

Regulatory Compliance in the Digital Age – A cloud-based statistical computing environment (SCE) to the rescue

Tyagrajan Swaminathan, Sridhar Vijendra, Efficacy Lifescience Analytics

ABSTRACT

The present-day processes in statistical programming for clinical trials are going through a major change, where geographically distributed teams located in multiple time zones across the globe work collaboratively under increasingly stringent data quality, integrity, security and privacy regulations that call for full traceability, data lineage, audit trails and adherence to data standards. To address these, our organization has recently deployed a modern web-based cloud enabled Statistical Computing Environment (SCE).

This paper will describe the experience of our organization in migrating to a cloud-based SCE. Migrating from a conventional in-house system to a cloud-based SCE was a big step but its many long-term benefits far out-weighed the initial efforts in setup and training. At an enterprise level, a cloud-based SCE establishes a shared environment with a single step login where sponsors and collaborators can work together. Such a system regulates access, maintains role-based security with a detailed audit trail, facilitates macros management, templatises study artifacts, customizes workflows and helps develop and maintain productivity metrics. From a manager's perspective, it helps track programming status and milestones, maintains version control, exercises control over the unblinding process and eases dependency management. And finally, for a statistical programmer, it provides a seamless and intuitive programming interface. In this paper, we will highlight the key challenges and long-term takeaways of implementing a cloud-based SCE.

INTRODUCTION

Day in and day out, deliverable after deliverable, the one thing that never changes is the statistical computing environment (SCE) we use to produce, deliver and store the complex analyses we create study after study. Conventional folder-based systems, common in many organizations, are simple SCEs in their own right. The most popular of such SCEs run on a UNIX back-end, provide a Windows-based mapped-drive interface and are most often coupled with utilities such as a version control system and a statistical analysis package, usually the ubiquitous SAS®. Many organizations accessorize such systems with proprietary reporting engines, global macro libraries and click-and-run interfaces. This, along with an independent document management system completes the setup that has been serving us well for the past couple of decades.

With the onset of globally dispersed teams and widespread use of remote connectivity, CITRIX-based applications and virtual desktop interfaces have come to our rescue to connect to remote servers halfway around the world. While the work gets done, we are most often left with multiple levels of logins and multiple windows to traverse that can exasperate even the most spirited user. Is a single-step login the answer to such woes?

Considering that almost every standard operating procedure (SOP) we read and the training we take ultimately highlights the all-important aspect of patient safety in our work, the term audit trail is not at all foreign to our industry. It is vital for each step in each of our processes to be tracked and recorded meticulously. However, there is a limit to what conventional SCEs can achieve in this regard. But to meet regulatory requirements, the lack of such abilities is compensated by additional steps and processes in the document management system and organizational SOPs that ensure compliance. But often such processes seem superfluous to the logical user. One such example is the use of a programming status tracker to track the various outputs in a study deliverable. Instead of having a disjointed tracker that needs an update every time an output is validated, wouldn't it be easier if there is an automatic update in the system when an output passes QC? Significant time and money are spent in attempts to streamline and speed up reporting activities using foolproof but convoluted processes. Is there a way to track the

various steps and tasks in an SCE and integrate our processes with the system we use so that we can ensure regulatory compliance without over complicating our processes?

A secure cloud-deployed SCE may have a solution to the serious limitations of our conventional SCEs. Sycamore SCE is the cloud-based SCE that our organization decided to adopt to assess how we could improve on our processes while ensuring regulatory compliance and make day-to-day tasks easier for our study teams. It boasts several features including one-step login, in-built task management, detailed audit trail, and role-based security while also supporting most of the features provided by conventional SCEs. It also allows a great deal of customization in various aspects. Would such an SCE help us turn around deliverables more quickly? Would it make our processes more efficient? Would it make tracking all activities and projects easier than with a conventional SCE? If so, what would be the trade off? Would it have a steep learning curve that will keep our teams away from it? Would it over-complicate simple things? Where would it stand in the risk-vs-benefit scale?

This paper delves into the details of the process we followed to adopt the Sycamore SCE system including all the moments where we were impressed by what we could achieve with such a system and the moments where we felt that the system could be somewhat enhanced further. There were challenges along the way and we addressed them. Moving to any new system requires change management and re-learning, which is never easy for any organization. However, once that learning was done we started seeing the benefits.

