Paper SM09

Advancements in the Evolution of our SMC Story: Process and Technology

Melanie LEVEQUE, Merck KGaA, Lyon, France
Waldemar MILLER, Merck KGaA, Darmstadt, Germany

ABSTRACT

The imperative for expedited Safety Monitoring Committee (SMC) output delivery is crucial in improving drug development processes, especially given the high time pressure associated with creating SMC output.

Our objective was to streamline workflows, improve deliverable usability to accelerate project timelines, and empower professionals to prioritize critical tasks.

We reached these goals through standardized processes with minimal study adaptation, rapid data availability using source data, accelerated result disclosure by eliminating manual result transfer, and improved data insights through dynamic outputs.

Our efforts culminated in the development of an internally created tool that evolved the conversion of SMC classical ASCII TLF format into interactive Patient Profile visualizations in HTML and TLFs directly within PowerPoint slides using advanced features of R. This tool functions as an interactive webpage, offering comprehensive subject information such as Patient Profiles and data summaries, complete with support for filtering, screenshots, and zoom functions.

INTRODUCTION

Safety Monitoring Committee (SMC) assess the safety, efficacy of the Investigational Medicinal Product (IMP) alone and in combination and perform benefit risk assessment to safeguard the interests of study participants. They monitor the overall conduct of the clinical study and compliance with the study protocol to protect the validity of the study. They assess pharmacokinetic and pharmacodynamics-guided biologically effective dosage regimens (PK and PD). They recommend on the continuation of the study, assess on further dose-(de)escalation and schedule, and identify MTDs and RDEs.

SMC members are Medical Lead (SMC Chairperson), Safety Strategy Lead or Delegate, Biostatistics Lead, Clinical Pharmacologist Expert Team Lead, Clinical Biomarker Lead, Coordinating Investigator or Principal Investigator.

SMC members have different roles, skills and expectations for the review of data outputs (Figure 1).

With our previous process, we identified several problems:

- SMC slides were not homogeneous between projects.
- Heterogeneity of involved CROs and involved sponsor staff.
- Time Pressure (Data from last patient were not transferred completely/ may change).
- Error- Prone and time-consuming copy-and-paste process for bringing data on SMC slides.

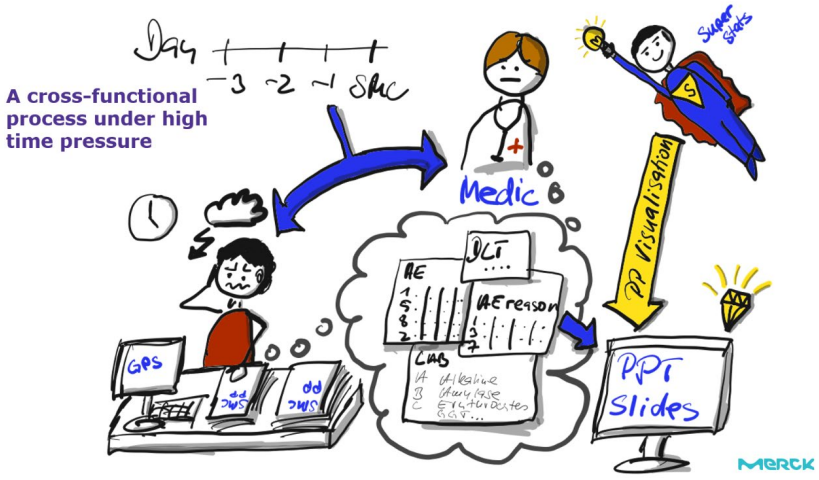

Figure 1: A cross functional process under high time pressure

Review of data outputs is required at every SMC Data Review Meeting and should be scheduled as soon as possible after all participants in a dosing cohort have completed the Dose Limiting Toxicity assessment period (i.e., 21 days) which is defined as the data cut-off and are expected to occur within 5 business days after data cut-off.

Compilation of participant's profiles / statistical outputs and SMC draft slides are prepared very quickly after data cut-off. Our objective was to streamline workflows, improve deliverable usability to accelerate project timelines, and empower professionals to prioritize critical tasks.

APPROACH

After engaging in several discussions with SMC contributors and stakeholders, we aligned on standardized SMC materials requiring minimal adaptations for each study.

This approach allows us to maintain consistency while also being flexible. We chose to work with source data (CDASH) instead of SDTM or ADaM, which helps us save time by omitting the generation and validation of SDTM and ADaM. This change provides quicker access to data for creating our SMC outputs.

Additionally, we have improved the process of result disclosure by eliminating manual result transfer e.g. copy-past results to power-point.

