“PHP is a very loosely typed language and that was one of the reasons I loved it so much because there weren’t as must restrictions as other strongly typed languages such as JAVA and C++. The issue with that though is that sometimes somebody who is looking at your code might have to do a lot more work to find out and make sure that they are setting the right variables and returning the right type of variable. I see this though as a good request to help structure and make code collaboration easier.”

Hi Adrian,

Thank you for bringing up this topic. When I first learned Perl 15+ years ago, I have to admit that I first hated it, and this was a big reason, coming from a C background. However, I eventually learned to like it because I could write code very fast and its handling of regular expressions was perfect for dealing with genomic data. Nevertheless, even though I am not experienced in PHP, I fully agree with you that implementation of this feature would make programs much easier to understand. I have been given the task of maintaining a lot of obfuscated scripts where I work, and something like this would make my job much easier. Another problem that I have encountered with software implementing the features of weakly typed languages is that they are vulnerable to “type errors”. Many times, I would not know that there was a problem until a program finished executing. Strongly typed languages would prevent this by reporting the error at compile time.

I noticed that you have a space between “get Average”. Is there any particular reason you did this instead of “getAverage” or “get\_Average”?

Regards,

Emanuel

Hi Zubair,

Thank you for reading and your reply. I fully agree with you that the decision to outsource is largely dependent on the organization itself. It sounds like in the first situation that you described, it would have been ideal for them to just build the system in-house. I’m actually a bit surprised that a large technology company chose to go the outsourcing route instead of just doing it in-house themselves.

The company I work for would never go the outsourcing route since we sell products that are predominantly based on trade secrets. One of our biggest products is consumable molecular probes that used for detecting specific DNA fragments. We have more than 8 million of these in our database systems, and when customers order them, they are immediately manufactured (only taking an hour), and then shipped. However, we do not disclose the exact formulation of them. If we were to ever contract out our website to another vendor, we would risk these formulations getting out into the public sphere. Considering that the revenue for our consumables is billions of dollars a year, it would make very little sense to not develop and maintain our website in-house (Life Technologies, 2014). However, for a non-profit organization that takes donations through its website, it would probably make more sense to outsource. While true that it may choose to record the information of its donors (e.g., name, contact information, credit card), I think the risk is minimal as long as the outside vendor does not have a bad reputation for security breaches.

Reference:

Life Technologies (2014) Life Technologies Announces Fourth Quarter and Fiscal 2012 Results [Online]. Available from: <http://www.lifetechnologies.com/us/en/home/about-us/news-gallery/press-releases/2013/Life-Technologies-Announces-Fourth-Quarter-Fiscal-2012-Results.html> (Accessed: 23 June 2014)

Hi Zubair,

That’s a good question. If I were the CIO of MTN South Africa, I would take a long-term view. This means that I would do everything within my power to build a team in-house to build and maintain the website. This is of course, easier said than done. A project like this would need a sponsor (which would probably be the CIO), a project manager, teams of software engineers for both development and QA/QC, and of course, a capable human resources team.

However, considering that you mention that there is a shortage of skills in Africa, this would not be an easy feat. I live in Silicon Valley, and there are still more open job positions than available software engineers. If there were an immediate need, I would do both: seek an outside vendor and attempt to build the necessary team. If it looked like the latter would not be possible the immediate future, I would outsource the project, but require that once the website was built, they are required to provide our selected employees training on how to maintain it. This would of course require that MTN South Africa identify its in-house employees capable of doing this. My organization has a website only accessible through our company’s intranet, which contain a profile page of every employee. Every employee is encouraged to complete their profile, which includes listing all of their skills. Perhaps something like this could be implemented at MTN South Africa. However, this would not happen overnight. Organizations such as these should always be thinking long-term, and this includes self-sufficiency.

Regards,

Emanuel

Hi Frank and Albert,

I’ve been programming in Perl for years, and I did not even know about this feature in the link you provided. This increment function is probably the same as the one utilized by the UNIX split command when naming its output files. For example, if I wanted to split a file containing 5000 lines into 1000 line files, I would enter at the prompt:

$ split –l 1000 filename

This would give me 5 files named: xaa, xab, xac, xad, xae.

My guess is that this split executable uses that same way of incrementing. Just off the top of my head, I think that this string incrementing feature could work if one were trying to find all of the words with a certain prefix.

I just gave it a try from the terminal program on my Macbook, and got the following:

$ perl -e '$string="fooa\n"; ++$string; print "$string\n";'

1

$ perl -e '$string="fooa\n"; ++$string; ++ $string; print "$string\n";'

2

I’m wondering if maybe this has been changed the version of Perl I have on my Macbook. Nevertheless, I will probably stay away from attempting to increment or decrement strings, and reserve it for integers. To me, it seems rather irrational. Even if it is allowed, I think that it’s hard to predict its behavior. I would probably only use it if I wanted to purposely obfuscate my code.

Regards,

Emanuel

“In my company this is a serious issue as most if not all of the legacy systems were developed by one vendor which dates back 14 to 15 years ago.”

Hi Nadia,

You bring up an important issue regarding legacy systems and their maintenance. Where I work, I sometimes am given the task of fixing legacy code developed not by outside vendors, but in-house. Some of this code is more than 10 years old, and is always a major pain to fix. The Agile community has a term they have colorfully named a project’s “truck factor”, which measures “The number of people on your team who have to be hit with a truck before the project is in serious trouble” (Bowler, 2005). So what it essentially means is that the lower the number, the worse the risk is to a project. Perhaps this issue could be brought up when seeking an outside vendor. If the vendor has a business environment that requires everything be documented, then employee turnover has less of an impact. Thus, the truck factor of each project is likely to be high. I think a good indicator is to see what certifications the vendor holds (e.g., ISO, CMMI). Vendors that have been certified by these bodies usually have strict documentation standards in place. Therefore, years down the road, their help can be sought out on old projects.

Regards,

Emanuel

References:

Bowler, M. (2005) *Truck Factor* [Online]. Available from: <http://www.agileadvice.com/2005/05/15/agilemanagement/truck-factor/> (Accessed: 25 June 2014)