

SCE Alchemy: The Return of the QC Process, and Rise of Training via LLM

Steve Griffiths, GSK, London, UK
Eleanor Sparling, GSK, London, UK

ABSTRACT

METEOR (Multilingual Environment to Transform, Enable, and Optimise Reporting), GSKs new language agnostic SCE solution powered by Domino Data Science Platform with GitHub Enterprise integration has now been widely adopted by study teams. In this final part of the trilogy, we will share how we've overcome performance issues whilst striving to deliver BAU (Business as Usual), incorporated new functionality and automation to improve the QC process, and explored LLM to bring our training documentation and delivery into the 21st century.

INTRODUCTION

In early 2020, a three-year mission was formally launched by GSK Pharma Biostats in partnership with R&D Tech to modernise clinical trial reporting. The aim was to leverage both modern technologies and methodologies to create a new Statistical Computing Environment (SCE) which would:

- Deliver Business Efficiency
- Provide Enhanced Capability
- Enable Faster Analysis & Reporting
- Simplify Ways of Working

Very early on in this mission, GSK announced a plan to integrate the Rx and Vx R&D units, and the new SCE, now known as SPACE (Statistics & Programming Analytics Computing Environment) was seen as the vehicle to deliver a single, multi-lingual, unified data analytics platform. Almost simultaneously there was a move from GSK IT to develop into a 'product-orientated DevOps organisation', structured around the Scaled Agile Framework, which for the majority of the team meant a whole new way of working.

Following a search of the SCE market and a Proof of Concept (POC) in 2021, Domino was chosen to power our new SCE, which was renamed METEOR (Multilingual Environment to Transform, Enable, and Optimize Reporting), and since it was already used within the "Statistical Data Sciences" group, GitHub was selected as the new code management system, with an appropriate branching model suitable for the development (and subsequent push to production) of study artifacts.

The Minimal Viable Product was delivered in the summer of 2022, and in mid-2023, the scale up of adoption and associated user support, and the migration of legacy studies commenced in earnest. METEOR was moving away from the minimal viable product phase and its' usage was considered business as usual (BAU).

BUSINESS AS USUAL

END OF A LEGACY

With all new studies now using METEOR and migration from legacy systems well underway (if not complete in the case of HARP), decommissioning work started.

Not all studies required migration; others had to be archived. Programmers and statisticians are very often under tight timelines, with many studies on the go at once. It's not difficult to see that sometimes the archival tasks often fall by the wayside, and there were many studies (with multiple milestone deliveries) which needed archiving. Users were instructed to prepare these studies and deliverables for archiving.

During the lifecycle of a study, there may be occasions when the study outputs become sensitive; early unblinding for an IDMC is a prime example. Access is restricted via linux groups applied either at the study or milestone (reporting effort) level. These groups needed to be removed before the support team could perform the final steps to archive the studies. However, even though time had elapsed, the team couldn't assume that the study output was no longer considered 'restricted'. Each 'group' was reviewed by therapeutic programming heads with the required project knowledge, and eventually permission was granted to remove the restrictions, thereby allowing the archival of these impacted components. With everything archived, it was now possible to revoke user access, decommission the URL and make the linux servers 'read-only'.

Running concurrently with the archiving was the removal of the legacy support system. Training courses were identified marked for deletion, as were the SME distribution list and the user support pages within the GSK intranet.

AUTOMATION

The move to BAU has also seen an improvement in the number of automated processes, especially with respect to the mounting of SDTM data and controlled access to 'restricted' Domino projects.

Previously, users would create Domino projects, and then complete request forms (helpdesk tickets) for SDTM teams to mount SDTM datasets (blinded or unblinded) to these projects. Project programmers / statisticians would also complete forms listing personnel to be granted access to restricted unblinded SDTM data; lists which were then re-approved by DBAs without study knowledge.

