What’s Next in Version Control for SAS environments?

PhUSE London 13 Oct 2014, David J Garbutt

01. Abstract

Since 2014 Version Control systems have become recognized as essential for SAS programmers working in medium to large scale environments and have been widely implemented as well as being included in environments such as Macumba EnintICeR® and SAS®Drug Development. But what next? What is new in the computing world and the world of version control? What are the essential drivers of change? Where can, and will, we be in two years time? This paper presents a view of the future that is nearly here.

The story so far...

First came custom in-house systems like GPS II and Macumba/Wave and then commercial systems like SAP® Drug Development and enintICeR® by Enint. All the systems so far are using the client server model and have a centralised version Control system standing at the ready.

02. Limitations

The systems mentioned are successful, how can we talk of limitations? These systems have structural problems that arise from their client server architecture.

• Centralised systems require big resources in all the costs and testing that goes within and achieving a costable response time is a constant effort. Centralised systems are visible for groups of 50 programmers or more.

• Final development on large servers are not dramatically faster than small local PCs.

• Current small systems can create a fix run on their own PC faster.

• Pharamaceuticals are globalised and to practice the centralised system does not support this well.

• Documentation is often not in the same system, it’s often not machine readable and changes cannot easily be linked to the code or the correct output version.

03. Drivers of change

The balance of power between remote and local PC and the transfer speed from the PC to the server.

The client server architecture grew popular in the go’s 070’s.

04. Distributed VC

DVC systems were developed in the 2000’s. They harness central repositories, every machine hosts a full repository, each node of the community constantly and automatically. They assume a model of changes being back and forth changes rather than a single file, these would be associated with labels (rather systems). This is somewhat important of the access control central backup server.

Two popular systems are and Mercurial. Both have active communities and integrated tools within hosting environments for easy sharing.

05. SVN & git commands

Wherever version control commands exist/


06. Integrated DVC

However current systems are not integrated with bug tracking (the best known systems for this are still client server systems like Redmine and HPBug). Integration would also be well suited with a wiki or other document management.

Systems are being built with all these pieces include Vcsavity (https://sourceforge.com/vcsavity) which was released in 2011. It also supports project tree tracking, and has agile features like burn down charts and support for sprints.

07. Distributed Databases

The last brick in the wall is the development of NoSQL databases. They are very varied but many have a distributed model as default (erexample Redis and Cassandra). These could be used on a single client for metadata and other shared and synchronised information, such as test results. Titles and footnotes could also be shared like this.

08. All-In-One Client

The time could be right to build a single machine client for use on a SAS IDE with version control. All the pieces seem to be in place.

• ADS/ICS there will be a shared repository for project number.

• ANoSQL database

• SAS® DB and R installation

Perhaps the last piece needed is the auto build system and an automated way to spread final runs across machines.

09. Virtualised

The advantage of the all-in-one client is the build and maintenance simplicity as well as the linear growth of costs with a fixed startup price. This would be a breakthrough in cost vs functionality for small – medium companies.

And for larger companies it can be noted that a system in a single machine can be more easily virtualised and therefore scale to hundreds of users.

In principle such a system could be hosted on cloud services such as the NNIT GCP cloud.

010. What is next?

Integrated, distributed systems are becoming the norm and over an opportunity for small and medium sized companies to have a real SAS IDE with Version Control integrated. Hope that this is the future for SAS VC systems.

Further Reading:
See the paper version which has more details and more references.

http://www.nanoteq.com/blogs/2014/03/26/what-is-the-future-of-sas-vc-systems/