

Process and Tools for Assessing Compliance with Standard Operating Procedures

Ginger Redner, Merck & Company, Inc., Upper Gwynedd, PA
Eunice Ndungu, Merck & Company, Inc., Upper Gwynedd, PA

ABSTRACT

In our efforts to develop drugs and vaccines that improve and save lives, we must maintain the integrity of patient data. This is accomplished by ensuring compliance with SOPs (Standard Operating Procedures) for developing and validating programs used to generate output for the Clinical Study Report (CSR) and submission deliverables. We have developed a suite of macros to help study teams ensure their deliverables are compliant with our SOPs. The main compliance macro identifies and reports specific areas of non-compliance. Another assesses the use of previously developed and fully validated standard macros. Using these macros greatly reduces the risk of error in reporting, decreases the number of SOP steps required by the project programmers to generate the deliverables, and increases efficiency by saving programming time. This paper discusses the macros and the process followed to ensure consistency and mitigate deviations.

INTRODUCTION

We must have a high degree of confidence in the quality and integrity of the results submitted to agencies for the approval of drugs and vaccines. Having a Standard Operating Procedure (SOP) for developing and validating programs used to create tables, listings and figures for the Clinical Study Report to be included in submissions to regulatory agencies, assures that we are complying with regulatory requirements and departmental policies. By documenting our program development and validation activities, we are able to prove that we did what we said we would do in the SOP. However, this being a manual process, it is easy to overlook some steps when working within tight timelines. We have developed a set of macros to assist our programmers and statisticians in assuring their work is compliant with our SOPs and that our standard macros are used as intended, by all study teams.

REQUIREMENTS

According to our SOP, programming requirement specifications and a developer testing checklist must be completed for every program. Individual specification documents are required for all programs which produce analysis datasets and for all project specific macros. Critical programs must be listed in a critical program memo by the statistician and are expected to undergo double programming. All output must be approved by the study statistician and documented in a study tracking file. A final checklist must be completed and reviewed by the project programming manager and lead statistician, and dates of review must be documented in the final checklist. Study teams are also expected to take full advantage of the standard macro library by using as many standards as possible. Creating study specific versions of the standard macros is highly discouraged as they may not undergo the same high level of validation as the standards, and they may increase variation in methodology and presentation of results across studies.

TRACKING STANDARD MACRO USE

We have a large library of standard macros which have been developed over the past 15+ years. These macros are fully validated and expected to be used by all project teams to increase accuracy, maintain consistency and decrease programming and validation time. In the past, we had no way of determining which teams and individuals were using the standard macros, or to what extent. As the standard macro library continues to grow, it becomes increasingly important to be able to identify which macros are heavily used and which are obsolete. This information helps the standards group in prioritizing macro updates, focusing on macro training and promotion, and learning more about the needs of the project teams.

We determined the best way to obtain the most accurate information on standard macro use was to develop a tracking macro (%update0usage0log) to write a record to a SAS dataset each time it's executed. A call to this tracking macro was then inserted into a subset of our standard macros to track each time the standard is run. Figure 1 displays the flow of programs used to track our standard macros.