It is also possible for GSK study teams and external partners to transfer study artifacts (data, programs, output) via a staging area and blob containers, which can be viewed by both parties via Microsoft Azure Storage Explorer. Whilst partners need to upload and download using functionality within the Storage Explorer, GSK team members can click the appropriate button within the METEOR UI to either move data directly from Domino into the staging area, or vice versa.

A similar approach is used to push Biostats data downstream for use internally within other GSK applications within and external to SPACE.

PERFORMANCE

Users can run programs in batch, via a python script called 'multijobs'. This takes a user defined configuration file as a parameter, which itself is a list of SAS and/or R programs including dependencies, however the execution runtime is not desirable for BAU; users are used to running multiple programs in a manner of minutes.

Within METEOR, these multijob runs can take considerably longer to complete. Jobs are not sent to the queue in bulk, but rather in smaller groups and each job then waits in the queue to execute. Each program spins up its own program container which takes processing time, and it's only then that the program can run.

Changes have recently been made to the system architecture which include improvements to Azure Kubernetes Service nodes / clusters, and the introduction of hardware tiers specifically configured to optimise SAS. An updated version of multijobs has also been released. Whilst it's unlikely we'll ever remove the overhead of operating in the cloud, this newer version allows users to submit jobs 'locally' in a single container as opposed to spinning up a separate SAS or R container for each program. The initial job still needs to be added to a queue, but once running all the jobs contained within the called config file execute without the need of queueing. When running 'locally', users can also run multiple jobs in parallel further reducing overall runtime.

In addition, this new multijob approach also supports file-based dependencies. Users can supply a list of input and output files for each job, and if the output files are missing or pre-date the input files then the program is executed, otherwise it is by-passed.

TRAINING AND DOCUMENTATION

Using METEOR appears daunting at first glance. There are new concepts in addition to the new way of working. It's not just learning the METEOR processes, but those associated with Domino and GitHub too. Hence the mandatory training items required for access to the SCE.

Curricula	Training Item	Training Document
Mandatory for Access	METEOR Overview	METEOR Overview Module
	Intro to GitHub	Intro to GitHub video
		Getting started with GitHub Enterprise Cloud
		Clinical Study Reporting GHE User Guide
	GitHub for Studies	Template repo – Instruction level detail in README
		Create a study repo video
		Branching Workflows Video
	Domino Basics for Analysis & Reporting	Domino Basics for Analysis & Reporting Document
	METEOR Program Development	METEOR Program Development Document
	Training Study Exercise	METEOR Training exercise

Following feedback from the userbase, these initial training requirements are getting streamlined so the Domino and GitHub components are 'essential for the job', but not mandatory for access to METEOR.

However, in addition to these initial trainings, there are a plethora of further trainings (either via 'read and understand' documents or short video vignettes) which users must undertake depending upon their role within the study team, a selection of which is displayed below.

Hopefully the contents of some of these trainings is obvious; 'Running Multi-Program Jobs' does what it says on the tin as it's quite niche with its content. Users are encouraged to bookmark the site for training materials as opposed to downloading and saving local copies of documentation for the simple reason that for certain topics the information (and working practice) is updated and improved frequently.

Role Based	Self - service PA Creation	Self Service PA creation
		SDTM Projects creation and data refresh
		Service Now Form
		Instructions to fill Copy SDTM
	Running Multi-Program Jobs	METEOR Running Multi-Program Jobs
	QC process in METEOR	QC Report
		QC Report Explanation
		CX_RUN_Checklist
	ADaM/ OPS Metadata	ADaM Metadata Management for METEOR
		ADaM Specification Template
		ADaM 04 - Working with ADaM specifications
	How to Specify legacy programs in METEOR	SPACIFY Legacy Programs for use within METEOR
	Metadata, Centralised Data and File Storage	Metadata, Centralised Data and File Storage Document
	Exchanging data with External parties	Exchanging Data with External Parties Guide
		Exchanging Data with GSK Teams
	Generation of P21E Report and Define.xml and ADaM SDP content	Generation of P21E Report and Define.xml and ADaM SDP content Document

The vast (and ever increasing) amount of documentation can make it difficult for the user to locate specific information if a refresher is needed. If a user needed to know how to apply a Personal Access Token in order for Domino and GitHub to 'communicate' which document would they locate the information. It could possibly be in either of the two GitHub documents, Domino Basics or even METEOR Program Development.

