

TOOLS AND METHODS FOR LARGE SCALE EMPIRICAL SOFTWARE ENGINEERING RESEARCH

THESIS ABSTRACT

by
Georgios I. Gousios
Department of Management Science and Technology
Athens University of Economics and Business

Abstract

Software engineering is concerned with the study of systematic approaches towards software development and maintenance. According to many authors, software engineering is an empirical science as it strives to produce models that capture the characteristics of the development process or to predict its behaviour. Being an empirical science, software engineering is in a constant need for data.

The emergence of the Open Source Software (OSS) movement has provided software engineering researchers with rich process and product data. OSS projects make their source configuration management, mailing lists and issue tracking database systems publicly available. Although they are free to use, OSS data come with a cost for the researcher. During a lifetime spanning multiple decades, several OSS projects have amassed gigabytes of data worth studying. The computational cost for processing such large volumes of data is not trivial and lays beyond the capabilities of single workstation setups. Moreover, each project uses its own combination of the aforementioned and other project management systems management tools, such as Wikis and documentation generators. Without the appropriate abstractions, it is challenging to build tools that can process data from various projects at the same time.

In the recent years, software engineering research benefited from the availability of OSS repositories and a new stream of research that takes advantage of the rich process data residing in those repositories emerged. To evaluate the extend and use of OSS data in empirical software engineering studies, we conducted a systematic literature review. Specifically, we constructed a classification framework which we then applied on 70 randomly selected studies published in various software engineering publication outlets from 2003 onwards. The classification provided interesting insights:

- Studies are being performed almost exclusively on data originating from OSS projects.
- The vast majority of studies use data from less than 5 projects.
- There is no cross validation of the results of published works.

We attribute the obtained results to the inherent complexity of experimenting with OSS data. To remedy the situation, we propose performing large scale software engineering research studies on an integrated platform, that combines easy to use and extend tools and readily analysed data. Drawing from the experiences of other mature empirical fields, we believe that shared research infrastructures are crucial for advancing the state of the art in research, as they enable rigourous evaluation, experiment replication, sharing tool, results and raw data and, more importantly, allow researchers to focus on their research questions instead of spending time to re-implement tools or pre-process data.

In this thesis, we investigate novel ways to integrate process and product data from various OSS repositories in an effort to build an open Software Engineering Research Platform (SERP) consisting of both software tools and shared data. We base our work on SQO-OSS, a tool designed to perform software quality analysis. We analyse the design of the raw data and meta-data storage formats and as part of its implementation, we develop novel solutions to the problems of: (i) representing distributed and centralised source configuration management data in relational format (ii) identifying and resolving developer identities across data sources, and (iii) efficiently representing and distributing processing workload towards fully exploiting the available hardware.

To demonstrate the validity of our approach, and the effectiveness of the proposed platform in conducting large scale empirical research, we perform two case studies using it. The first one examines the effect of intense email discussions on the short-term evolution of a project. The hypothesis under investigation is that since OSS projects have limited human resources, intense discussions on mailing lists will have a measurable effect on the source code line intake rate. After examining the characteristics of intense communications, we construct a model to calculate the effect of discussions and implemented it as an extension to our platform. We run the study on about 70 projects and we find that there is no clear impact of intense discussions in short term evolution.

In the second case, we correlate maintainability metrics with key development process characteristics to study their effect on project maintenance. Specifically, we study whether the number of developers that have worked on a project or on a specific source module is indicative of how maintainable the project as a whole or the module is. Using the SERP platform, we run the study on 210 C and Java projects. We find no correlation between the number of developers that have worked on a project or a source code module and its maintainability, as this is measured by the maintainability index metric.

One of our findings is that in both case studies, the application of bias on the selection of the examined sample, would lead to completely different results. In fact, we show that there are more hypothesis validating cases (even though the hypotheses have been overall invalidated) for each case study than the average number of cases evaluated per case study in currently published studies, which we derived from the systematic literature review. We consider this result as a strong indication of the value of large scale experimentation we advocate in this thesis.

Overall, our contribution has both a scientific and a practical aspect. More specifically:

- We describe a framework for classifying empirical software engineering research works and we use it to analyse the shortcomings of the current state of the art.
- We analyse the requirements and describe the design of a platform for large scale empirical software engineering studies.

