Paper SM04

The ABC of Managing a Standards Macro Library

Amit Bhaniani, MSD, London, United Kingdom
Carl Herremans, MSD, Brussels, Belgium

ABSTRACT

SAS® macro is a powerful programming language to develop dynamic programs. It enables the end-users to customize various attributes of the output tables, listings, and figures (TLFs) based on the value of macro parameters. While developing and maintaining high quality macro programs as a standards macro library is a common practice in statistical programming, efficient management of a vital resource like macro library is not an easy task. Three main challenges in the management of standards macro library are: (1) adding to and updating macros in

the standards macro library to address changing business needs; (2) ensuring adherence to best programming practices and departmental SOPs, and (3) fine-tuning communication channels established for end-users, stakeholders and sponsors of the standards macro library. This paper shares some management tools and techniques to illustrate how a standards macro library can be managed efficiently.

INTRODUCTION

Management of a standards macro library requires multi-dimensional approach and cross-functional collaboration. Individual elements to help manage the library can be broadly classified into three levels: Active Roles & Responsibilities, Building the Standards Macro Library, and Communication (*Figure 1*). The first level represents identifying active team members to support the library e.g. statisticians, statistical programmers, stakeholders, sponsors, and macro library management. The second level represents building the standards library i.e. adding new and updating existing macros, to address changing business needs. The third level represents communicating feedback and status updates within macro development team, collaborating on business needs and promotion newsletter with stakeholders, developing and delivering user training, and providing performance metrics to help sponsors forecast workload and allocate resources to support library development.

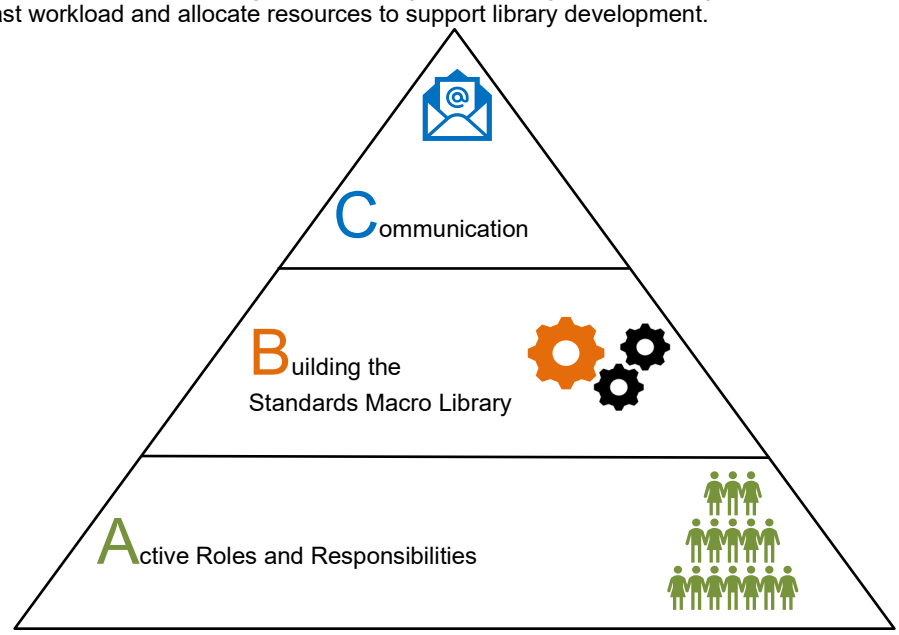

Figure 1 - Management of a standards macro library

Positive collaboration amongst its team members is required for successful management of the library. A detailed RACI matrix (Responsibility assignment matrix 2021) documents individual roles who support the library, detailed activities which establish everyday functioning of the library, and activity specific responsibilities which are expected from these roles. This promotes clarity and efficiency required to manage such a resource intensive project. The RACI matrix is a vital document referenced by the team to help them understand the scope of their roles subject to

various activities performed. In addition, the departmental SOPs and guidelines also help them appreciate how these role-specific activities should be performed.

