July 8, 2018

Course: CIS570 – Business Intelligence

Name: Robert Palumbo

Assignment: Reading Discussions – Week7-Session1

Due Date: Sunday, July 8 @ 11:59pm

Question 1. What are some downsides to open-source BI solutions?

One of the major drawbacks of *any* open-source solution which was not fully addressed in the article has to be software maintenance. When you use open-source software of any variety that you obtain for *free* under any general license, you will undoubtedly at some point in time encounter an issue or *software bug* that will adversely affect your application and use of the software. This typically occurs due to the nature of open-source; there are many people in the Internet universe of developers that are working on the software and making changes to it on a daily basis. As such, this world is not quite the same as that of a business that develops a specific product line, e.g. SAS for BI, in which there is *typically* a well-defined software support process for managing customer issues.

For open-source, depending on the nature of the issue, it may be very problematic finding an appropriate contact that can assist with a resolution path. Developers that work on open-source solutions provide contact information, but you are clearly at the mercy of the developer as to when the problem will be resolved once it is successfully reported.

The open-source development world does not often have a stringent test process to follow and with myriad developers working on a typical project it can and often is the case that the changes submitted by one developer may break the code of another subsystem that they might not even be aware of. Further, developers are not bound to provide a fix or resolution within any specific time-frame. That being said, most developers do not want to be associated with writing poor quality code and have their reputations tarnished so many will address issues as soon as they are reported. But the path to a fix may take quite a bit longer than acceptable. When that happens, if you have the proper skills you may be able to debug and resolve the problem on your own – or find someone that can assist you.

This exact problem was encountered in my CIS575 Data Visualization class (Spring 2018). We used an R package for *Gephi* visualization support. Both of these are open-source. In between semesters of course offerings, the Gephi package was modified by a developer and some of its functionality stopped working. It was not apparent until the package was used for a class assignment. Our instructor, Dr. Hayne, identified the issue and tried to find a fix for it online. While someone did in fact try to fix the problem, it was not fully resolved. Dr. Hayne spent time on his own and corrected the issue. There in lies both the pro and con of using open-source.

<https://www.r-project.org/>

<https://gephi.org/>

Question 2. Discuss your organization's experience with open-source BI.

I spent a considerable portion of my career working for Qualcomm Inc., a company that is arguably the leader in next-generation mobile technologies and platforms. During my time at Qualcomm I worked on a project that required integrating and open-source software package named *LWIP* or *LightWeight TCP/IP.*

This package supports a full TCP/IP software stack but does so by using a very small amount of RAM making it very useful for embedded software development – which is what the majority of Qualcomm projects are based upon. The purpose for using this package was to implement a TCP/UDP framework within the core of Qualcomm’s MSM7X multi-core platform to allow the various processors to communicate diagnostic messages and metrics to a central processor which could then be offloaded to a number of different repositories for processing.

Now Qualcomm could have implemented something similar without the need for this open-source package, but several factors overruled this option. These factors included budget constraints, time-to-market, LWIP was a tried and true option used by other organization, and there was no monetary code associated with using it.

The caveat with using the package however was that the Qualcomm legal team had to give approval first. Because the software was not developed by Qualcomm proper, it could not be used first hand in any development project that would be released into the commercial marketplace without first being vetted by the legal team. The reason being is that Qualcomm, being such a dominant player in the telecommunication spectrum, could not afford to release any product software that included open-source which could be subject to unknown licensing issues, and further, might introduce adverse product performance or security risks.

The reference LWIP source-code has to be provided in full to the legal team which then performed a full vetting. The open-source license in play at the time was scrutinized to ensure it would not conflict with current Qualcomm any licensing requirements or could restrict or otherwise bind Qualcomm into unwarranted legal doctrine.

Qualcomm utilized a software product from *BlackDuckSoftware* that is specifically designed for vetting of open-source software and associated licensing. The software analyzes the open-source code providing security and risk analysis and full visibility into the source-code identifying any potential software vulnerabilities. The Black Duck product offers companies a full open-source management platform to fully track and maintain all open-source software in use. It is highly recommended for any project containing open-source software.

<https://savannah.nongnu.org/projects/lwip/>

<https://www.blackducksoftware.com/>