THE EXISTING ENVIRONMENT

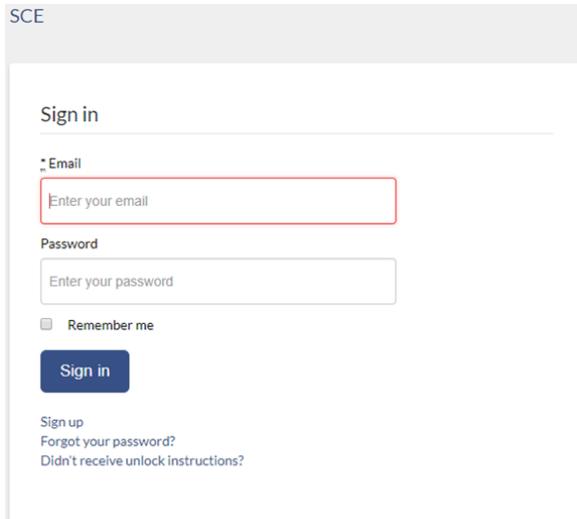
Our organization has been providing statistical analysis and reporting services to several pharmaceutical companies and other larger CROs. Projects that were executed within our organization and delivered to our clients used a Linux-based system that users remotely connect to via a VPN connection. The server ran SAS 9.4 and document management was done using Microsoft Office files such as MS Word and MS EXCEL with collaboration accomplished through SharePoint, accessed by all team members. Transfer of raw data and deliverables was done through a secure FTP setup. Administration of the environment was done by the IT administrator of the organization working in conjunction with the managers and study team leads. The folder structure to be used for all studies followed the relevant SOP unless it was necessary to make exceptions for a specific study. All programming and administrative tasks followed the organizational SOPs.

MIGRATING TO THE NEW SCE

We were provided with the URL of the new cloud-based SCE. This link could be accessed using any Internet browser. Display 1 shows the main login screen to the system, which is very similar to contemporary web-based interfaces. Every user was invited via email to the new SCE before they could login. To start with, the new SCE allows for two types of users to login – the foremost being the indispensable administrator and the second being a study owner. All other types of users had to be explicitly defined in the system before they could be invited.

THE FOLDER HIERARCHY

The first step was to set up the folders for the projects and studies. We referred to our SOP related to project and study folder setup to create a folder hierarchy similar to the hierarchy in our existing system. The new SCE allows a 2-level hierarchy (project and studies within a project) as well as a 3-level hierarchy (sub-projects within project and studies within sub-project). We chose the 2-level hierarchy since we have multiple studies for each client. This 2-level hierarchy is depicted in Figure 1. The new SCE provides a standard set of folders for every study and these folders are created when a new study is created. These standard folders include at the top level – programs, data, results, utility and documents. It also allows new folders to be created within the sub-folders of the study-level folders. The new SCE offers a unique way of defining folders – when they are created, folders are categorized into a specific folder-type. This allows access restrictions to be applied to folders of the same type without applying them to individual folders one by one. For instance, if a certain folder in every study is to have restricted access, it is useful to have that folder to be defined of a specific type so that explicit restrictions need not be applied to that folder every time a new study is created.



Display 1: The new SCE login screen

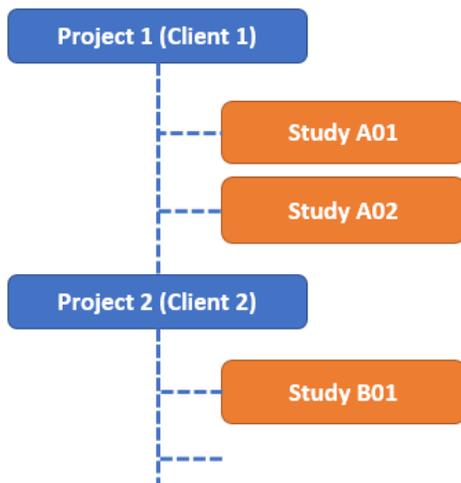


Figure 1: The Project-Studies Hierarchy

Once a study is created, the new SCE allows new studies to be created with a single click under any project using an existing study as a template.