The most dynamic aspect of a macro library is building and maintaining one. To accomplish this challenge, the management team uses the scrum framework (Scrum (software development) 2021) (Hughes 2015). The library manager assigns resources (statisticians and statistical programmers) to various macro development and validation activities and keeps the user community informed about status updates for individual macros in the library. Management team also encourages knowledge sharing via guidance documents, smart design (Redner Ginger 2007) and best programming practices amongst the programming team to promote development of robust and dynamic macros.

The final challenge for the management team is to establish and maintain multiple communication channels. These channels are used within the macro development team, and between individual roles supporting the library to:

1. accept feedback and requests for updates to existing and development of new standard macros, review and approve these requests, and share status updates with the development team.
2. communicate the promotion newsletter for updated macros to end-users and conduct training courses for statistical methodology and business-critical macros.
3. share accomplishments of the macro library with stakeholders and sponsors, and report performance metrics to the sponsors for efficient resource allocation and workload estimation.

ACTIVE ROLES AND RESPONSIBILITIES

We used the functional framework Scrum (Dootson 2014) to define roles and responsibilities for our standards library. It provides functional structure required to support management of the library, like a scaffolding. A well-constructed framework clearly defines the key roles and assigns them strategic responsibilities to accomplish business-critical tasks efficiently. This framework is extended to include detailed explanation of responsibilities to help all team members understand the scope of their role. Finally, a well-documented Standard Operating Procedure (SOP) is a blueprint to help all these roles accomplish their designated responsibilities with high quality and compliance.

TEAM MEMBERS AND STAKEHOLDERS

Scrum is a framework to help address complex adaptive problems while delivering high-quality outputs. It's designed for projects where team members split individual assignments into tasks, referenced as Product Backlog Items (PBIs) in Figure 2*,* These tasks can be completed within a time-boxed iteration or sprints, which typically last a couple of weeks. Scrum is especially useful for implementation of agile processes in a highly regulated environment, like the pharmaceutical industry. If designed and implemented correctly, scrum can be used to address some of the key management challenges e.g. quality control, safety, efficiency, traceability, and validation. Some of the key components of scrum are roles, values, and activities, and some of the key roles defined in the scrum framework are:

- Product owner = individual or group of individuals who owns the requirement specifications and gives directions on what to implement.

- Stakeholders = community of statistical programmers, statisticians and consumers of the macros present in the library and are represented by the product owner.

- Scrum master = classically known as the library manager who ensures the project is running smoothly. The scrum master validates the business needs with the product owner, ensures that backlog of work gets correct prioritization, and approves production deployment.

- Development team = Team members made up of developers and testers who implement the code, complete validation, and update documentation.

These definitions are role dependent, i.e. applicable to the tasks or activities performed and not dependent on the individual(s) performing them, e.g. product owner for one macro can be a stakeholder for another macro. The strategic positioning of all these roles within a typical scrum framework is shown in *Figure 2*. Scrum master is the only role which extends across all macros in the library. The scrum master is responsible for continuously prioritizing time-boxed iterations (or sprints) to process updates of existing, and addition of new macros in the standards library. All other roles are subject to the activities performed within the scope of individual macros.

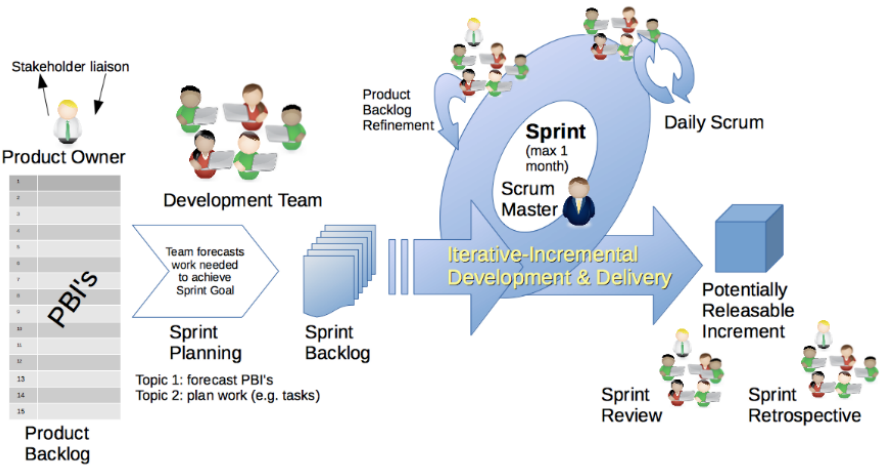

Figure 2 - Scrum framework (Dr Mitchell 2021)