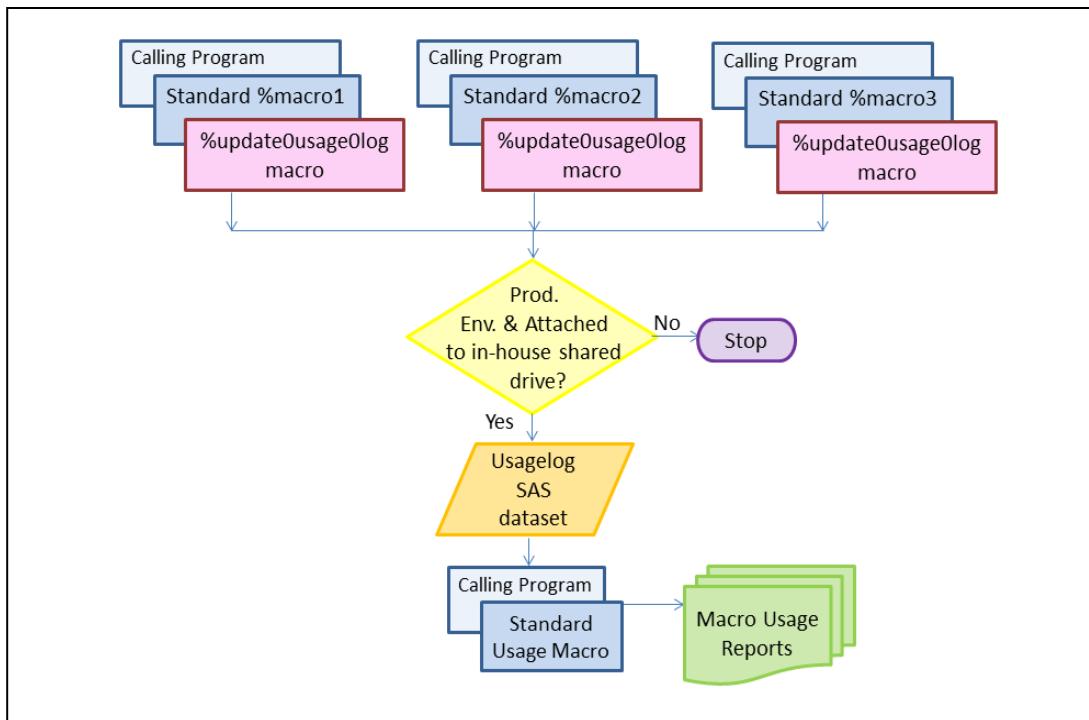


Figure 1, Program Flow for Tracking Macro Usage

THE TRACKING MACRO

The %update0usage0log tracking macro is fairly simple as only one parameter is passed to the macro and that is the name of the standard macro from which the tracking macro is being called. One observation is appended to the usage log dataset each time %update0usage0log is executed. Only calls executed in the production environment are tracked, as tracking macro use when programs are being developed doesn't provide useful data. A check for this is included in the tracking macro.

Appending to a single SAS dataset presented a challenge for us as there are many programmers located world-wide who execute our standard macros, and a risk of contention exists when multiple users attempt to access the same SAS dataset concurrently in write mode. To mitigate this risk, a SAS/Share server is setup in our UNIX environment and the tracking dataset to be updated is stored on the shared server. SAS/Share enables multiple users to simultaneously read and write to a single SAS dataset. Programmers and statisticians across all the study teams have access to this shared tracking dataset.

A sample listing of the tracking dataset is shown in Display 1. The dataset contains the user ID of the person executing the standard macro, the project and protocol the user is working on, and the date and time the program was executed. This data enables us to easily determine which macros are being used, the frequency of use, and the teams and individuals using the standards. Teams often call one macro multiple times using different sub-populations as input. When this occurs multiple records are written to the tracking dataset to record each time the macro is executed, as can be seen highlighted in grey in Display 1.

| USERID | MACRONAME | PROJECT | PROTOCOL | DATE | TIME |
|----------|-----------------------------|-----------|-------------|-----------|----------|
| zhouham | convert0sdtmpplus0data | v507-9val | 002 | 20AUG2013 | 12:29:34 |
| anksapur | gr0disposition | mk994-tum | prot096-csr | 30AUG2013 | 2:05:48 |
| anksapur | gr0disposition | mk994-tum | prot096-csr | 30AUG2013 | 2:06:21 |
| anksapur | gr0disposition | mk994-tum | prot096-csr | 30AUG2013 | 2:07:02 |
| anksapur | gr0investigator0by0site | mk994-tum | prot096-csr | 30AUG2013 | 2:10:35 |
| anksapur | gr0baseline0characteristics | mk994-tum | prot096-csr | 30AUG2013 | 2:11:27 |
| anksapur | gr0baseline0characteristics | mk994-tum | prot096-csr | 30AUG2013 | 2:11:45 |
| anksapur | gr0medical0history | mk994-tum | prot096-csr | 30AUG2013 | 2:13:10 |
| anksapur | gr0prior0therapy0vaccine | mk994-tum | prot096-csr | 30AUG2013 | 2:14:03 |

Display 1, Macro Usage Output Dataset

USE OF THE TRACKING MACRO

When determining which of our standard macros should be tracked, we selected those most project teams are expected to use, larger macros that are likely to result in substantial time savings for project teams, and macros that may no longer be needed. A call to the %update0usage0log macro has been installed in many of our standard macros, and its use is transparent to the end users.

