Machine Learning for Software Engineering

Alberto Bacchelli

Machine Learning for SE: Day plan

When	What
9:00 - 10:00	Introductions and overview
10:30 - 11:15	**PyDriller on Software Ownership and Quality**
11:15 - 12:00	GHTorrent on Pull Request Acceptance
13:00 - 15:00	Mining Unstructured Software Data
15:30 - 17:00	Machine Learning and Mobile Apps

Machine Learning for Software Engineering

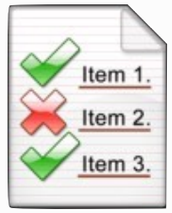
PyDriller on Software Ownership and Quality

Machine Learning for SE — Workflow

software engineering tasks helped

software development

software maintenance



software testing

...

empirical evidence

what is a good bug report?

what is the impact of code ownership on quality?

is pair programming useful?

machine learning and software analysis techniques

classification

program analysis

clustering

parsing

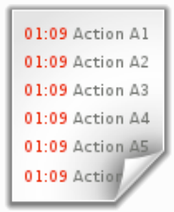
...

The software development process generates much data

Versioning system

Development environment

Issue tracking system



Program execution

Debugger application

Emailing system

...

Machine Learning for SE — Strong impact

Don't Touch My Code! Examining the Effects of Ownership on Software Quality

Christian Bird
Microsoft Research
cbird@microsoft.com

Nachiappan Nagappan
Microsoft Research
nachin@microsoft.com

Brendan Murphy
Microsoft Research
bmurphy@microsoft.com

Harald Gall
University of Zurich
gall@ifi.uzh.ch

Premkumar Devanbu
University of California, Davis
ptdevanbu@ucdavis.edu

ABSTRACT

Ownership is a key aspect of large-scale software development. We examine the relationship between different ownership measures and software failures in two large software projects: Windows Vista and Windows 7. We find that in all cases, measures of ownership such as the number of low-expertise developers, and the proportion of ownership for the top owner have a relationship with both pre-release faults and post-release failures. We also empirically identify reasons that low-expertise developers make changes to components and show that the removal of low-expertise contributions dramatically decreases the performance of contribution based defect prediction. Finally we provide recommendations for source code change policies and utilization of resources such as code inspections based on our results.

Categories and Subject Descriptors

D.2.8 [**Software Engineering**]: Metrics—*Process metrics*

General Terms

Measurement, Management, Human Factors

Keywords

Empirical Software Engineering, Ownership, Expertise, Quality

1. INTRODUCTION

Many recent studies [6, 9, 26, 29] have shown that human factors play a significant role in the quality of software components. *Ownership* is a general term used to describe whether one person has responsibility for a software component, or if there is no one clearly responsible developer. There is a relationship between the number of people working on a binary and failures [5, 26]. However, to our knowledge, the effect of ownership has not been studied in depth in commercial contexts. Based on our observations and discussions with project managers, we suspect that when there is no clear point of contact and the contributions to a software component are spread across many developers, there is an increased chance of communication breakdowns, misaligned goals, inconsistent interfaces and semantics, all leading to lower quality.

Interestingly, unlike some aspects of software which are known to be related to defects such as dependency complexity, or size, ownership is something that can be deliberately changed by modifying processes and policies. Thus, the answer to the question: *"How much does ownership affect quality?"* is important as it is *actionable*. Managers and team leads can make better decisions about how to govern a project by knowing the answer. If ownership has a big effect, then policies to enforce strong code ownership can be put into place; managers can also watch out for code which is contributed by developers who have inadequate relevant prior experience. If ownership has little effect, then the normal bottlenecks associated with having one person in charge of each component can be removed, and available talent reassigned at will.

We have observed that many industrial projects encourage high levels of code ownership. In this paper, we examine ownership and software quality. We make the following contributions in this paper:

1. We define and validate measures of ownership that are related to software quality.
2. We present an in depth quantitative study of the effect of these measures of ownership on pre-release and post-release defects for multiple large software projects.
3. We identify reasons that components have many low-expertise developers contributing to them.
4. We propose recommendations for dealing with the effects of low ownership.