ROLES AND RESPONSIBILITIES

To ensure smooth interaction between all roles defined in a scrum framework, it's important that they understand their individual responsibilities and how they collaborate with each other to support the macro development process. A responsibility assignment matrix (RACI) is an efficient tool to document key roles and their responsibilities for completing individual tasks or sprints (Responsibility assignment matrix 2021) in cross-functional or departmental projects and processes. Four key responsibilities typically defined in a RACI matrix are:

- R = Responsible
- A = Accountable
- C = Consulted
- I = Informed

	Stakeholders	Product owner	Scrum master	Development team
Request for a new macro or update to an existing macro	A,R	I	I	
Work order approval	C	A,R	C	
Work assignment		I	A,R	I
Work order execution		C	C	A,R
Production deployment	I	I	A,R	R

Example of a RACI matrix for supporting a typical macro library.

MACRO APPROVAL PROCESS

Once the functional framework, with individual roles and responsibilities is defined, the next step is to review and approve request to develop a new standard macro. The most critical task in the approval process is translating this request into a work order, defined and prioritized to address business need. This is accomplished by the collaboration between the scrum master and product owner. Once the work order is defined and prioritized, it's split into multiple chunks of macro(s) by the scrum master and assigned to an analyst/developer to translate into a smart macro design (Redner Ginger 2007). In a pharmaceutical context it's recommended to present this design to the product owner and/or a representative of the stakeholders to ensure their business needs are correctly understood. After the code is developed and tested, it's recommended to re-visit this code and outputs with the product owner and stakeholders to make sure that the code meets the specifications approved by the stakeholder and outputs meet expectations of the product owner. This interaction encourages development of high-quality macros for the library and initiates the macro development process, which is discussed in the next section.

BUILDING THE STANDARDS MACRO LIBRARY

Compliance and traceability of the comprehensive macro development process, including validation, and timely promotion of macros are essential for building a stable standards library. The process of developing a new macro or updating an existing macro (*Figure 3*) starts when the scrum master assigns the approved work order to an analyst/developer. This is followed by comprehensive documentation, development work, and validation of the macro by its development team before it's checked for compliance and deployed to production by the scrum master.

Figure 3 – Standard macro development process

FOUR STAGES OF VALIDATION PROCESS

Validation of the developed macros ensures that they support business needs while using correct statistical methodology and smart programming design (Redner Ginger 2007). It also helps to assess that these macros can address expected data scenarios and create desired outputs. Validation process for individual macros in the library undergoes the following sequential stages:

1. The product owner, scrum master and analyst/developer collaborate on the Programming Requirement Specifications (PRS) document, explaining programming logic and statistical methodology used, sample output created by the macro, its revision history, and version updates.

2. The analyst/developer implements the approved specifications to develop the requested macro with complete testing for macro parameters, data conditions and output structure.

3. An independent statistician tests statistical macros (e.g. macro to create hazard ratios for time to event analysis) for the choice and use of statistical methodology and accuracy of statistics calculated.

4. Another member of the development team completes independent review of the macro. This review ensures that the macro is developed according to the specifications document and checked for all possible data scenarios and macro parameter values.

5. A programmer creates a sample code package for the macro i.e. a self-contained folder structure with sample data, input macros and programs required to generate sample outputs. This sample package is part of user documentation and it helps end-users understand the scope and efficient use of the macro.

Using a guidance document with detailed explanation of all validation activities will ensure consistent and high-quality macros developed for the library. Steps 1 to 4 are reported in detail in a separate paper (Snow Fu and James Wu 2004).