- We introduce a relational schema for storing metadata from software repositories, which provides our platform with enough abstractions to retrieve software process metadata across projects and across software repositories.
- We introduce an algorithm for mapping semi-structured data from software configuration management repositories in a relational format.
- We introduce algorithms for resolving developer identities across data sources and for distributing the load of computation across nodes in a cluster environment.
- We validate our platform by conducting two case studies using it. We find that intense email discussions do not affect short term project evolution and that development team size does not affect software maintainability at the module or project level.
- We show that the results of the aforementioned case studies could be radically different if bias is applied on the selected experimentation dataset, thereby validating our thesis on the importance of conducting experiments on large scale datasets.

Finally, we make the software we developed and the data we produced available to the research community under non-restrictive licenses.

Περίληψη

Ο τομέας της μηχανικής λογισμικού ασχολείται με τη συστηματική μελέτη των διαδικασιών ανάπτυξης και συντήρησης λογισμικού. Σύμφωνα με πολλούς συγγραφείς, η μηχανική λογισμικού είναι μια εμπειρική επιστήμη, καθώς προσπαθεί να δημιουργήσει μοντέλα που εξηγούν τα χαρακτηριστικά της διαδικασίας ανάπτυξης ή/και τα προβλέπουν. Ως εμπειρική επιστήμη, η μηχανική λογισμικού χρειάζεται δεδομένα για την διεξαγωγή ερευνών.

Το κίνημα Ελεύθερου Λογισμικού / Λογισμικού Ανοιχτού Κώδικα (ΕΛΛΑΚ) έχει καταστήσει προσβάσιμα στους ερευνητές μηχανικούς λογισμικού πλούσια δεδομένα που αφορούν τη διαδικασία ανάπτυξης και το ίδιο το λογισμικό για διάφορα έργα. Τα έργα ΕΛΛΑΚ παρέχουν πλήρη πρόσβαση σε δεδομένα του συστήματος ελέγχου εκδόσεων, των λιστών ηλεκτρονικού ταχυδρομείου και της βάσεων δεδομένων λαθών. Αν και τα δεδομένα είναι ανοικτά στην έρευνα, η χρήση τους έχει μεγάλο κόστος για τους ερευνητές. Κατά τη διάρκεια ζωής των έργων, που σε μερικές περιπτώσεις επεκτείνεται σε παραπάνω από μια δεκαετίες, τα έργα ΕΛΛΑΚ έχουν συσσωρεύσει μεγάλους όγκους δεδομένων που αξίζει να μελετηθούν. Το υπολογιστικό κόστος για την επεξεργασία τόσο μεγάλων όγκων δεδομένων σε συνδυασμό με την πολυπλοκότητα των μεθόδων ανάλυσης καθιστούν ασύμφορη χρονικά ή αδύνατη την εκτέλεση μεγάλων πειραμάτων στους τυπικούς υπολογιστές. Επιπλέον, κάθε έργο ΕΛΛΑΚ χρησιμοποιεί διαφορετικούς συνδυασμούς των εργαλείων διαχείρισης που προαναφέρθηκαν. Χωρίς τις κατάλληλες αφαιρετικές δομές, είναι δύσκολο να κατασκευαστούν εργαλεία που μα μπορούν να επεξεργαστούν δεδομένα προερχόμενα από διαφορετικά έργα.

Πρόσφατα, η έρευνα στην τεχνολογία λογισμικού ωφελήθηκε από την διαθεσιμότητα των αποθετηρίων ΕΛΛΑΚ, και σαν συνέπεια δημιουργήθηκε μια καινούρια ερευνητική κατεύθυνση που εκμεταλλεύεται τα πλούσια δεδομένα διαδικασίας και προϊόντος που υπάρχουν σε αυτά. Για να εκτιμήσουμε την έκταση και την χρήση δεδομένων που προέρχονται από έργα ΕΛΛΑΚ στην έρευνα, μελετήσαμε συστηματικά τη βιβλιογραφία. Συγκεκριμένα, κατασκευάσαμε ένα πλαίσιο ταξινόμησης με τη βοήθεια του οποίου ταξινομήσαμε 70 τυχαία επιλεγμένες δημοσιεύσεις που έγιναν από το 2003 και έπειτα. Η ταξινόμηση μας παρείχε χρήσιμες πληροφορίες:

- Οι τρέχουσες μελέτες γίνονται σχεδόν αποκλειστικά με δεδομένα από έργα ΕΛΛΑΚ.
- Η μεγάλη πλειοψηφία των μελετών χρησιμοποιούν δεδομένα από λιγότερα των 5 έργα.
- Δεν βρήκαμε περιπτώσεις επαναληπτικής επαλήθευσης μελετών