THE NEW SCE SECURITY MODEL

With the folder structure in place, it was time to define how users in our organization would access the new SCE. Every user that would log in to the new SCE to access a study would do so in a specific role. The subtle difference between roles and users is shown in Figure 2.

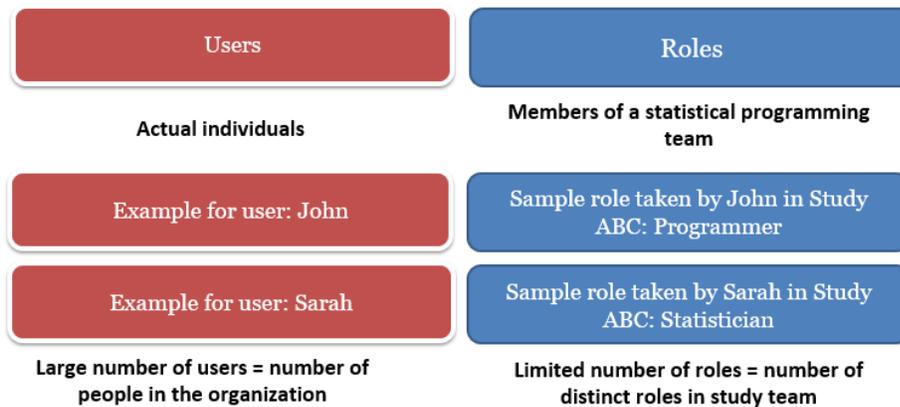


Figure 2: Roles and Users

The role of a user would decide what the user would see (access control) and what they would be allowed to do (privilege control) in the study. In a well-designed setup, there would be a limited number of roles in the system and these roles would map approximately to the typical members of a study team. Examples of roles are *Statistical programmer*, *Study manager*, *Biostatistician* etc. Based on the typical teams we used for study deliveries and consistent with our SOPs, the following roles seemed to fit the bill:

- Statistical Programmer
- Lead Programmer
- Study Creator (Manager)
- Statistician

Since security in the new SCE is enforced in two dimensions – the access-control dimension and the privilege-control dimension, it was important to carefully analyze our existing SOPs and the various roles in our programming teams to understand how best they could fit into the new SCE. As an example, in our organization, the statistical programmers (the role of *Statistical Programmer*) are not allowed to have write access to raw data in a study which is typically loaded by the study lead (the role of *Lead Programmer*). This would mean that the raw data folder of every study should be set to have read-only access for the *Statistical Programmer* role while having read-and-write access to the *Lead Programmer* role. Contrary to the more tedious approach of restricting access to specific folders as used in conventional systems, the new SCE makes setting access restrictions to folders less taxing using the concept of folder-types. In the above example, by providing read-only access of the *raw_data* folder-type to the role of *Statistical Programmer*, a restriction is effectively applied to any statistical programmer trying to write to the raw data folder of any study. The access-control matrix of role vs. folder-type we implemented in the new SCE for our organization is shown in Figure 3.

Folder Type	data	<u>data_rw</u>	programs	documents
Role				
Statistical Programmer	R	RW	RW	R
Lead Programmer	RW	RW	RW	RW
Statistician	R	R	R	RW

R=Read only access RW=Read & Write Access
*Not all folder types used in the SCE are listed

Figure 3: The Role x Folder Type Access-control matrix

The second dimension of privilege control is handled as part of the role definition and decides which role can perform what action in a study. For instance, not all team members in a study can invite new programmers or statisticians to a study. This privilege should be held by a study manager or a lead programmer role. Similarly, assigning programming tasks to team members is a privilege held by the lead or senior statistical programmer role. The new SCE also has program execution as a privilege that may not be given to a reviewer type of role. Some of the typical privileges in the new SCE as we set it up in our organization is shown in Figure 4.

Privilege	Create a program	Create new study	Create a workflow	Invite a new user to SCE	Create a new role
Role					
Admin	✓	✓	✓	✓	✓
Study Owner	✓	✓	✗	✗	✗
Statistical Programmer	✓	✗	✗	✗	✗

Figure 4: The Role x Privilege Privilege-matrix

All roles defined in the system were accessible globally and in addition, the new SCE allows for definition of study-specific roles in case the global roles do not meet the requirements of a study. Study level access control is integrated into each study and this allows the study in-charge (in our case the lead programmer) to provide study access to new study team members. Study level access control includes rights to create new study-specific roles at any point in the study.