TRACEABILITY OF MACRO DEVELOPMENT PROCESS

The scrum master is responsible for maintaining traceability of the macro development process. The development of a macro is a project which requires support from its development team of statisticians and statistical programmers. It may be easy to keep all the team members well-informed about various completed and on-going activities for a single macro. It can be a challenge managing this information for multiple macros with different levels of complexity and validation stages. A simple spreadsheet (*Figure 4*) structure can easily capture validation activities along with responsible team members and completion dates, making it easy to assess the macro status, and plan for its promotion.

	A	B
1	Derivable Category	macro library
2	Deliverable	a0hazard0ratio
3	Priority/Tier	1
4	Version	7
5	Programming status	09_ready for promotion
6	Status flag	Bugfix
7	Additional descriptions	Increased precision of p-value.
8	Statistician Assigned to	statistician 1
9	PRS Assigned to	statistician 2
10	PRS Assigned Date	01/Dec/2099
11	PRS Target Date	05/Dec/2099
12	PRS Actual Delivery Date	07/Dec/2099
13	PRS Approved	07/Dec/2099
14	Programmer Assigned to	programmer 1
15	Assigned Date	07/Dec/2099
16	Start Date	07/Dec/2099
17	Target Date	09/Dec/2099
18	Actual Delivery Date	10/Dec/2099
19	Independent Testing Programmer Assigned	programmer 2
20	Independent Testing Assigned Date	10/Dec/2099
21	Independent Testing Start Date	10/Dec/2099
22	Independent Testing Target Date	12/Dec/2099
23	Independent Testing Actual Delivery Date	12/Dec/2099
24	QC Sample Code Programmer Assigned	programmer 3
25	QC Sample Code Assigned Date	13/Dec/2099
26	QC Sample Code Target Date	15/Dec/2099
27	QC Sample Code Acutal Delivery Date	15/Dec/2099
28	Initiated	01/Dec/2099
29	Promoted to production	

Figure 4 - Macro project plan

	A	D	R	W	AA	AC
1	Derivable Category	Version	Actual Delivery Date	Independent Testing Actual Delivery Date	QC Sample Code Acutal Delivery Date	Promoted to production
2	a0hazard0ratio	7	10/Dec/2099	12/Dec/2099	15/Dec/2099	31/Dec/2099
3	<macroname1>	1				
4	<macroname2>	2				
5	<macroname3>	3				

Figure 5 – Macro library project plan

The scrum master should also use a macro library project plan (*Figure 5*) for a quick overview of the library development and selection of fully validated macros suitable for promotion. More advanced/complex project management (e.g. Jira (Nandigama 2014)) and Source Code Management tools (e.g. Git (Philp 2012) (Peters Amy 2020), Subversion) can also be used, but the underlying principle remains the same.

PROMOTION OF STANDARD MACROS

Checking the documentation for completeness and validation for compliance of fully validated macros is essential, before promoting them to the standards library. A script can easily check various validation activities for compliance by comparing information from macro library project plan (*Figure 5*) against macro validation programs, output files generated and their timestamps. Some of the compliance issues checked by this script should be:

1. The existence of a Program Requirements Specifications (PRS) document saved and approved before the corresponding macro has finalized its development and validated.
2. The existence of sas/log/lst files with filenames as per the conventions pre-specified by the scrum master.
3. The development work and validation activities are completed in the required sequence e.g. developer testing completed before statistical review; sample code package prepared after independent review etc.
4. A complete rerun of the testing programs for each validation stage to ensure backward compatibility of the updated macros.

Once the periodic batch of promotion ready macros are cleared by the compliance check report, scrum master releases them to the standards repository and updates their status in the macro library project plan (Figure 4). The scrum master also creates a macro promotion newsletter which is shared with the development team(s), product owner(s), stakeholders, and sponsors, to inform them about latest updates in the library. This forms an important element of the communication strategy, which is discussed in the next section.

