HubSpot Dev & Design

DEV & DESIGN

Home (/)

Blog (/blog)

Jobs (/jobs)

OPEN SOURCE

Home (http://github.…

Projects (http://githu…

Contributors (http://…

Git by a Bus (http://dev.hubspot.com /blog/bid/57694/Git-by-a-Bus)

Edmund Jorgensen (http://dev.hubspot.com/blog/author/edmund-jorgensen) **January 5, 2011 (http://dev.hubspot.com/blog/bid/57694/Git-by-a-Bus)**

Tweet   Vote on HN

Developers are always being told to document and collaborate on their code in case they are "hit by a bus." Usually "hit by a bus" is code for "go to another company," but the point is the same: if you are a developer with unique knowledge of code, that knowledge is at risk if you depart either your job or this life.

Recent Posts

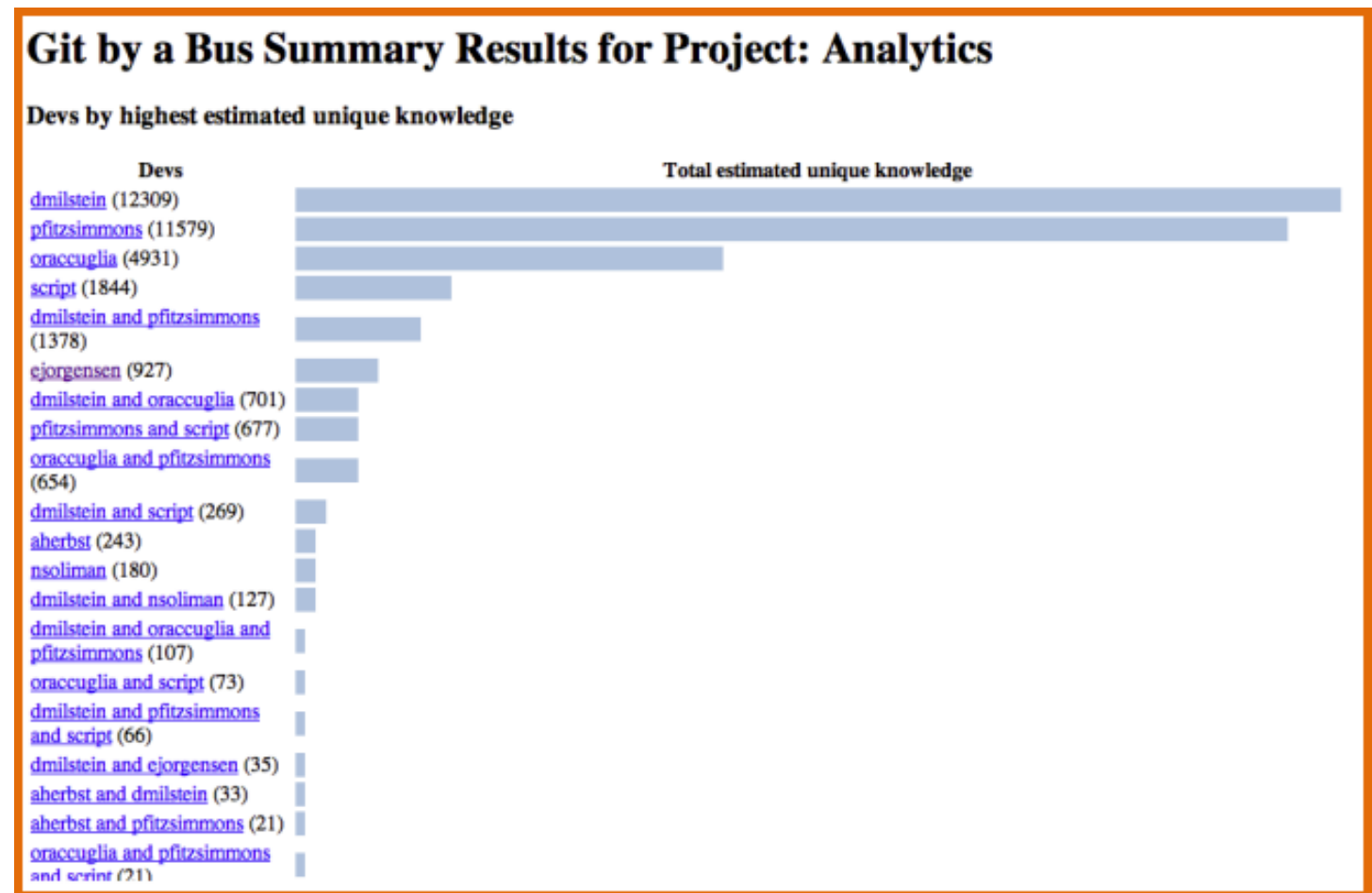What We've Learned About Interviewing Engineers (http://dev.hubspot.com/blog/interview-loop)

Announcing Mixen (http://dev.hubspot.com/blog/announcing-mixen)

5 Whiteboard Coding Tips for Interviews (http://dev.hubspot.com

I got interested in the problem of identifying and quantifying that risk, and Git by a Bus is the result. It analyzes the history of your git repository (there is also experimental svn support), estimates unique knowledge per file per developer or group of developers, and then writes an html summary of unique, at-risk, and orphaned knowledge for each project and file in your repository.

If you're interested in trying it out or hacking on the code, HubSpot has let me open the source, which is available on github here (https://github.com/tomheon/git_by_a_bus). Read on for more details on how it works.