During the definition of a new global role or a study-specific role, the system provides the creator of the role the option to decide what actions the new role can perform. This menu has options to decide whether the new role will have access to the unblinded area of a study, can execute programs, invite new team members to a study and manage tasks, which will be discussed later in this paper.

WORKFLOWS IN THE NEW SCE

Workflows mimic real-world tasks in the SCE so that the execution of these tasks are recorded in detail in the SCE as part of the audit trail. Before commencing on delivering new studies, we found that the new SCE encourages setting up of workflows for typical activities including SDTM/ADAM programming, TLF Programming, statistician’s review activities, and other document management activities done as part of study execution that involve specific roles performing various activities in sequence over several steps. There is a foolproof link between workflows and the roles that process them ensuring greater compliance.

To re-create our real-world workflows in the new SCE required some planning. The new SCE allows the definition of individual steps in a workflow and allows associating roles to each step. Individual output artifacts in a study such as reports, datasets or documents each become an instance of that workflow that is completed by following the steps in the workflow. An example is shown in Figure 5.

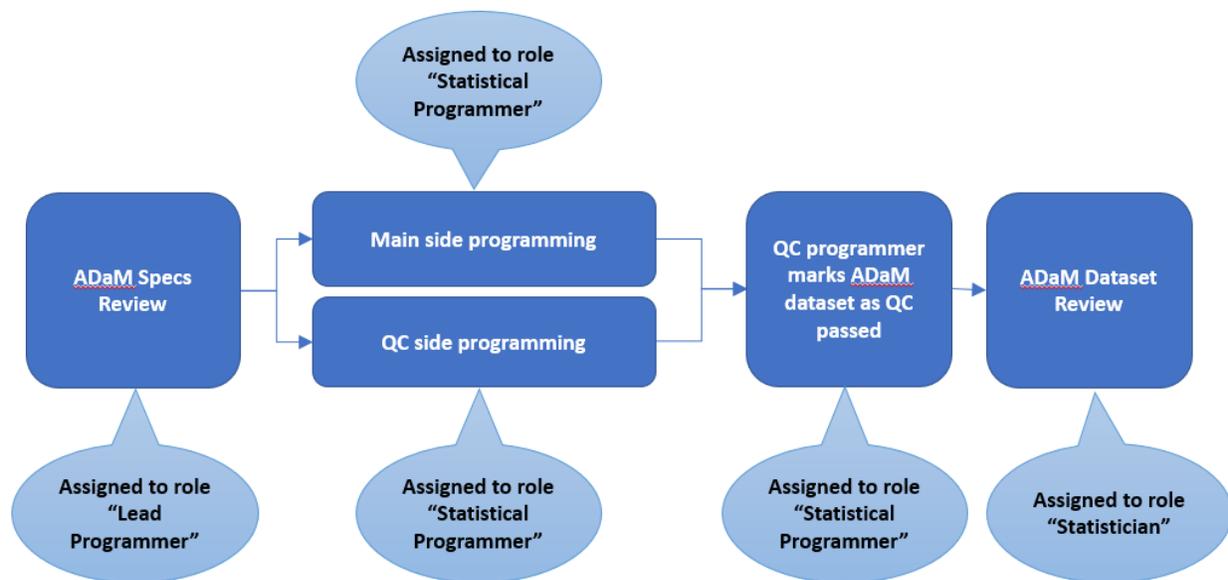


Figure 5: A sample ADaM creation workflow

The ADaM workflow could be mimicked in the new SCE in several ways, limited only by one’s imagination. All workflows can be defined only by the administrator and are globally available to all studies. Workflows in the system consist of single and double steps, with a double-step available for activities that can be done in parallel such as main side programming and QC programming. The system allows individual steps in the workflow to be assigned to specific users that are already assigned to the study. Note that these users must be assigned to the study in the specific role allowed by that step in the workflow.

