

## Auto-validation of SAS® Macros through Regression Testing

Madhu Annamalai - Technical Lead, Statistical Programming, Algorics

Umayal Annamalai - Assistant Manager, Statistical Programming, Algorics

### ABSTRACT

Validation of SAS® macros is a critical step for reporting quality outcomes. Typically, statistical programmers spend extensive time and effort in validation of macros. Furthermore, frequent code changes demanding repetitive manual validation make it even more resource intensive and affect quality of outcomes. Regression testing is a key step in agile product development. Its benefits have been established with a proven record of accomplishment of testing automation in many other fields. Although, extremely limited content is published on the application of regression testing in SAS Macros validation. In this session, our goal is to demonstrate the comparison between both validation of SAS Macros manually versus auto-validation through regression testing and the efficiency in time and efforts for validation. We will be highlighting to you our experience where automated regression testing is possible for most of the standard macros. This will be done simultaneously through test program execution by generating a report to confirm existing functionality is working as expected by comparing the current output against the validated output. From this session, key takeaways are the use of regression testing techniques for the SAS Macros validation, learning from our experience of having done this and how this can be implemented.

### INTRODUCTION

Quality, consistency, and efficiency are the key factors for clinical study reporting, reducing the time for regulatory submission. SAS Programming is used widely for Clinical study reporting and has numerous inbuilt advanced functionalities for effective programming and reporting. One well-known advanced functionality is SAS macros which significantly reduces the programmer's time & effort needed. In addition, metadata-driven automation can be triggered based on certain data points, making SAS macros more robust. However, validating SAS Macros plays a substantial role in regulatory submissions for accurate reporting. While SAS has the provision to build the framework for macros validation, additional efforts might be needed to automate complex macros for programming efficiency and reduce the programmer's time and effort.

### Current validation process

Currently, the SAS validation process includes 1. creating a series of test cases for clean and dirty scenarios 2. SAS macros will be validated using test cases by manual approach (run & check). The purpose is to validate that each macro feature performs as expected. Fig 1 represents the current validation approach being followed.

Step # 1: Test Plan and various test cases created suitable for each macro feature.

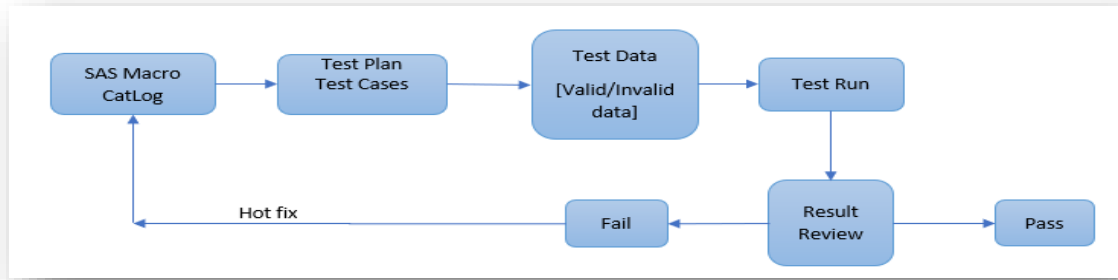
Step # 2: Test Data preparation based on the test cases.

Step # 3: Test execution (Run) based on the test cases.

Step # 4: Result Review (Pass or Fail).

Step # 5: If it fails, bugs are fixed and repeat the Step # 3 & Step # 4.