Git by a Bus Summary Results for Project: Analytics

Devs by highest estimated unique knowledge

Devs	Total estimated unique knowledge
dmilstein (12309)	
pfitzsimmons (11579)	
oraccuglia (4931)	
script (1844)	
dmilstein and pfitzsimmons (1378)	
ejorgensen (927)	
dmilstein and oraccuglia (701)	
pfitzsimmons and script (677)	
oraccuglia and pfitzsimmons (654)	
dmilstein and script (269)	
aherbst (243)	
nsoliman (180)	
dmilstein and nsoliman (127)	
dmilstein and oraccuglia and pfitzsimmons (107)	
oraccuglia and script (73)	
dmilstein and pfitzsimmons and script (66)	
dmilstein and ejorgensen (35)	
aherbst and dmilstein (33)	
aherbst and pfitzsimmons (21)	
oraccuglia and pfitzsimmons and script (21)	

I experimented with a few different models for unique knowledge as I was writing the script, but the one I settled on is pretty simple and has the virtue of allowing you to estimate risk with the joint probabilities of more than one developer being hit by a bus.

In the final model, the "knowledge" contained in a file at any point in time is represented as the number of lines in the file plus an adjusted "churn" value. "Churn" is the number of lines that are estimated to be changed rather than added in a revision, and we multiply by a constant (by default 0.1) to translate that churn into a knowledge value. In other words, we estimate that for every line added to the file, one knowledge point is added, for every line removed, one knowledge point is removed, and for every ten lines changed, one knowledge point is added.

Using this model, Git by a Bus marches chronologically through the log of each file in your repository, applying the following algorithm for each revision:

- If the revision resulted in a net gain of lines, assign that net gain as unique knowledge to the author of the revision (we'll call him A).
- If the revision resulted in a net loss of lines, find the percentage of the total knowledge represented by those lines and destroy that percent of unique knowledge held by each dev or group of devs.
- For the "churn" of the revision, as defined above, multiply it by the churn knowledge constant (0.1 by default) and assign that as new unique knowledge to A.

- For the "churn" of the revision, as defined above, multiply it by 1 - the churn knowledge constant. That gives us an estimate of knowledge that used to be unique to other developers or groups of developers, but which A now shares. Find the percentage of the total knowledge of the file represented by this newly shared knowledge, and then for each dev or group of devs that does not include A, remove that percentage of their unique knowledge and assign it to a group including the original dev or group of devs as well as A.

At the end of this process, we have basically built up a venn diagram of how all the knowledge in the file is distributed. For example, if three devs A, B, and C had participated in a file's history, the output of our algorithm would show how much knowledge each dev held uniquely, how much A and B held jointly, or A and C, or B and C, and how much estimated knowledge was shared between all three.

Now, using supplied input files that tell us which devs have already departed, and the estimated risk of each remaining dev getting hit by a bus in some time frame, we can calculate:

- "Orphaned" knowledge--that is, knowledge that was estimated to be held uniquely by a dev or groups of devs who are all departed.
- The expected unique knowledge that each file / project stands to lose in that time frame, calculated by multiplying the unique knowledge of each dev or group of devs by the joint probability that all dev or devs in that group will be hit by a bus. The algorithm assumes that the probability any two devs will both be hit by a bus is independent, so it doesn't take into account devs who

bike into work together, are conjoined twins, or enter into bizarre death-by-bus pacts.

(http://www.hubspot.com/jobs)

Comments

Amy Eastment 1:21 PM on January 05, 2011

Dude, I love this! Can't wait to see what our internal risk analysis looks like.

Dan Dunn (hubspot.com) 2:04 PM on January 05, 2011

My ego is not pleased. Evidently, my checkins aren't unique or special. I feel . . . like I don't write important code!

Dan

Mike Volpe (http://www.mikevolpe.com) 2:58 PM on January 05, 2011

Damn! Who the hell is "script"? I think s/he is underpaid.

Brian Whalley (www.emptyport.com) 9:33 PM on January 05, 2011

Dude. You have no ideas how many compliments/praise I've gotten today from other folks who discovered this and are now using it. Awesome.

Publius 11:48 AM on February 06, 2011

Isn't it important to know why lines are removed, not just added or changed? You're hit by a bus if you lose the developer that knew the code well enough to know which lines could be removed. ("I made this letter longer than usual because I lack the time to make
it shorter.
Pascal, Provincial Letters XVI".)

Add a comment

Name

Email

Website

Comment

Submit Comment

HubSpot

Home
(http://www.hubspot.com)

Products
(http://www.hubspot.com
/software)

About
(http://www.hubspot.com
/internet-marketing-
company/)

Projects

Messenger
(http://github.hubspot.com
/messenger)

Facewall
(http://github.hubspot.com
/facewall)

jQuery Zoomer
(http://github.hubspot.com
/jquery-zoomer)

Humanize
(http://github.hubspot.com
/humanize)

Teeble
(http://github.hubspot.com
/teeble)

Dev & Design

Home
(http://dev.hubspot.com)

Blog
(http://dev.hubspot.com
/blog)

Jobs
(http://dev.hubspot.com
/jobs)

OS Projects
(http://github.hubspot.com
/projects)

Developer API

Developer
Documentation
(http://developers.hubspot.c

Powered by the HubSpot COS (http://www.hubspot.com/products/content-optimization-system)