Those consideration were the trigger to develop the aSMC R-package, which automates the creation of SMC slides. Figures 2 and 3 illustrate the result.

Figure 2: Presentation generated by aSMC R-package

…

Figure 3: Presentation generated by aSMC R-package

…

We improved data insights through dynamic outputs by preparing an HTML page (Figure 4 and 5). This file contains Patient Profiles, Listings, Tables and Figures … You can easily filter, select, zoom and capture your flexible reports and add them in your slides as needed.

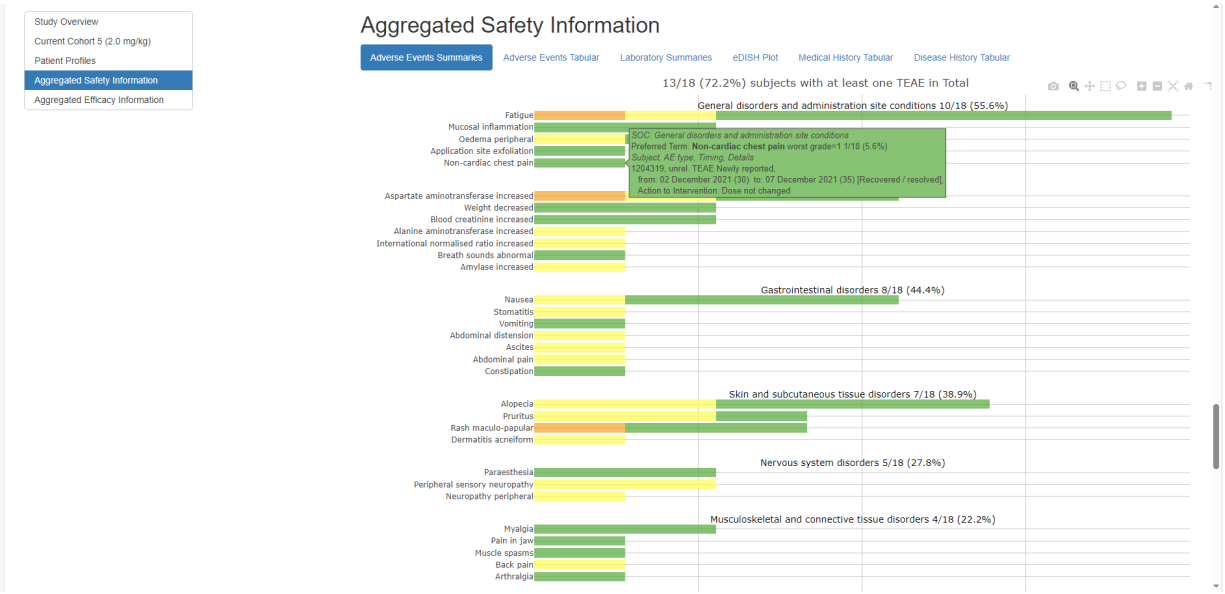

Figure 4 : HTML Safety section example

Figure 5 : HTML Patient Profile section example

With all these improvements, we can deliver SMC material one day after data transfer, SMC core members can review them on Day 3 and 4 and SMC can take place on Day 5 as expected but without stress (figure 6).

