Who Should I Contact?: Helping New Developers Find Experts

Patrick Carlson and Dr. Judy M. Vance

Introduction

For a new developer working on a software development project, knowing who to contact can be a difficult task. This is especially important for large geographically distributed software development efforts such as those that exist in Open Source projects.
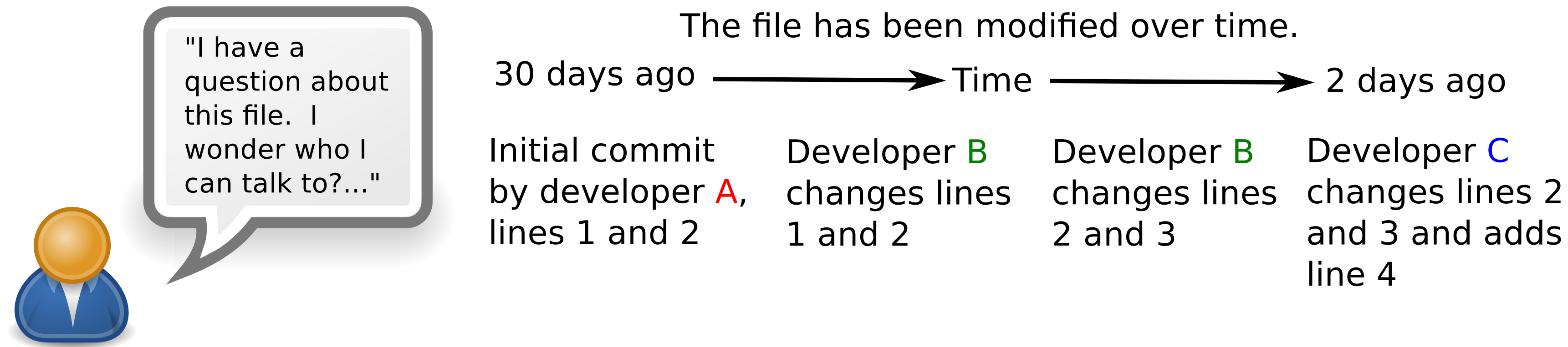
"I need help..."

Developer Expertise

Developer expertise represents the knowledge a developer has of a particular module or section of source code. As developers make changes, the codebase fluctuates and this knowledge degrades over time. New developers do not have the previous experience to guide them in identifying who to contact if they have a question. Our approach is to develop an algorithm, based on an examination of each line of a source code file and the historical changes that are made, to identify who is actively contributing to the project in an effort to improve new developer entrance into Open Source projects.

Algorithm

The algorithm uses exponential decay of knowledge to account for lost knowledge as well as historical source code revisions.

"I have a question about this file. I wonder who I can talk to?..."

The file has been modified over time.

30 days ago → Time → 2 days ago

| Initial commit by developer A, lines 1 and 2 | Developer B changes lines 1 and 2 | Developer B changes lines 2 and 3 | Developer C changes lines 2 and 3 and adds line 4 |

As an example, let's calculate developer B's expertise. Expertise is calculated as a ratio.

Days	30	10	4	2
1	A	(B)		
2	A	B	(B)	C
3			(B)	C
4				C

source code lines

The algorithm steps back through revisions for each line and determines the last time the author modified that line (if ever). Using a user-specified exponential decay amount it weights the knowledge for that line based on how long ago it was added or modified. These values are then summed across all the lines to determine the numerator.

$$(1 - 0.01)^{10} + (1 - 0.01)^4 + (1 - 0.01)^4 + 0 \longrightarrow \frac{2.82}{8} = 0.35 \quad \text{Expertise for developer B}$$

Number of days since commit

Exponential decay amount

Days	30	10	4	2	
1	(A)	(B)			2
2	(A)	B	(B)	(C)	3
3			(B)	(C)	2
4				(C)	1

source code lines

$$\frac{}{8} +$$

To determine the denominator, the algorithm counts the number of unique developers who contributed to each source code line and sums them.
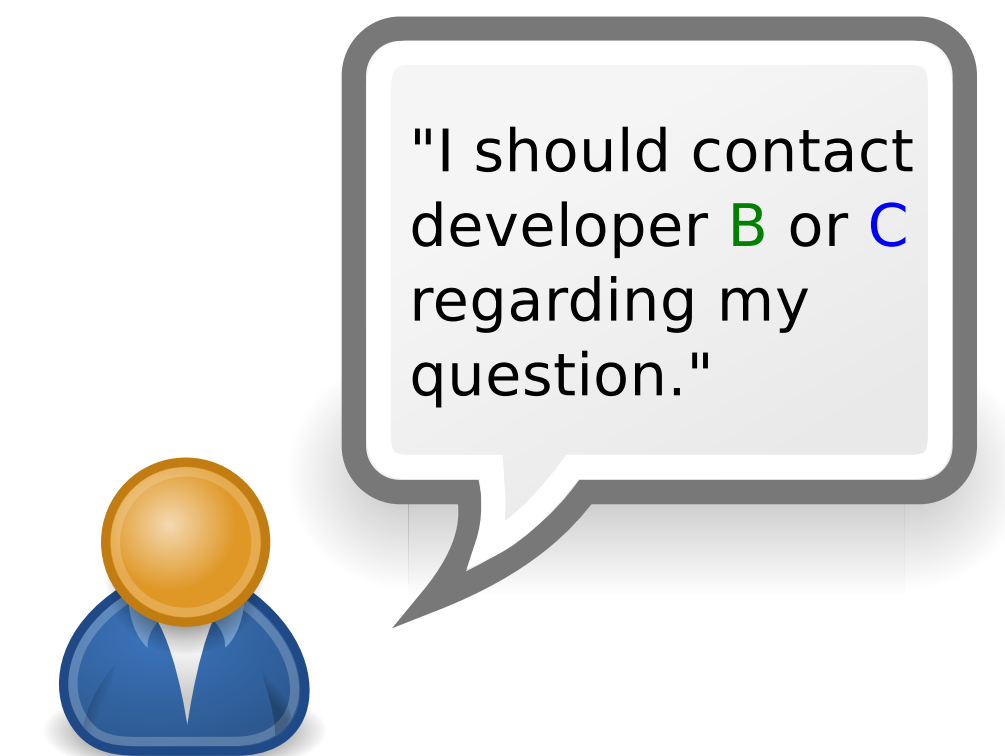
After calculating the expertise amount for all three developers, this information can be provided to the user.

Developer A: 0.18
Developer B: 0.35
Developer C: 0.36

Summing expertise across all developers is at maximum a value of 1, but is usually less. This is because the exponential decay of knowledge is decreasing this value based on the amount of time since the revision was made.

Result

This new developer now has suggestions for expert developers to contact regarding this source code file.

"I should contact developer B or C regarding my question."

Benefits

Expertise can be calculated at varying granularity levels from specific lines of a file all the way up to the entire project.

Future Work

Future work will compare the algorithm against existing expertise algorithms using online source code repository data. This will help validate the algorithm as well as identify how to best present the expertise information to the end user.