COMMUNICATION

A well-planned communication strategy is the central nervous system of the macro library. It relies on successful collaboration between scrum master and other key roles defined in the scrum methodology using two-way channels (*Figure 6*). The scrum master uses feedback on existing macros, collaborates on development of new standard macros, and shares status updates within the macro development team(s). The development team and stakeholders collaborate with scrum master on improved macro requirements to better address their business needs, macro promotion newsletter to know about latest macro updates, and specialised training courses for complex macros. The scrum master uses the macro library accomplishments and performance metrics to forecast development work and plan resource requirements for discussion with the sponsors. Answers to the following three questions can provide useful insight to plan communication strategy for the library:

1. **Who** is the recipient?
2. **What** is the key message?
3. **How** frequently it is communicated?

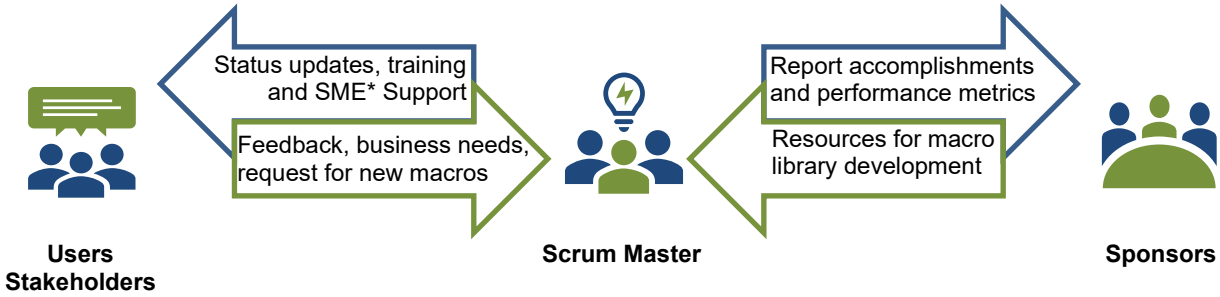

Figure 6 – Communication Strategy for Macro Library
* Subject Matter Expert (SME)

FEEDBACK COLLECTION

Feedback is essential for the management of a macro library where the scrum master collaborates with development team(s) and stakeholders on development of new macros to address evolving business needs. Important feedback on bugs, enhancements, usability, and efficiency of existing macros in the library is collected from users and stakeholders. This feedback is implemented to ensure that the developed macros are user-friendly, efficient, and support business needs. Communication strategy to support feedback collection can be planned considering:

1. **Who** is the recipient: *Product owner and scrum master*
2. **What** is the key message: *requests for new macros and user feedback on existing macros*
3. **How** frequently it is communicated: *on-going basis*

It's important to document and share user feedback regularly and incorporate it into everyday functioning of the library. This feedback can be documented in a spreadsheet (*Figure 7*) along with review comments and approval status from the scrum master. Once approved, these requests are translated into work orders and assigned suitable priority before they are incorporated into the macro library development plan.

	A	B	C	D	E	F	G	H	I
1	Macro Name	Enhancements / New Macro description	Requester	Date Requested	Review Comments	Approval Status	Priority Assigned	Target Completion Date	Completion Status
2	a0hazard0ratio	Increased precision of p-value.	statistician 1	01/Dec/2099	Important to implement for a specific project deliverable	Approved	High	15/Dec/2099	Completed
3	<macroname1>								
4	<macroname2>								
5	<macroname3>								

Figure 7 – Feedback for existing macros and requests for new macros

PROMOTION NEWSLETTER AND USER TRAINING

Communicating the latest macro library updates is as important as developing macros for the library. Macro library newsletter is an important communication tool as it contains information on enhancements and bugfixes for the latest macros deployed to the standards repository. This information is used by project teams to select latest version of standard macros for timely completion of their project deliverables. Communication strategy to support macro promotion newsletter can be planned considering:

1. **Who** is the recipient: *development team(s), stakeholders, and sponsors*
2. **What** is the key message: *new macros and enhancements/bugfixes for existing macros*
3. **How** frequently it is communicated: *Periodically e.g. once a month*

Periodically the scrum master generates a promotion newsletter using an automated macro which combines data from macro library project plan with macro library metadata, for macros identified as ready for promotion (Figure 4). This newsletter is shared with macro library team members, stakeholders, and sponsors to keep them informed of the latest macro library updates.