2. THEORY & RELATED WORK

A number of prior studies have examined the effect of developer contribution behavior on software quality.

Rahman & Devanbu [30] examined the effects of ownership & experience on quality in several open-source projects, using a fine-grained approach based on fix-inducing fragments of code, and report findings similar to those of our paper. However, they operationalize ownership differently,

ESEC/FSE'11, September 5–9, 2011, Szeged, Hungary.

Examining the effects of ownership on software quality

- Source of data
- Features
- Machine learning algorithm

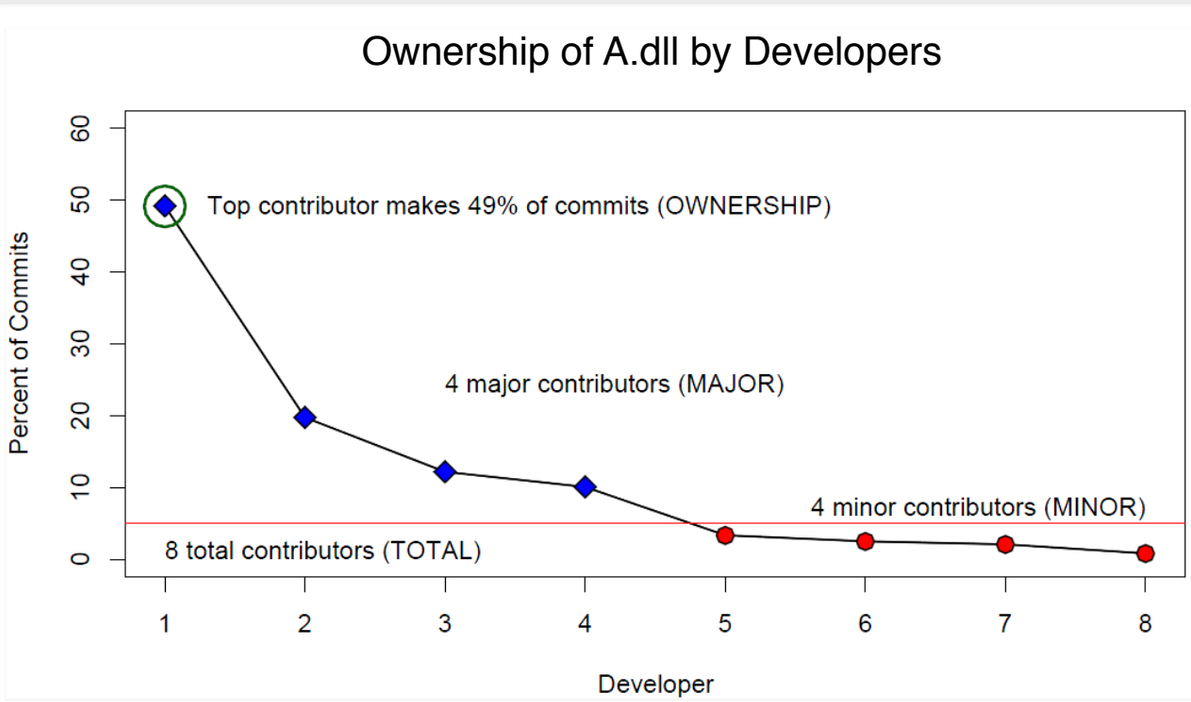
Examining the effects of ownership on software quality