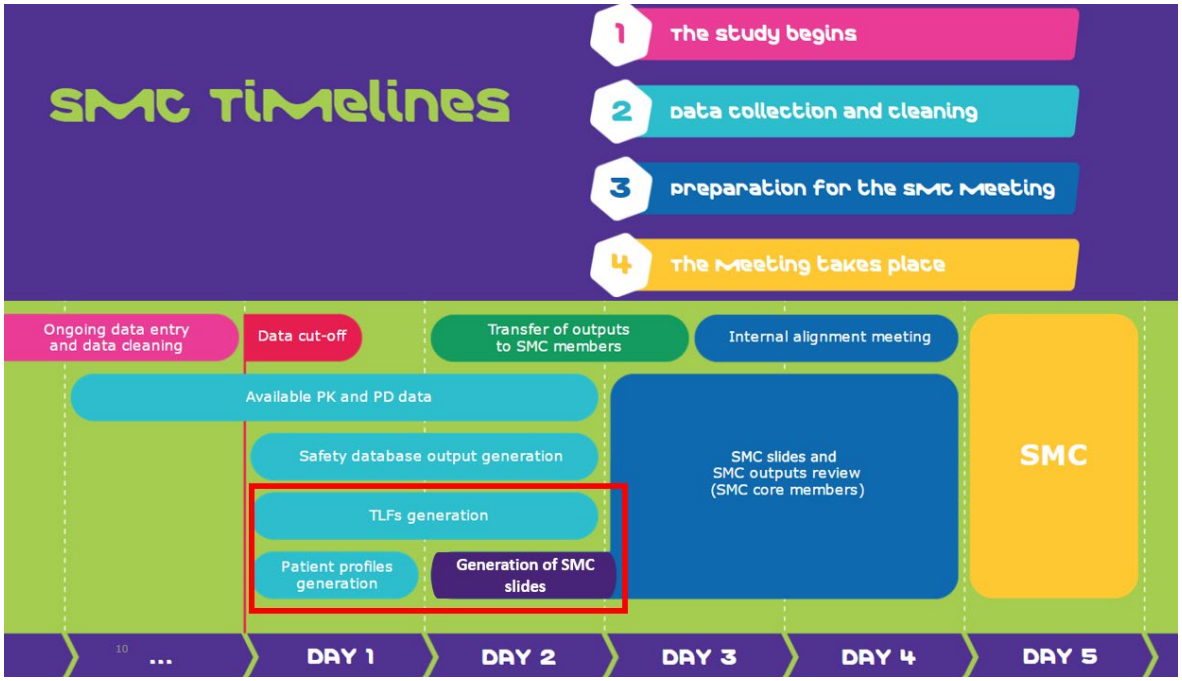

Figure 6: Merck SMC process and timelines

TOOL DEVELOPMENT

Initially, the development of the SMC tool was driven by motivated individuals of the Biostatistics department and Statistical Programming department. During this phase, the goal of this collaboration was to identify opportunities to enhance the experience of the SMC. A prototype was implemented in R, using R markdown to generate PowerPoint slides and HTML outputs. Besides static TLF outputs, the packages `plotly` and `DT` allowed some level of interactivity on the HTML outputs. This proof of concept was considered successful, such that the previously mentioned internal R package `aSMC` was created.

In fact, the idea of automated generation of HTML and Slides for SMC was so successful, that it found great adoption across numerous studies. Along that, the tool evolved and the project-driven "evolution pressure" lead more features, more adoption, but also revealed some challenges, as the SMC tool was outstretched by project needs. The codebase became complex and adaption, maintenance and bug fixing required a sound understanding of the inner workings of the tool. To mitigate this and ensure a future-proof tool, a major refactor of the SMC tool codebase was considered, with three leading questions in mind:

1. How can we reorganize the code and make the tool easier to maintain and bugfix?
2. How can we make the tool easier to use by the project programmer without overloading them with "must-know"?
3. How can we design the tool such that we can reuse it (maybe as a blueprint) for other purposes as well?

During the planning phase of the refactor, we considered some more questions:

4. Can we borrow concepts and ideas from software design and software development?
5. Can we increase the quality of the code, the design of our files and rethink the order-of-operation?
6. How can we deliver tools and features and reduce maintenance in the future?

These considerations regarding the architectural design of the SMC tools lead to a prototype of a "tool framework", which consists of an initialization script and a set of modules. The initialization script performs some housekeeping tasks: Loads packages, loading the framework modules, loading the data model R files and controller R files and composing an R object that acts as the point of interaction with the framework and its modules, data models and controllers. These modules can be considered stand-alone "mini tools" provide technical functionality while encapsulating their complexity. As of now, the following modules are incorporated:

- `DataModel`: Manages the data layer of the tool.
- `HtmlGenerator`: Programmatically generate HTML files.
- `PptGenerator`: Programmatically generate PowerPoint slides.
- `LegacyAsmc`: Make functions from the initial SMC tool available from within the framework.

A mention on the inner workings for the HTML generator, as it is of special interest: After collecting all information to be included into the output (this information may be text, data sets and figures), an intermediary R markdown file will be generated before rendering it.

As we took inspiration from design patterns of frameworks for other programming languages, we considered the separation of the codebase into a data access layer (the 'model') and into business logic (the 'controller') very useful for the kind of tools we intent to create. Hence, the framework requires three kinds of R files to be specified for the tool:

- A "model" R files utilize the frameworks `DataModel` to make data available throughout the tool: Each model file instantiates the `DataModel` object, points it to their respective `Rds` file. Additionally, a 'hotfix' area allows for quick changes when timelines are short.
- A "controller" R file consists of a list that contains functions, i.e. controller methods. These methods act as the "business logic" of the tool: They access data model (or other controllers), perform derivations, calculations, etc., and return an output, that may be used by other controllers or by the 'main' scripts to compile, for example, a report.
- The "main" R file is the script, that - on the highest level - generates an output, an html or ppt file. It initializes the framework, accesses the controllers, and compiles the outputs.