GenAI & SPACE

Rather than trawl through documents, it would be simpler to use Generative AI to give the answer.

Ignoring the publicly available tools like ChatGPT, Microsoft Co-Pilot or Google Gemini, there are two main flavours of GenAI available within GSK, once the mandatory training about responsible usage has been completed.

Jules is a web-based scientific LLM to help answer document-based, scientific, and general questions and is accessible for everyone within GSK R&D. It has limited knowledge after 2021.

GiGi is a GSK specific LLM, trained on information such as GSK processes, policies and knowledge articles and is a secure environment for using GSK data.

The current training documents can be loaded into either platform and queried by the user. The following table shows a subset of parameters used within the config file for multijobs.

Parameter	Required / Optional	Description
[JOBID]	Required	this is the job name and is reference elsewhere without the brackets. A valid name can be any string that does not contain ']', ',', or '\n'
Command	Required	This must be a valid path to a SAS / R program or a .sh script starting from /mnt/code/
Tier	Optional	Valid values are the name of any hardware tier that the executing user has access to. If not specified, the job will use the project's default hardware tier
Depends	Optional	If present, the job will not start until the specified jobs have completed. Valid values include any other defined JOBID in the configuration file. Multiple jobs dependencies can be included by separating each job dependency with a comma (without spaces). If no dependencies are listed, jobs will run in parallel. Additional validation is performed to ensure no circular dependencies exist.
environment	Optional	Valid values are the IDs of any compute environment that the executing user has access to. Template provides correct Environment ID for SAS, R and QuIT script. If not specified, jobs will run in the default environment for that project.

When this document is loaded into GiGi, the following questions can be asked:

What is the JOBID parameter?

As documented in *METEOR Running Multi-Program Jobs V13.pdf*, the JOBID parameter is required and represents the job name. It is referenced elsewhere without the brackets. A valid name can be any string that does not contain], ,, or \n.

⚠ Double check the citations and outputs prior to using them.

How do I identify the job within multijobs?

As documented in *METEOR Running Multi-Program Jobs V13.pdf*, each job in the configuration file has a unique JOBID parameter, which serves as the job name. This JOBID is referenced elsewhere in the configuration file without the brackets. A valid JOBID can be any string that does not contain], ,, or \n.

To identify a job within multijobs, you can look at the JOBID parameter in the configuration file. Here is an example format:

```
[job1]
environment: 6297c79c84e697145c857132
command: /mnt/code/prod/datasets/job1.sas
```

In this example, job1 is the JOBID for the job.

However, documents loaded directly into GiGi (or Jules) are only accessible to the user loading them. This can be a good way to test information coverage, but it's also a great way to summarise each document with the aim of creating knowledge-based (KB) articles for use within GiGi. It is not recommended to just drop the document content into a KB template as there is a subtle change to the language.

For example, when writing instructions/steps:

- Write in the **present tense**; you should not state what you did or actions that were done but what to do. For example, "Click Start" instead of "After you have clicked Start, then..."
- Provide only the words required to communicate intent. Additional filler phrases are not necessary, you should be direct and concise in writing. For example, "Click OK" instead of "Click on the OK button".

Each document may be split down into multiple KB articles, and could be spread across different categories:

- Getting Started – To provide a general overview or basic information about a process/product,
- Best Practice – Contains best practices when engaging with a process/product,
- How to – Provides step-by-step instructions on how to complete a specific activity,
- Known problems – Informs about known issues and workaround for a process/product. It should include maximum 5 issues.

These KB articles are peer-reviewed and following approval are added to the database, which is used by GiGi to answer queries related to GSK processes.