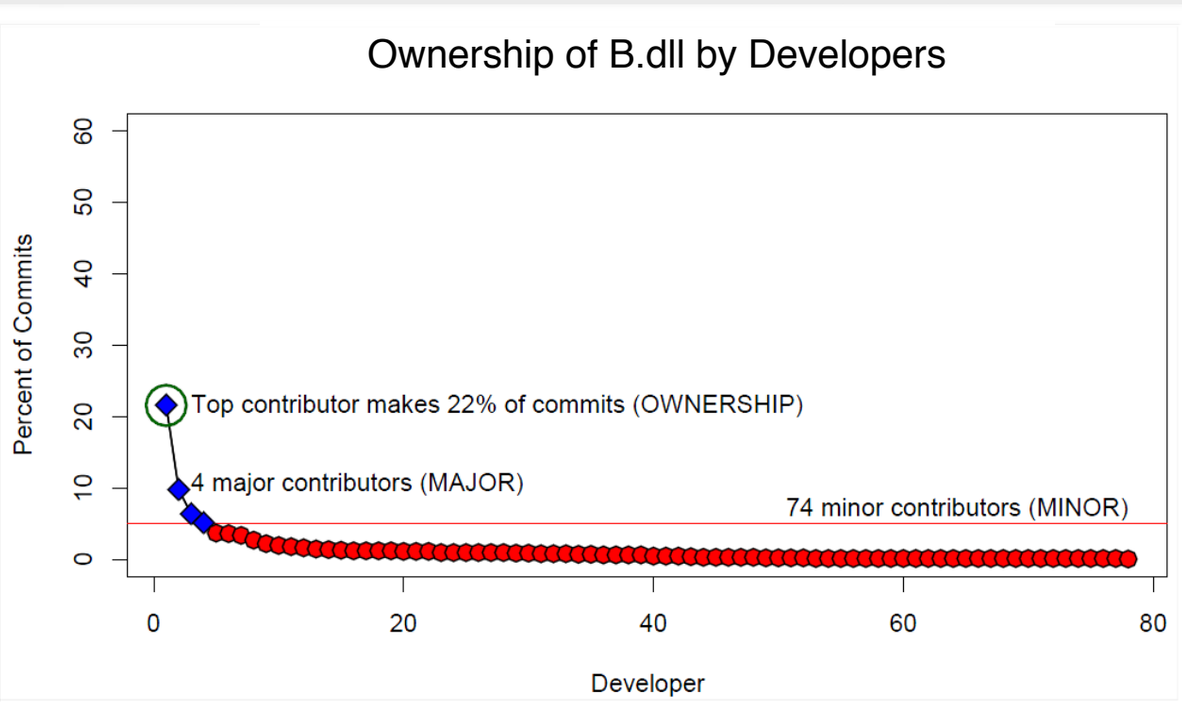
- What is ownership?
- Source of data
- What is software quality?
- Features
- What are the limitations?
- Machine learning algorithm

Examining the effects of ownership on software quality

- **What is ownership?**
  - code contributors
- **What is software quality?**
  - number of failures
- **What are the limitations?**
- **Source of data**
- **Features**
- **Machine learning algorithm**

Examining the effects of ownership on software quality





Examining the effects of ownership on software quality

- **What is ownership?**
  - code contributors
- **What is software quality?**
  - number of failures
- **What are the limitations?**
- **Source of data**
  - code changes to binaries
- **Features**
  - MINOR, MAJOR, TOTAL, OWNERSHIP
  - SIZE, COMPL, CHURN
- **Machine learning algorithm**
  - linear regression

Examining the effects of ownership on software quality

Model	Windows Vista		Windows 7	
	Pre-release Failures	Post-release Failures	Pre-release Failures	Post-release Failures
Base (code metrics)	26%	29%	24%	18%
Base + Total	40%*(+14%)	35%*(+6%)	68%* (+35%)	21%* (+3%)
Base + Minor	46%*(+20%)	41%*(+12%)	70%* (+46%)	21%* (+3%)
Base + Minor + Major	48%*(+2%)	43%*(+2%)	71%* (+1%)	22% (+1%)
Base + Minor + Major + Ownership	50%*(+2%)	44%*(+1%)	72%* (+1%)	22% (+0%)

PyDriller

https://github.com/ishepard/pydriller