Enterprise-level Transition from SAS® to Open-Source Programming for the whole department

Kevin Lee, Assistant Vice President at Genpact;
Manikandan Jeeva, Assistant Vice President at Genpact;

ABSTRACT

The paper is written for those who wants to learn the enterprise-level transition from SAS to open-source programming. The paper will introduce the transition project that the whole department of 150+ SAS programmers has completely moved from SAS to Open-source programming.

The paper will start with the scopes of the project – Analytic platform switch from SAS Studio to R Pro Server, converting the existing SAS codes to R/Python codes, Window server to AWS Cloud computing environment, and the transition of SAS programmers to R/Python programmers. It will also discuss the challenges of the project such as inexperience in Open-source Programming, new analytic platform, and change management. The paper will introduce how the transition-support team, executive leadership and SAS programmers have overcome the challenges together during the project.

The paper will also discuss the difference in SAS and Open-source language and programming, and it will show some examples of the conversion of SAS codes to R/Python codes. Finally, it will close with the benefits of the Open-source programming transition and the lessons learned from the project.

SCOPE OF OPEN-SOURCE TRANSITION PROJECT

The scope of open-source transition project was
- Analytic platform switch from SAS Studio to R Pro Server
- The Transition from SAS programming to R/Python/SQL programming
- Conversion of existing SAS codes to R/Python/SQL codes
- Window environment to AWS Cloud Computing (EC2, Redshift, S3)
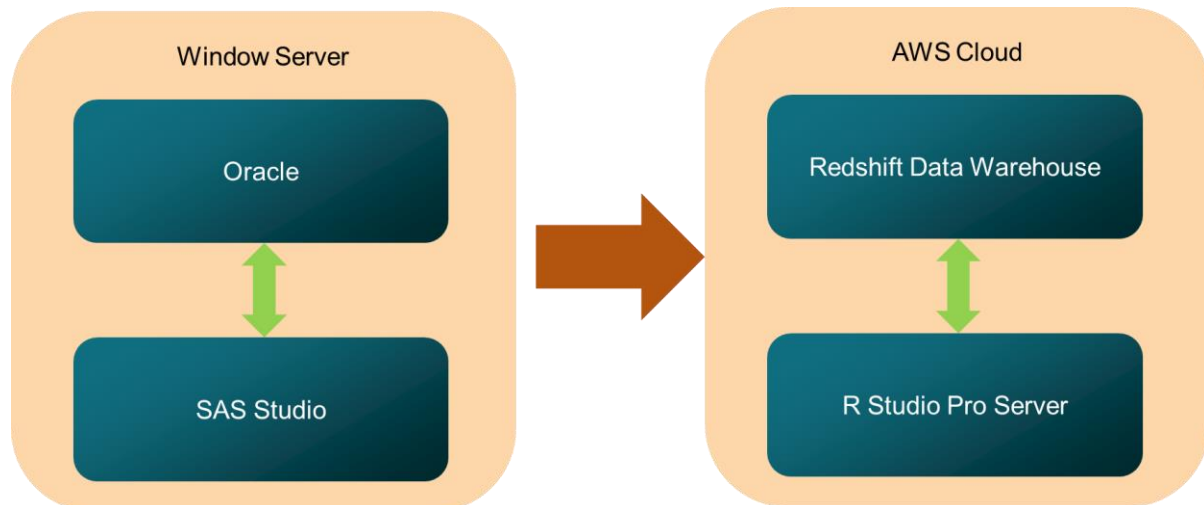
CHALLENGE OF THE OPEN-SOURCE TRANSITION PROJECT

The complete transition from SAS to Open-source programming was not easy. Most of programmers were used to SAS programming and SAS analytic system.   They should learn the new languages and new systems such as R, Python, R Studio and Jupyter. The transition supporting team, executive leadership, sponsor team members, and SAS programmers had to go through following challenges.
- Programming difference in open-source programming in R/Python
- Analytics Platform difference in R Studio / Jupyter
- Computing Environment in Cloud Computing
- Operation System in Linux
- Governance / Version Control in Open-source programming using Git
- Lack of enterprise customer support during the transition
- Daily works as well as existing SAS codes conversion

- New Culture created by Open-source Programming

CODE DIFFERENCE IN SAS AND OPEN-SOURCE (R / Python / SQL)
There are a lot of similarities between SAS codes and R/Python codes, and PROC SQL is very much identical to SQL codes. So, SAS programmers tend to learn R/Python languages relatively faster.   But there are also a lot of differences between SAS and R/Python codes.

