Network Analysis of Software Repositories: Identifying Subject Matter Experts

Andrew Dittrich, Mehmet Hadi Gunes, and Sergiu Dascalu

Abstract. A software developer joining a large software project faces a steep learning curve before they are able to make real contributions. One challenge is finding the subject matter experts who can answer questions about a specific area of the software or to review changes. This is especially true of large projects with many modules and a large number of authors. In this paper, we describe a method to model a software project as a network using information mined from the project's version control repository, and demonstrate how network analysis techniques can be used to identify the key authors and subject matter experts. We investigate metrics that can be gathered using network analysis, such as which groups of authors typically work together, and how closely knit the developers are on a project. We analyze several specific projects to demonstrate the applicability of these techniques and several hundred projects to show general trends.

1 Introduction

A new developer starting on a large has a lot to learn before they can be a productive member of the team. The project contains many different modules, each of which can be complex on its own. Typically, a junior developer will turn to a more senior developer to ask questions, and to gain insight into the overall architecture of a project. However, it can be difficult to identify experts for a particular area. A good candidate to start with is the person who last modified a file in a module, but this person may have just fixed a formatting problem or a compiler warning, and might not be the best person to ask.

Identifying the most experienced author for a specific area of the project is also a problem for project managers. If a bug is found in a specific module of a large software project, then ideally, the most experienced developer in that area of the project

Andrew Dittrich · Mehmet Hadi Gunes · Sergiu Dascalu
University of Nevada, Reno
e-mail: andy.dittrich@gmail.com, {mgunes,dascalus}@cse.unr.edu

R. Menezes, A. Evsukoff, & M.C. González (Eds.): Complex Networks, SCI 424, pp. 187–198.
springerlink.com © Springer-Verlag Berlin Heidelberg 2013

should be assigned to fix it. Unfortunately, there is not an easy way to identify that individual. If the manager has been working on this project for a while, then they most likely have the experience to know who the key developer is in this area. Alternatively, they can survey the team members to find someone who is familiar with the area of the code in question.

A project manager may also be interested in how the development team works together. If each developer works on a separate part of the project, and there is no overlap in responsibilities, then there is increased organizational risk from team members leaving the organization. A manager can mitigate this risk by analyzing which members work together and organizing the team such that there is more overlapping knowledge [12]. This risk is difficult to quantify, as there are limited methods for measuring team cohesiveness.

Researchers have investigated collaborative networks to understand different aspects of collaborations [8]. This paper proposes modeling the version control repository as a network, and applying network analysis techniques to identify the key authors for the project and to measure team cohesiveness.

The next section discusses related work. Section 3 discusses how data can be gathered from a source control repository. Section 4 discusses how network analysis techniques are applied. Sections 5 and 6 discuss the results of this analysis on some specific projects, and general trends resulting from the analysis of a few hundred projects. Section 7 analyzes the results. Section 8 concludes the article and suggests future research in this area.

2 Related Work

There are many metrics that can be used to analyze a software project, but there are very few metrics to identify key authors. Commonly used metrics include defect rate, complexity, test coverage, and productivity [10]. These metrics are rarely used to judge a specific author. Associating software metrics with specific authors can cause authors to feel threatened, and is not recognized as a best practice in industry [13]. Hence, typical software metrics are not available to solve this problem.

Other techniques have been developed to identify individuals familiar with specific areas of software. One such method is described by Linstead et.al. [6]. This method searches the source code for keywords or topics, and associates authors with the topics based on the history contained in the revision control repository. This method is able to identify an author who is familiar with a particular topic in the source code. Based on their results, this method is effective in identifying subject matter experts for specific areas of code. However, this method does not consider which authors are the core developers for the overall project, and does not take advantage of the existing relationships between authors that are available in the version control repository.

Another network analysis method is described by Lopez-Fernandez et.al. [7]. This method mined open source version control repositories to identify networks of authors and gain insight into the overall structure of a group of developers. The