QUALITY CONTROL

Quality control (QC) has always been an integral part of clinical reporting. The primary purpose of QC is to ensure the accuracy, reliability, and validity of the data and the resulting analyses. In clinical research, where decisions can significantly impact patient outcomes, maintaining high standards of data integrity is not just an operational necessity but an ethical imperative. Implementing robust QC measures helps to identify and mitigate errors in data collection, processing, and analysis, ultimately fostering trust among stakeholders—including researchers, regulatory bodies, and patients—who rely on precise and transparent results.

The importance of quality control extends beyond mere compliance; it plays a vital role in the reproducibility and interpretation of statistical methodologies. At GSK we are implementing a risk-based QC process. By evaluating the complexity, significance, and potential impact of different programs/analyses, teams have autonomy over type of QC required and can therefore allocate their resources more efficiently. Code review, double parameterization and double programming are three types of QC techniques that significantly bolster the reliability of statistical analyses. Double parameterization involves using two independent sets of parameters to validate results. This redundancy helps to confirm that findings are consistent across different analytical frameworks, reducing the likelihood of errors stemming from misinterpretations of the data. In contrast, double programming entails having two analysts independently code the same statistical analysis. By comparing the outcomes, discrepancies can be identified and resolved, enhancing confidence in the results. This method not only serves as a check against human error but also promotes collaborative learning and knowledge sharing among team members. Implementing this risk-based approach fosters a proactive QC culture, encouraging teams to anticipate and mitigate risks before they escalate into significant issues.

Having discussed the importance of risk-based QC in maximising resource effort, we must also acknowledge that in many lights QC can be seen as an additional extra step outside of the production pipeline. QC within our new statistical computing environment must instead flow seamlessly with the production pipeline. Tracking, monitoring, and approving of QC stages must also hold tangible value-add rather than simply being a tick box exercise for audit trails.

At GSK we are exploring multiple possible systems and tools to effectively track QC stages. These have included, Azure DevOps (ADO), GitHub Projects, Jira, and Domino Governance. Each of these has its own unique selling points. Azure DevOps offers a comprehensive project management solution including

integrated pipelines for CI/CD, while GitHub Projects comparatively provides simplicity with direct integration with version control. GitHub in particular provides advanced automation of code quality checks. Jira excels in customizable workflows and detailed reporting with in-depth metric dashboards, making it suitable for agile environments. Lastly, Domino Governance stands out for data science projects, focusing on compliance and direct integration with the production pipeline. Each tool has its strengths, and selecting the right one involves considering factors such as team structure, existing toolchains, and specific QC requirements.

CONCLUSION

Our journey to design and implement a new statistical computing environment continues. We have been agile in our research and development and will continue to be so to meet our goals in the most effective way possible. We have accomplished our goal of achieving BAU, having fully migrated from, in addition to decommissioning, our legacy systems. Whilst 'business as usual' describes a system that can match previously functionality, our SPACE objective is to transform and surpass old ways of working. New QC tracking and automation is just one step towards a seamless integrated pipeline which prioritises efficiencies without compromising quality. Lastly, incorporating the use of GenAI and LLMs is helping to bring our training documentation and delivery into the 21st century. This has already significantly reduced the support and training overhead required and users are able to self-support more effectively.

REFERENCES

AS01: SPACE – Launching GSK Biostatistics Towards New Horizons [SPACE - Launching GSK Biostatistics towards new horizons \(phuse.s3.eu-central-1.amazonaws.com\)](https://phuse.s3.eu-central-1.amazonaws.com/SPACE-Launching-GSK-Biostatistics-towards-new-horizons)

AD05: Further Adventures in SCE Alchemy [PAP_AD05.pdf \(phuse.s3.eu-central-1.amazonaws.com\)](https://phuse.s3.eu-central-1.amazonaws.com/PAP_AD05.pdf)

CONTACT INFORMATION

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

Eleanor Sparling
Email: eleanor.x.sparling@gsk.com

Steve Griffiths
Email: Stephen.j.griffiths@gsk.com