The main difference between SAS and R/Python programming is how the data is processed. Usually, SAS manipulate the data a row by a row, but R/Python/SQL manipulates the data by a column.   For example, when creating DM.SEX from SEXC, SAS codes are written like below

```
data DM;
        set DEMO;
        if SEXC = 'Male' then SEX = 'M';
        else if SEXC = 'Female' then SEX = 'F';
        else if SEXC = 'Unknown' then SEX = 'U';
run;
```

Python codes are written like below.
```
DM['SEX'] = DM.SEXC.replace(['Male','Female','Unknown' ], ['M', 'F', 'U'])
```

Here Python will pick only SEXC variable and use it to create the new variable of SEX. By doing it, Python will save the memory since it does not need to navigate other variables to create SEX variable.

However, if the programmers want to use two variables to create the new variable, SAS does not need to call out the required variables, but just to use them in data step.   However, Python will need to call out two variables.

Another main difference comes from how open-source codes are structured.   SAS is the enterprise analytic system, so all the necessary components are included in BASE/SAS, but Python or R should import the necessary components. For example, if the programmers want to read and write SAS datasets from the local drive in Python, the programmers need to import SAS function in Jupyter using "import xport".

HOW TO SUPPORT SAS PROGRAMMERS DURING THE TRASITION
SAS programmers are eager to learn and start using the new open-source languages of R, Python and SQL, but starting to use the new languages are not easy transitions. It will take time and efforts. To support SAS programmers, the leadership and transition support team work together to support a smooth transition.

First, SAS programmers could learn R and Python themselves, but in order to support and expedite a smooth transition, transition supporting team provided the trainings on R and Python programming in the beginning of the transition.

The transition supporting team also provided many workshops on new analytical systems and show the followings.
- How to use R Studio, Jupyter and SQL workbench
- How to connect data from R Studio, Jupyter and SQL workbench to Redshift Data Warehouse
- How to exchange files between AWS and local environment
- How to use Git

Secondly, there are more than 250 existing SAS codes. Some are 2 or 3 pages long, and some are longer than 50 pages. During the transition, the programmers are also performing daily tasks such as processing the data and preparing the reports. It would be too much for the programmers to convert existing SAS codes while performing daily tasks, so the transition supporting team have converted all the existing SAS codes to R/Python codes.

After converting SAS codes to R/Python/SQL codes, transition supporting team has supported SAS programmers with User Acceptance Test (UAT).
- Transition supporting team walked through the newly converted R/Python/SQL codes with SAS programmers
- Transition supporting team provided new SOP documents so that SAS programmers could refer to them to understand the newly converted codes, input data, intermediate data, final data and output.
- SAS programmers were given time to run and review the newly converted R/Python/SQL codes.
- If there are any questions regarding the newly converted R/Python/SQL codes and SOP document, transition supporting team supported SAS programmers.

During UAT process, SAS programmers will be able to learn how to run the newly converted R/Python/SQL codes in new analytical system.
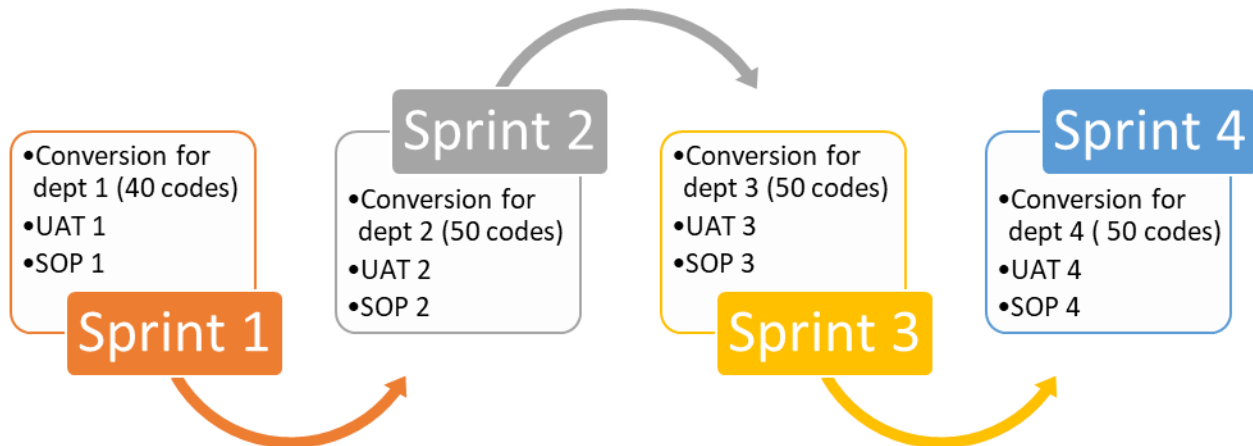
Thirdly, during the transition time, there are a lot of questions from the team regarding R/Python programming, R Pro Server, Jupyter Notebook and AWS Cloud Computing, and it is very important that transition team should be able to answer those questions and walk through with programmers. Some programmers could easily transition into the new languages and system, but for some, it could be a big challenge. If programmers know that they could trust a transition support team, it will make this kind of transition a success.