Developing high-quality macros based on complex and novel statistical methodology is useful only if the programming team and stakeholders are trained to use them efficiently. Comprehensive training courses, user documentation and SME support promotes efficient use and quick adoption of such macros. To accomplish this, scrum master collaborates with stakeholders to manage network of SMEs who compile user documentation, develop and deliver required training courses, customised for the experience level of the team members. These training courses also provide the opportunity to collect user feedback on programming issues and possible enhancements while the development team is updating the standard macro. The product owner or the development lead for the new macro is an ideal candidate for the role of SME.

PERFORMANCE METRICS

Continuous development and maintenance of the library requires high level of resources which can only be justified by equally high performance of the standard macros and quality of their outputs. This is where sponsors play a vital role as they provide guidance and resources to support the on-going development and upkeep of the library. Periodically, scrum master reports accomplishments of the library to stakeholders and sponsors, and performance metrics to forecast development work and plan resource requirements for discussion with the sponsors. Sponsors evaluate these metrics (*Figure 8*) against the objectives, priorities and performance standards set for the library and to allocate resources for the next tenure of the macro library management team.

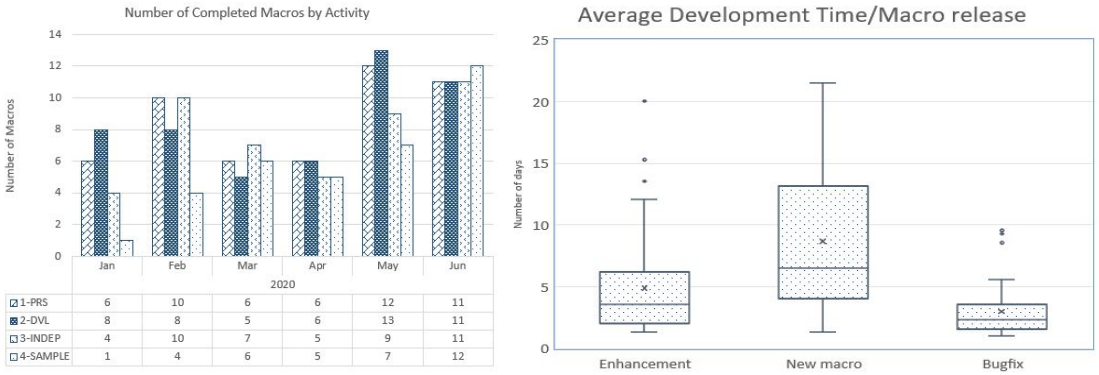

Figure 8 – Sample performance metrics for the macro library

DISCUSSION

The successful management of a standards macro library relies on the structure of its development team, and the key roles and responsibilities defined in its scrum framework. Efficient execution of these responsibilities and transparent communication from conception of a macro to its release to the standards repository, help build a robust library to address business needs. In this section, we will briefly touch upon some of the concepts, tools, and techniques explained throughout this paper.

A standards macro library requires continuous management support and cross-functional collaboration. Scrum is the methodology which helps construct the functional framework to define key roles and their strategic responsibilities. The clarity in defining these roles and responsibilities is essential for positive collaboration amongst them. A well-documented SOP is a blueprint to help all these roles accomplish their designated responsibilities with high quality and compliance. A thorough understanding of their individual responsibilities is important to avoid conflicts within the macro development team. A positive collaboration is important between all stakeholders, especially scrum master and product owner to define and prioritise work order(s) aligned with business needs.

Bringing compliance of the validation activities, traceability of the development process, and timely promotion of the macros within a single functional framework is essential for its success. The process of developing a new macro or updating an existing standard macro starts when the scrum master assigns the approved work order to an analyst/developer, followed by collaboration between the product owner and stakeholder. Detailed specifications document is the output from this collaboration which is assigned to a developer for implementation. This is followed by documentation and validation of the macro by the development team, to ensure that it supports business needs while using correct statistical methodology and smart programming design. It also helps to assess that these macros can address expected data scenarios and create desired outputs. Finally, scrum master checks the promotion ready macros for compliance before releasing them to standards repository.