THE PROGRAMMING ENVIRONMENT

With the folders, roles and workflows in place, it was time to move to the center stage of the programmer’s playground – the programming interface. To the statistical programmer, the programming interface plays a key role in ensuring productive coding and prompt turnaround of output artifacts. Beyond just making the code “work”, the programmer should have quick access to program versions, understand invisible dependencies and have knowledge of the available global macro libraries.

Program Editor and Versioning

Program editing is a simple matter but considering the amount of time spent by programmers in the program editor, it is imperative that editors are simple, have intuitive menus and are prompt to respond at the speed of the programmer’s mind. The new SCE offers an inbuilt editor that boasts a host of features including one-click version control, a visual dependency tracking aid and a click-and-run functionality. The commonly used menu options in the program editor include a program run button, check-out/check-in, file version history, file renaming, job history and dependencies.

In conventional systems, versioning of programs is often accomplished using one of a variety of available tools ranging from the simple in-built RCS in UNIX to advanced third-party software such as Visual Source Safe. We found that versioning in the new SCE is inbuilt for all files and folders of every study. Every new file creation or file upload automatically creates the first version of that file. File updates made using a simple Edit button followed by a Save, creates the next version of the file. Differences between various versions of a file can be viewed with the help of the version history menu. File differences are limited to text and SAS program files.

Dependency Management

In conventional systems including ours, dependencies are most often tracked in the programming status tracker where the dataset listed first is the one that is run first, followed by the rest of them in the order

listed. Once all datasets are re-run, tables and figures are run next followed by the listings. This is easily error-prone since re-runs must be tracked meticulously at all steps in the process to ensure up-to-date output artifacts. Dependency tracking in the new SCE includes warning flags indicating outdated outputs and a graphical display of the inputs and outputs of every program. Color-coded warning flags helps a programmer quickly decide if a program needs a re-run for a new input. Programs re-runs can also be set to trigger with an update in any of the inputs.

The AUTOEXEC and Global Macro Libraries

Access to folders within a study for assignment of SAS LIBREFs was accomplished using relative paths within the system. We set up an AUTOEXEC file that could pick up the current location of the program and identify the relative path to the important LIBREFs such as the folders containing raw, SDTM and ADAM data sets.

To provide access to our global macro library, we created a top-level project that could be accessed from any study in the system. The AUTOEXEC contained the SASAUTOS path to the global macro library to allow for macro calls without the `%include` statement.

SAS-Studio® Integration

The new SCE allows for integration with SAS-Studio for program editing and interactive development with the version control mechanism allowing for check out in the SCE, editing and saving in SAS-Studio, followed by check in of the updated program back in the SCE.

DOCUMENT MANAGEMENT

Study documents that were managed previously in a disconnected SharePoint environment were moved to the documents folder within each study. The documents folder is also write-controlled for specific roles in the system but otherwise needed no separate setup. The new SCE has an inbuilt viewer for text, .PDF and .CSV files but .RTF, MS EXCEL and other files need to be downloaded for viewing. Updates made to these files followed by an upload creates a new version of the files.

EXECUTING THE PILOTS

With the setup complete, a few pilot study teams trained on the use of the system and some of the SOPs updated, it was time to execute a few pilot studies starting with the transfer of the raw data until creation and delivery of the TLFs. The global roles that were defined for the programming team were available in all studies and the workflows defined for the programming tasks could be used to assign tasks to the study team members. The study documents were placed in the documents folder of the respective studies. When a programmer logged into the system, the programmer was only able to see the studies they were assigned to in the dashboard.

STUDY LEVEL ACCESS CONTROL

All the pilot studies we ran worked well with the global roles in the system. Study-level access control was handled by the lead programmer.

TASK MANAGEMENT

After loading the raw data, the first step was to complete the SDTM programming. This was done using the SDTM workflow. A task was created for each of the SDTM outputs to ensure that the SDTM workflow would be followed for those tasks. These tasks were then assigned within the SDTM programming team. The system sent out email notifications to users for each task assignment. As and when the programmer completed a step in the workflow, they would update the task with the status and the task would move to the next step in the workflow and this was repeated until the task was completed. For each step completion, the user assigned to the subsequent step received an email about the pending action for the step. A task board in the system shows details of the status of all the current and past tasks. Through a color-coded, visual representation of each workflow, the task board allows the viewer to see the status of

each task and the currently active step of each task in its workflow. This allows easy differentiation of yet-to-be-started, ongoing and completed steps in each task.