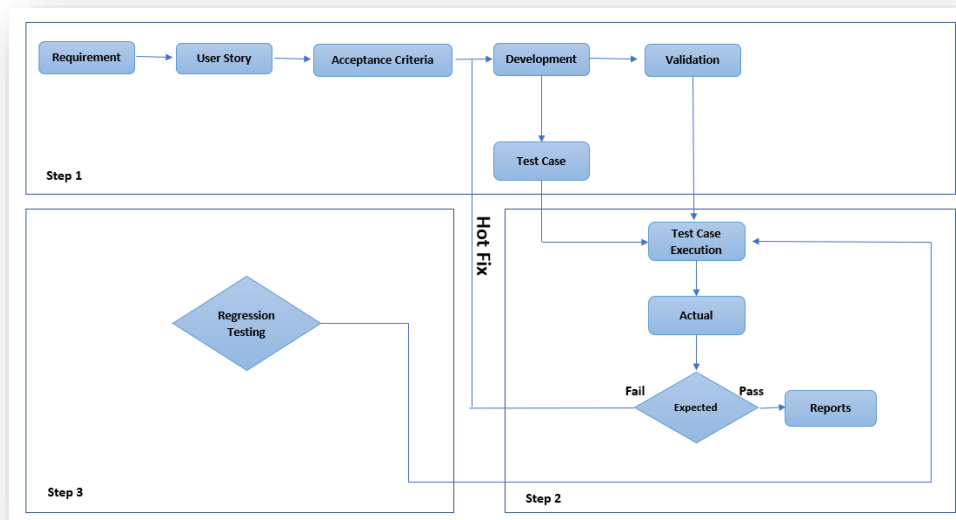
Figure 1: Current Validation approach



### Our Validation approach via Regression Testing

Regression testing is a well-known validation concept in the product development lifecycle that ensures features perform as desired for any code changes/updates/enhancements. Therefore, existing system functionalities will have stability. Our current approach is to leverage regression testing for SAS macros through automation. Figure 2 outlines a framework for the Regression testing process.

Figure 2: Regression testing Framework



**Step 1:** Based on the requirements, SAS Macros will be created, and the corresponding test cases will be generated for initial validation.

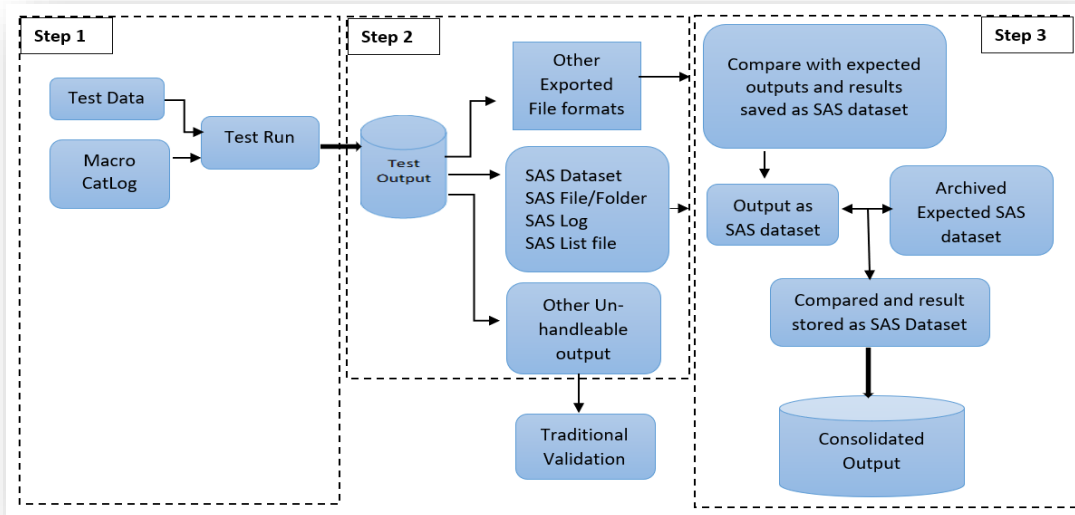
**Step 2:** The test cases are validated based on the tested macro catalog and test data. The initially validated outputs are archived as expected for further regression testing, a one-time process for all features.

**Step 3:** After hotfixes, the macros are re-validated to confirm the existing functionalities within the macro framework. With the revised macros, testing procedures will be auto validated, and the outputs will be generated for review.

## AUTO-VALIDATION VIA REGRESSION TESTING

Figure 3 denotes the Auto-validation regression testing workflow and iterative into 3 different steps.

Figure 3: Auto-validation via Regression testing



### Workflow:

- **Step 1:** The test execution for the hotfix to the macros is performed based on regression testing framework that was set up based on figure 2 and figure 3. With the revised macro catalog and the same testing environment, the test execution is performed.
- **Step 2:** The test outputs generated as a part of Test run are of two types. One that is compatible with Auto validation where we will be able to automate the regression testing. The other one is based on some outputs where we will have to follow the current approach where the regression testing might not be compatible, but we can facilitate the review of such outputs in traditional methods.
- **Step 3:** The current test execution generates the test results in SAS dataset formats. The previously validated SAS dataset outputs will be used as a standard expected results as the reference for comparison of results. The generated test results are compared against the validated archived test results to get compare reports, log summaries and overall testing status for the programs. The compare reports saved as a SAS dataset is further used to consolidate the results based on module level, macro level, program level, log level consolidation summaries as discussed in table in next page.