Αποδίδουμε τα παραπάνω αποτελέσματα στην εγγενή πολυπλοκότητα του πειραματισμού με δεδομένα που προέρχονται από έργα ΕΛΛΑΚ. Για την αντιμετώπιση της κατάστασης, προτείνουμε την διενέργεια μεγάλης κλίμακας μελετών σε μία ολοκληρωμένη πλατφόρμα που συνδυάζει εύκολα στη χρήση και επέκταση εργαλεία και δεδομένα που έχουν ήδη αναλυθεί. Βασιζόμενοι στην εμπειρία

άλλων, ώριμων, εμπειρικών πεδίων, πιστεύουμε ότι οι διαμοιραζόμενες υποδομές έρευνας είναι ζωτικής σημασίας για την προώθηση της έρευνας, καθώς επιτρέπουν την αυστηρή αξιολόγηση, την ανταλλαγή δεδομένων, εργαλείων και αποτελεσμάτων και, ποιο σημαντικά, επιτρέπουν στους ερευνητές να εστιάσουν τις προσπάθειές τους στην απάντηση ερευνητικών ερωτημάτων αντί στην ανάπτυξη εργαλείων ή την προ-επεξεργασία των δεδομένων.

Σε αυτή τη διατριβή, διερευνούμε καινούργιους τρόπους για να ενσωματώσουμε δεδομένα διεργασίας και προϊόντος από διάφορα αποθετήρια έργων ΕΛΛΑΚ, δημιουργώντας την ανοιχτή πλατφόρμα πειραματισμού τεχνολογίας λογισμικού SERP . Η πλατφόρμα περιλαμβάνει τόσο εργαλεία όσο και δεδομένα. Βασίζουμε τη δουλειά μας στο εργαλείο ανάλυσης ποιότητας λογισμικού SQO-OSS . Αναλύουμε το σχεδιασμό των διαμορφώσεων αρχείων για την αποθήκευση πρωτογενών δεδομένων και μεταδιδόμενων και παρουσιάζουμε τις λύσεις που δώσαμε στα προβλήματα της α) αναπαράστασης δεδομένων από κεντρικοποιημένα και διεσπαρμένα αποθετήρια λογισμικού σε σχεσιακή μορφή, β) της αναζήτησης και αντιστοίχισης ταυτοτήτων των μελών της ομάδας ανάπτυξης σε όλα τα πρωτογενή δεδομένα γ) της αποτελεσματικής αναπαράστασης φόρτου εργασίας σε μορφή που επιτρέπει την κατανομή του σε συστοιχίες υπολογιστών με σκοπό την ταχύτερη εκτέλεση πειραμάτων.

Για να αποδείξουμε την ισχύ και τη χρησιμότητα της προσέγγισής μας, και την αποτελεσματικότητα της προτεινόμενης πλατφόρμας στην έρευνα με πολλά δεδομένα, παρουσιάζουμε 2 μελέτες περίπτωσης που τη χρησιμοποιούν. Η πρώτη μελέτη εξετάζει την επίδραση των έντονων συζητήσεων που συχνά εμφανίζονται στις λίστες ηλεκτρονικού ταχυδρομείου έργων ΕΛΛΑΚ στην βραχυπρόθεσμη ανάπτυξη του έργου. Η υπόθεση που εξετάζουμε είναι ότι αφού τα έργα ανοιχτού λογισμικού έχουν περιορισμένους ανθρώπινους πόρους, οι έντονες συζητήσεις θα έχουν κάποιο μετρήσιμο αντίκτυπο στον ρυθμό παραγωγής κώδικα του έργου. Αφού εξετάσουμε τα χαρακτηριστικά των έντονων συζητήσεων, σχεδιάζουμε ένα μοντέλο που υπολογίζει την προαναφερόμενη επίδραση και το υλοποιούμε σαν επέκταση στην πλατφόρμα SERP και το εφαρμόζουμε σε δεδομένα από περίπου 70 έργα. Βρίσκουμε ότι γενικά δεν υπάρχει συγκεκριμένη συσχέτιση μεταξύ των έντονων συζητήσεων και της βραχυπρόθεσμης ανάπτυξης του έργου.