For the next stage of programming activities including ADAMs and TLFs, we learnt that we could upload a programming tracker in a specific format that could be used to populate the tasks automatically. In cases where TLF mock issues and data issues affected the report outputs, these could be tracked and closed within the individual tasks. The status of all the tasks in the study was available to be downloaded in a PDF format at any time.

THE AUDIT TRAILS AND ACCESS HISTORY

The new SCE tracks actions performed at all levels in the system starting at the folder level, to the study and project level and until the global level. Every action of every user including file creations, file modifications, file views, program runs, study assignments and task status changes are recorded in the system. This audit trail is easily accessed and exported to a PDF or CSV file using an audit trail menu visible to all users. The system also records a separate access history that is accessed from the access control portion of every study.

CONCLUSION

This paper summarizes our experience with migrating to a new secure cloud-based SCE that offers one-step login and several features that are very relevant to the way we conduct our clinical trial programming activities. After being provided with access to the system, we began with the setup of the underlying framework that had to be customized to our SOPs. A couple of pilot studies were then picked up to be executed in the system. The overall experience of using the new system is best described in terms of some of the challenges encountered and the key highlights.

CHALLENGES ENCOUNTERED:

- Every new system comes with its learning curve and this new SCE is no different. However, given the intuitive generic web-based interface, the learning curve was not too steep, and the environment was adopted by our programming teams after some training.
- The lack of inbuilt viewers in the new SCE for popular formats such as RTF, EXCEL etc. had to be handled with the use of a parallel explorer interface such as WinSCP to enable seamless throughput in programming activities.
- Fitting real-world workflows into a digital system is often not a straightforward task, given the intricacies of the tasks themselves and how we have been doing them all along. In our case, it required some tweaks in organizational SOPs and minor work-arounds in the process to achieve a good fit. This has the potential to be easier if the workflows are designed suitably.
- Document management in the system is slightly different from the file versions maintained for programs and text files. With no options provided for check-in and check-out of documents and workbooks, the only way to create a new version would be to download the existing one, make the necessary updates and upload it back in the same folder. This has the downside of not capturing cumulative updates made by multiple users editing a single file simultaneously.
- We are yet to ascertain how well the system would work alongside standard enterprise tools used in the industry including Pinnacle21, JReview etc.

KEY HIGHLIGHTS:

- It is hard to deny that the level of detail provided by the system audit trail can be invaluable in situations that require tracing back a couple of thousand steps to understand any unintended access.
- The role-based security model makes life easier for the administrators and simplifies a lot of processes such as execution of workflows. Tying workflows tightly to specific roles in a study allowed for greater compliance in our study execution. In addition, the study-level access control facilitates easier management of study team members by the respective team leads avoiding the long-wound

IT-dependencies.

- Our programmers appreciated the availability of a visual aid for dependency management and used it extensively for cross-checking program inputs and outputs and their status.
- There is no doubt that the single-point sign-in is a welcome relief and a sign of things to come.
- Integrating task management into the study keeps the focus of all team members on the tasks needed to be completed for a deliverable and ensures smoother tracking of individual tasks
- Providing access to users outside the domain is very straightforward in the new SCE, thanks to the simple email address-based login. This is invaluable in cases where a client wishes to access and review certain deliverables or documents.

In summary, SCEs like this cloud-based SCE we adopted have the potential to replace our age-old computing environments with modern intuitive interfaces that, are as sophisticated or even more advanced than the gadgets we use every day. While the system does come with its challenges, over the long run as we use it more, the more we seem to like it better than our previous setup. And over the long run, its many benefits could potentially out-weigh the risks and challenges that are faced during the initial learning curve and efforts in setup.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Tyagrajan Swaminathan
Ephicity Lifescience Analytics
tyagrajan.swaminathan@ephicity.in

Sridhar Vijendra
Ephicity Lifescience Analytics
sridhar.vijendra@ephicity.in

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 trademarks of their respective companies.