Lastly, but probably the most importantly, leaderships have been heavily involved from the beginning and managed the changes throughout the project. Below are change management activities by leadership.
- **Clear goals, plan and timeline** of the open-source transition project
- **Executive Involvement / Support** throughout the project
- **Enough Resources and Budgets** for the project
- **Frequent, transparent communication** on the project progress/update
- **Full Participation** from the whole team
- **Dedicated Supporting Team /SME**

SPRINT CYCLE OF SAS CODES CONVERSION
The transition supporting team applied a sprint cycle process for SAS codes conversion. Rather than converting all 230 SAS codes at once, transition supporting team converted the portion of SAS codes at a time (a sprint). For each sprint, transition supporting team converted SAS codes for each department for the scheduled time, provided the new R/Python/SQL codes to SAS programmers, and SAS programmers reviewed the codes, ran and validated the outputs within a given time.

Sprint 1
- Conversion for dept 1 (40 codes)
- UAT 1
- SOP 1

Sprint 2
- Conversion for dept 2 (50 codes)
- UAT 2
- SOP 2

Sprint 3
- Conversion for dept 3 (50 codes)
- UAT 3
- SOP 3

Sprint 4
- Conversion for dept 4 (50 codes)
- UAT 4
- SOP 4

BENEFIT OF THE OPEN-SOURCE TRANSITION PROJECT
During the transition period, SAS programmers were able to learn the new languages in R and Python, the new analytic system in R Studio/Jupyter, and AWS Cloud computing environment (EC2, Redshift, S3). This helps the whole organization and the individuals to learn and improve themselves rather than staying in the status quos.

Since open-source solutions are openly available and free to use, the team tends to implement the new solution faster rather than waiting until the enterprise system make them available, so programmers are more open to new, innovative, advanced analytics such as Big Data, Machine Learning and Visualization.

LESSONS LEARNED FROM THE OPEN-SOURCE TRANSITION PROJECT
Learning new languages and new systems take a lot of times. The fact that the open-source transition project did not happen in a rush but went through many trainings and workshops over a course of a year really helped programmers to learn the new programming languages and the new analytical systems.

During the transition, the leadership of the department and transition support team has played a major role. The leadership has provided the proper timeline and resources, so programmers could have the proper support during the transition.

CONCLUSION
The transition from SAS to Open-source programming for the whole department was not an easy transition. It was a huge challenge, especially when the transition happened to the whole department with more than 150 SAS programmers who did not know much about open-source programming and system. But under the leadership of the sponsor executives and the know-how of transition supporting team, the whole department was able to successfully transition to open-source programming in Cloud computing. After the successful transition, the new culture has

evolved. The team tends to be adopting new technologies faster and more proactive since programmers are less dependent on tools and more open to the new technologies such as Machine Learning.

REFERENCES
- R Pro Server in https://www.rstudio.com/
- Jupyter lab/notebook in https://www.anaconda.com/
- AWS Cloud Computing in https://aws.amazon.com/

CONTACT INFORMATION
Your comments and questions are valued and welcomed. Please contact the author at

Kevin Lee
AVP of AI/Machine Learning Consultant
kevin.lee@genpact.com

Manikandan Jeeva
AVP of Commercial Analytic Solution
Manikandan.Jeeva@genpact.com

TRADEMARKS
SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries.

® indicates USA registration. Other brand and product names are registered trademarks or trademarks of their respective companies