### Use Case example: Auto-validation using Regression testing.

The regression testing framework is initiated based on the below requirements:

- SAS macro will be created for environment set up with program names and folder structures.
- SAS macro will be created for the execution of test cases and to collect the corresponding results.

- SAS macro will be created to examine the logs and results.
- SAS macro will be created to compare the current result with the expected results and save the compared information in a dataset.
- SAS macro will be created to consolidate the macro level and program level results into the excel file as per the testing plan.

With the successful execution of SAS Macros which was set up for Regression testing, we can achieve the following outputs:

1. Macro level Testing status:

Macro_Name	Validation Type	Testing status	More Description
Macro_1	Automatic	Pass	Passed
Macro_2	Manual	Fail	Failed: <compare mismatch summary>, <Log summary> Manual: Manual check of xx file required

2. Program level status with relevant messages for status identification:

Test case	Macro_Name	Validation Type	Overall testing status	Execution status	SAS output	SAS log	More Description
Test case 1	Macro_1	Automatic	Pass	Executed	Compare matched	Clear	Passed
Test case 2	Macro_1	Automatic	Pass	Executed	Compare matched	Clear	Passed
Test case 1	Macro_2	Automatic	Fail	Executed	Compare mismatched	Issues found	<compare mismatch summary>, <Log summary>
Test case 2	Macro_2	Manual	Manual check needed	Executed	NA	Clear	Manual check of xx file required

3. Log issues:

program_name	macro name	logissue_type	logissue_count	logissue	logissue_rownumber
Test_case1.sas	macro_2	NOT FOUND	1	NOTE: The file xxx.xxx (memtype=CATALOG) was not found but appears on a DELETE statement.	225
Test_case1.sas	macro_2	ARGUMENT TO FUNCTION	4	NOTE: Invalid argument to function INPUT at line 41 column 90.	7098
Test_case1.sas	macro_2	INVALID ARGUMENT	4	NOTE: Invalid argument to function INPUT at line 41 column 90.	7098
Test_case1.sas	macro_2	_ERROR_=1	4	FIRST.values_name=1 LAST.values_name=1 _vovar_num= _voadd_vorder=1 _ERROR_=1 _N_=647	7100

## CONCLUSION

Time & efforts invested in establishing the process for the automation of test cases and its testing will be a great deal to SAS programmers. The repetitive testing and testing burden for a different version of SAS macro programs can be significantly reduced via the auto-validation process through regression testing. In addition to reducing time & effort, the quality output can also be driven by such automation. Therefore, bringing in the regression testing process in SAS Macros validation would enhance the process and enable faster regulatory submission.

## REFERENCES

- Fewster, Mark. Graham, Dorothy. 1999. *Software Test Automation: Effective use of Test execution tools*. Addison-Wesley. ACM Press Series
- Mo, Albert. 2018. "Developing, Managing, and Evaluating a Standard SAS Macro System" *Proceedings of the SUGI 2018*, 018–031.
- Available at <https://support.sas.com/resources/papers/proceedings/proceedings/sugi31/018-31.pdf>
- Karafa, Matthew. 2011. "Building Better Macros: Basic Parameter Checking for Avoiding "ID10T" Errors." *Proceedings of the SAS Global Forum 2011, Coder's Corner*, 096–2011.
- Available at <https://support.sas.com/resources/papers/proceedings11/096-2011.pdf>

## ACKNOWLEDGMENTS

We would like to thank Mukesh Babu K, Head of Clinical & Data Operations and Syed Yusuff Hussain, Head of Product Development for support and guidance.

## RECOMMENDED READING

- *SAS® 9.4 Macro language: Reference*
- *Software Testing, A Craftsman's approach*
- *The Next Step: Integrating the Software Development Life Cycle with SAS® Programming*, Gill

## CONTACT INFORMATION

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

Umayal Annamalai  
Assistant Manager, Statistical programming,  
Algorics  
Email: [Umayal.Annamalai@algorics.com](mailto:Umayal.Annamalai@algorics.com)

Madhu Annamalai  
Technical Lead, Statistical programming,  
Algorics  
Email: [Madhu.Annamalai@algorics.com](mailto:Madhu.Annamalai@algorics.com)

Any brand and product names are trademarks of their respective companies. ® indicates USA registration.