση συγκεκριμένα χαρακτηριστικά

Clusters	Feature				Difficulty
	# Commits	Duration	# Comments	# Users Involved	
#1	Low	Small	Low	High	**Low**
#2	Medium	Large	High	Low	**Medium**
#3	High	Large	High	Medium	**High**

Στον παραπάνω πίνακα φαίνεται το αποτέλεσμα της διενέργειας clustering με τη χρήση 4 χαρακτηριστικών, για την εύρεση issues που παρουσιάζουν κοινή συμπεριφορά ως προς τα χαρακτηριστικά που υποδεικνύουν τη δυσκολία αντιμετώπισής τους. Από τα clusters που σχηματίστηκαν και από την εξέταση των χαρακτηριστικών που ανήκουν σε καθένα από αυτά, ταυτοποιήθηκε ο βαθμός δυσκολίας που αυτά παρουσιάζουν. Όπως φαίνεται και από τον πίνακα, στο πρώτο cluster εμφανίζονται τα issues με χαμηλό βαθμό δυσκολίας, ενώ αντίθετα οι μέθοδοι με πολλά commits και comments και μεγάλη διάρκεια χαρακτηρίζονται από υψηλό βαθμό δυσκολίας.

- **Αξιολόγηση της φάσης ανάπτυξης του έργου**

Ένα ακόμη ερώτημα που μπορεί να αποδειχθεί ιδιαίτερα σημαντικό, τόσο για τους άμεσα εμπλεκόμενους στο έργο λογισμικού, όσο και για τους project leaders που θέλουν να παρακολουθούν την εξέλιξη του έργου είναι η αξιολόγηση και η εξακρίβωση της φάσης ανάπτυξης στην οποία βρίσκεται το έργο σε μία δεδομένη χρονική στιγμή. Ένα παράδειγμα τέτοιου διαχωρισμού αποτελούν τα στάδια ανάπτυξης (active development) και συντήρησης (maintenance) του έργου. Για τον σκοπό αυτό, μπορούν να χρησιμοποιηθούν χαρακτηριστικά όπως ο αριθμός των commits που γίνονται στο αποθετήριο ανά εβδομάδα, ο αριθμός των γραμμών που μεταβάλλεται σε κάθε commit, ο αριθμός των νέων issues που δημιουργούνται ανά εβδομάδα, το ποσοστό των issues που αφορούν διορθώσεις (errors/bugs) ή ανάπτυξη νέων χαρακτηριστικών κ.ο.κ.

- **Εντοπισμός του κατάλληλου προγραμματιστή ανάλογα με τη θεματολογία του έργου**

Καθώς κάθε προγραμματιστής διαφέρει ως προς τα πεδία ενδιαφέροντός του, τις βιβλιοθήκες/γλώσσες προγραμματισμού που γνωρίζει και τα έργα στα οποία έχει συμμετάσχει, αποτελεί ιδιαίτερη πρόκληση η αναγνώριση των πιο κατάλληλων προγραμματιστών που μπορούν να συνεισφέρουν σημαντικά σε ένα νέο έργο λογισμικού. Σημαντικά χαρακτηριστικά για την επίτευξη αυτού του στόχου μπορούν να αποδειχθούν τα topics που χαρακτηρίζουν τα έργα λογισμικού στα οποία έχει εμπλακεί ένας προγραμματιστής, ο αριθμός των commits που πραγματοποίησε σε αυτά τα έργα μαζί με τον αριθμό των γραμμών που τροποποίησε κ.ο.κ.

- **Διαχωρισμός των προγραμματιστών με βάση τα χαρακτηριστικά τους**

Σε συνέχεια του παραπάνω ερευνητικού ερωτήματος, πολλές πληροφορίες που αφορούν τις δραστηριότητες ενός προγραμματιστή μπορούν να χρησιμοποιηθούν, με σκοπό τον διαχωρισμό των προγραμματιστών σε ομάδες, ανάλογα με τα χαρακτηριστικά τους, χρησιμοποιώντας δεδομένα όπως τα έργα που έχουν συμμετάσχει και η θεματολογία αυτών, ο αριθμός των issues που έχουν ανοίξει/κλείσει/σχολιάσει κ.ο.κ.

- **Διαχωρισμός των αποθετηρίων με βάση τα χαρακτηριστικά τους**

Τέλος, ένα ενδιαφέρον ερευνητικό ερώτημα αποτελεί η εύρεση παρόμοιων αποθετηρίων και έργων λογισμικού που εμφανίζουν κοινά χαρακτηριστικά. Εκτός από τη θεματολογία κάθε αποθετηρίου, όπως αυτή εκφράζεται στα topics του, σημαντικό ρόλο μπορεί να παίξουν ο αριθμός των commits και issues, ο αριθμός των contributors, καθώς και οι γλώσσες προγραμματισμού που εμφανίζονται σε αυτό.