The implementation of the refactor was performed alongside a study, so the overall requirement was to deliver outputs to the SMC as required. As part of our R tech stack, notable packages used in the tool are:

- `R6` – for encapsulation of modules, OOP
- `Rmd`, `knitr` – document generation
- `ggplot2`, `plotly` – figure output
- `tibble`, `kable`, `flextable`, `DT` – tabular output

In addition to framework features and assemble useful controllers for reporting demographic data, patient profiles, safety listings, etc. In designing the data models, we opted for a relational data representation that is somewhat close to the CDISC ADaM, however, we did not purse completeness and correctness according to the standard. Rather, we favor a data layer that we are already familiar working with, but which is also fit for purpose – that is – deliver HTMLs and PPTs and not a submission to authorities. Focusing on this data representation, the actual derivation of data is

excluded from the tools' core, and now poses its own – preceding – task. With this, we anticipate feeding the tool from multiple sources of varying data maturity (CDASH, SDTM, ADaM) without touching the internals of the tool. Apart from that, we gain operational flexibility, as the derivation task does not require expert knowledge of the tool itself and is performed completely outside the SMC tool.

Overall, eight data models were defined:

- `AdverseEvents`,
- `ConcomitantMedication`,
- `DiseaseHistory`,
- `Dose`,
- `LaboratoryData`,
- `MedicalHistory`,
- `Subjects`, and
- `VitalSigns`.

Various controllers were implemented. These controllers could either be associated with a specific data model, or 'themed' controllers. Model-extending controllers such as the `SubjectsController` or `AdverseEventsController` provide methods that aim to add functionality associated with their respective data model. For example,

- `SubjectsController$all_subjects()`,
- `SubjectsController$subjects_on_treatment()`,
- `SubjectsController$enrolled_subjects()`, and
- `SubjectsController$safety_analysis_set()`

return vectors of subject IDs corresponding to the appropriate group, and

- `AdverseEventsController$summary_of_adverse_events()`,
- `AdverseEventsController$summary_of_adverse_events_toxgr_GE3()`, and
- `AdverseEventsController$overview_SAE()`

return data frames listing the adverse events of all toxicity grades, toxicity grades of 3 or higher, and serious adverse events, repectively. As an example for 'themed' controllers, the

- `StudyOverviewController$TrialStatus()` and
- `StudyOverviewController$Disposion()`

generate output-ready R objects (flextables, in this case) that can be directly integrated in the output compilation.

Finally, two 'main' programs were set up, one to generate the HTML output, and one to generate the PowerPoint slides. During runtime, both programs initialize the same framework setup, therefore having access to all data models and controllers simultaneously. Here, one somewhat important but unfamiliar concept must be mentioned: Inversion of Control, we utilize this concept by implementing controller methods. When calling the controller method, framework generates the output on-the-fly, which is, in fact, unanimously true for controller methods.

In retrospective summary, the refactor of the SMC tool led to the idea to create a framework, which is a major step in achieving high levels of automation. As the further development of both the SMC tool is ongoing, a backlog of features waits for its implementation, further enhancing functionality and programmers experience. We look forward to extending the codebase of the SMC tool to adjacent scenarios, such as DMC reporting, and integrating the tool within a bigger landscape of static (e.g. PDF/RTF) and interactive (Shiny) reporting.

CONCLUSION

The development of our SMC tool represents a significant step forward in supporting safety monitoring in clinical trials. As an interactive webpage, it provides comprehensive patient information, including profiles and data summaries, along with features for filtering, screenshots, and zoom functions that facilitate efficient reviews.

The automated SMC slide creation process enhances content quality and accelerates preparation, ensuring successful execution of SMC meetings by delivering targeted data insights. This streamlined approach not only boosts efficiency but also empowers our teams to make informed decisions faster and accurate to ensure the safety and well-being of our patients.

Looking ahead, we are excited to explore further improvements and adapt according to demands of our teams and leveraging the technology to expand. aSMC R-package has transformed our reports into interactive, user-friendly resources and insight sharing among teams. Ultimately, these advancements aim to achieve our goal to bring more of the right medicines to the right patients faster.

REFERENCES

- https://policymanual.nih.gov/3014-503
- FDA Guidance for Clinical Trial Sponsors: Establishment and Operation of Clinical Trial Data Monitoring Committees (March 2006)

CONTACT INFORMATION

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

Melanie LEVEQUE
Associate Director Statistical Programmer, Merck KGaA
melanie.leveque@merckgroup.com

Waldemar MILLER
Senior Principal Statistical Programmer, Merck KGaA
waldemar.miller@merckgroup.com