Στην δεύτερη μελέτη περίπτωσης, συσχετίζουμε τον δείκτη συντηρησιμότητας ενός έργου με χαρακτηριστικά της διαδικασίας ανάπτυξης για να δούμε αν υπάρχει κάποια επίδραση. Συγκεκριμένα, μελετάμε εάν ο αριθμός των προγραμματιστών που έχουν δουλέψει στο έργο ή σε ένα κομμάτι κώδικα του έργου είναι ενδεικτικός της συντηρησιμότητας. Χρησιμοποιώντας την πλατφόρμα SERP , αναλύουμε 210 έργα γραμμένα στις γλώσσες προγραμματισμού C και Java . Βρίσκουμε ότι δεν υπάρχει κάποιου είδους συσχέτιση μεταξύ των 2 μεταβλητών.

Ένα ενδιαφέρον εύρημα και των 2 μελετών είναι ότι αν από το συνολικό δείγμα επιλέξουμε συγκεκριμένα έργα για μελέτη, τα αποτελέσματα που θα λάβουμε θα είναι εντελώς διαφορετικά. Μάλιστα, δείχνουμε ότι στο δείγμα μας υπάρχουν περισσότερες έργα που επαληθεύουν τις υποθέσεις και των 2 μελετών (τις ίδιες που γενικά απορρίψαμε) από τον μέσο έργων που χρησιμοποιούνται στις μελέτες που αναλύσαμε στην συστηματική ανάλυση της σχετικής δουλειάς που κάναμε. Θεωρούμε ότι το αποτέλεσμα αυτό αποτελεί ισχυρή ένδειξη της

αξίας του πειραματισμού σε μεγάλη κλίμακα που υποστηρίζουμε σε αυτή τη διατριβή.

Γενικά, η συνεισφορά μας έχει επιστημονική και πρακτική αξία. Ειδικότερα:

- Περιγράφουμε ένα πλαίσιο για την ταξινόμηση των μελετών εμπειρικής τεχνολογίας λογισμικού και το χρησιμοποιούμε για να αναλύσουμε την τις αδυναμίες των ερευνών που δημοσιεύονται σήμερα
- Αναλύουμε τις απαιτήσεις και παρουσιάζουμε το σχεδιασμό μιας πλατφόρμας για μελέτες λογισμικού μεγάλης κλίμακας
- Εισάγουμε ένα σχεσιακό σχήμα για την αποθήκευση μεταδιδόμενων από αποθετήρια λογισμικού, το οποίο παρέχει στην πλατφόρμα μας ένα ικανοποιητικό αφαιρετικό μηχανισμό για να ανακτά δεδομένα διαδικασίας ανάπτυξης από πολλά έργα ταυτόχρονα.
- Εισάγουμε ένα αλγόριθμο που μετατρέπει ημί-δομημένα δεδομένα από συστήματα διαχείρισης εκδόσεων λογισμικού σε δομημένα-σχεσιακά δεδομένα.
- Εισάγουμε αποδοτικούς αλγόριθμους για την αντιστοίχιση μελών της διαδικασίας ανάπτυξης μεταξύ πηγών δεδομένων και για την κατανομή του φόρτου εργασίας στους κόμβους μιας συστοιχίας υπολογιστών.
- Αποδεικνύουμε την χρησιμότητα της πλατφόρμας μας διενεργώντας 2 μελέτες περίπτωσης χρησιμοποιώντας τη. Βρίσκουμε ότι οι έντονες συζητήσεις στις λίστες ηλεκτρονικού ταχυδρομείου δεν επηρεάζουν την βραχυπρόθεσμη ανάπτυξη ενός έργου και ότι το μέγεθος μιας ομάδας ανάπτυξης δεν επηρεάζει την συντηρησιμότητα σε επίπεδο έργου ή ενότητας λογισμικού.
- Δείχνουμε ότι τα αποτελέσματα της προαναφερόμενης μελέτης περίπτωσης μπορεί να τροποποιηθούν δραματικά αν το δείγμα επιλεγεί κατάλληλα, επικυρώνοντας έτσι τη σημασία της διεξαγωγής έρευνας σε μεγάλο όγκο δεδομένων που πρεσβεύουμε στη διατριβή μας.

Τέλος, παρέχουμε το λογισμικό που αναπτύχθηκε και τα δεδομένα που αποτελούν την πλατφόρμα SERP στην επιστημονική κοινότητα με μη περιοριστικές άδειες χρήσης.