Communication **of** the latest macro library updates is as important as developing macros in the library. It relies on successful collaboration amongst the key roles defined in the scrum methodology. Communication strategy involves collecting feedback on macro usability and enhancements, and sharing status updates on development, updates, validation, and promotion of the macros. It includes planning and delivering training courses on complex statistical methodology and business-critical macros leveraging on SME support network for wider adoption and rapid deployment. It also includes reporting of performance metrics to the sponsors for estimating macro library workload, and efficient resource planning for the next tenure of the management team.

Managing a Standards Macro Library is a learning experience, an opportunity to implement well-established methodologies and test innovative techniques to address changing business needs. Some of the lessons learnt while developing and managing this resource, and recommendations are:

1. It's important to ensure that the standards library is embedded into the business process, which acts as a bridge between the library and its business environment. This approach streamlines business integration of the macro library.

2. The user community should consider the library as their toolkit, supporting their day to day jobs and not a standard toolkit imposed on them. This can be achieved by including the community into the development of standards and having their buy-in.

3. The library can easily be extended to include other standards e.g. standard template macros, ADaM specifications etc. to cover all aspects of project delivery and address business needs.

Developing and maintaining a standards macro library is not a daunting challenge provided the elements identified as: A (Active Roles & Responsibilities), B (Building the Standards Macro Library), C (Communication), and explained throughout this paper are defined, implemented, and executed in collaboration amongst scrum master, product owner, development team, stakeholders, and sponsors of the macro library. A high-quality standards library is a resource intensive activity which pays itself back through high quality outputs and shorter turnaround times for projects. This is an invaluable business asset and a valuable learning experience.

REFERENCES

Dootson, Alistair. “Agile Project Management in a Regulated Environment.” *PHUSE 2014.* 2014.

Dr Mitchell, Ian. “Own work, CC BY-SA 4.0.” *Wikimedia.* 06 July 2021. https://commons.wikimedia.org/w/index.php?curid=44894952.

Fu Snow, Wu James. “Organizing and Building a Centralized SAS® Macro Library.” *Proceedings of the 2004 Pharmaceutical SAS® User’s Group Conference, Paper AD11.* 2004.

Hughes, Troy Martin. “Toward Adoption of Agile Software Development in Clinical Trials.” *PharmaSUG 2015 – Paper AD13.* 2015.

Nandigama, Sameera. “SAS in clinical trials – A relook at project management, tools and software .” *PhUSE 2014.* 2014. Paper AD07.

Peters Amy, Zimmerman Danny Whiteis, Grace, Flynn Joe, and Polanski Stan. “Git for the SAS® Programmer: Using Source Control to Organize Your Code and Collaborate with Others.” *SAS Global Forum 2020.* 2020.

Philp, Stephen. “Programmers, An Introduction to Git Version Control for SAS.” *Westren Users of SAS Software.* 2012.

Redner Ginger, Zhang Liping, Herremans Carl. “Techniques for Developing Quality SAS® Macros.” *Proceedings of the 2007 Pharmaceutical SAS® User’s Group Conference, Paper TT10.* 2007.

“Responsibility assignment matrix.” *Wikipedia, .* 06 Jul 2021. https://en.wikipedia.org/wiki/Responsibility_assignment_matrix.

“Scrum (software development) .” *Wikipedia.* 6 jul 2021. https://en.wikipedia.org/wiki/Scrum_(software_development)#Sprint.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the contribution and continuous support of the statisticians and statistical programmers who worked on development and regular updates to the standard macros in our library. We also thank our stakeholders for their honest and regular feedback and sponsors for their guidance and support to ensure continuous progress for the library.

CONTACT INFORMATION

Amit Bhaniani
MSD
Pancras Rd 2, Kings Cross,
London N1C 4AG [United Kingdom]
Tel.: +441992452504
E-mail: amit.bhaniani@msd.com
Web: http://www.msd.com

Carl Herremans
MSD
5 Lynx binnenhof
1200 Brussels [Belgium]
Tel: +3227766305
E-mail: carl.herremans@merck.com
Web: http://www.msd.com