Occasionally our macros may be run in production in a strictly PC environment or by CROs or other partners. There is also a chance that the programs may be executed by a regulatory agency. In these situations access to the shared server containing the tracking dataset is not available, and the users may not have a copy of the %update0usage0log macro. To avoid errors should these circumstances be encountered, each call to the macro includes a check to verify our SAS shared server is accessible to the user. If the shared server is not found, the tracking macro is not executed and use of the standard macro is not tracked. The SAS code added to our standard macros to call the tracking macro is shown in Display 2.

```
%*** Call compliance tracking macro if working in our environment ***;
%if %qsysfunc(libname(shared /opt/bards/BARDSPRO/bardssys/share,
                     remote, server=bards._99999)) = 0 %then %do;
  %update0usage0log(macroname = clda);
%end;
```

Display 2, Sample Call to the Tracking Macro

MACRO USAGE SUMMARY LISTINGS

There are currently four macro usage summary listing programs defined and planned to be released after more usage data is collected. These listings may be generated for a particular month, quarter or year as the time period is flexible and parameter driven. The number of listings and graphs and their complexity will most likely expand in future releases as we analyze the data over time. Output 1 shows a simple overall count of the number of times each macro was called over one month. From this output we can quickly spot the macros that were used most frequently, and those that were not executed during the month.

| Macro Usage Summary for Month JUL2013 | |
|---------------------------------------|-------|
| Macro Name | Count |
| gr0baseline0characteristics | 30 |
| kmdata | 15 |
| gbars | 7 |
| convert0sdtmplus0data | 5 |
| implement0br91 | 5 |
| gkm | 3 |
| repeatd | 0 |

Output 1, Overall Macro Usage Listing

Output 2 is a sample summary listing showing the overall count of each macro for each quarter. This listing assists in determining how often each standard was used over time.

| Macro Usage Summary by Quarter | | |
|--------------------------------|---------|-------|
| Macro Name | Quarter | Count |
| gbars | Q12012 | 3 |
| | Q22012 | 1 |
| | Q32012 | 6 |
| | Q42012 | 8 |
| | Q12013 | 10 |
| | Q22013 | 15 |
| gr0baseline0characteristics | Q12012 | 20 |
| | Q22012 | 20 |
| | Q32012 | 31 |
| | Q42012 | 35 |
| | Q12013 | 42 |
| | Q22013 | 43 |

Output 2, Listing of Macro Usage Over Time

Output 3 is a sample summary listing which shows the macros used by each deliverable, and the number of times each macro was executed for each deliverable over a span of time. This listing may also be generated summarized to the user id and macro name level, replacing project and protocol by user id in the listing. This report is helpful in identifying specific project teams or individuals who are not finding the standard macros useful, or may be using non-standard program code instead of the standards to produce their deliverables. Once identified the standards team may then reach out to these users and provide support or training.

| Macro Usage Summary by Compound/Protocol 3 rd and 4 th Quarters of 2013 | | | |
|--|-------------|-----------------------------|-------|
| Project | Protocol | Macro Name | Count |
| V999-9vol | 009 | gbars | 1 |
| | | kmdata | 1 |
| | | --- | --- |
| | | | 2 |
| Mk999-tol | prot099-csr | convert0sdtmpplus0data | 1 |
| | | implement0br91 | 1 |
| | | gr0baseline0characteristics | 12 |
| | | constrained0lda | 3 |
| | | --- | --- |
| | | | 17 |

Output 3, Listing of Standards Use by Project Team

Output 4 is another sample summary listing showing each standard macro being tracked and the count of project/protocols for which the macro was executed at least once over a specified time period. This information assists in identifying macros that are being used by only a few project teams or therapeutic areas. We then investigate whether the macro needs to be more widely publicized, or if the functionality of the macro should be expanded to cover the needs of more project teams or therapeutic areas.

| Percentage of 12 Deliverables Using Each Macro 3 rd and 4 th Quarters of 2013 | | |
|--|-------------------|------------------------|
| Macro Name | Deliverable Count | Deliverable Percentage |
| gr0baseline0characteristics | 12 | 100.0% |
| kmdata | 9 | 75.0% |
| gbars | 6 | 50.0% |
| convert0sdtmpplus0data | 6 | 50.0% |
| implement0br91 | 6 | 50.0% |
| gkm | 4 | 33.3% |
| repeatd | 0 | 0.0% |

Output 4, Listing of Standards Use Over All Deliverables

COMPLIANCE PROCESS

The main compliance macro is executed in the development area after most of the programming is completed, and reports specific areas of non-compliance. Any deviations are reviewed and addressed by the project team. The compliance macro also generates data that is used by the compliance team to assess and track compliance. Compliance reporting macros generate consolidated project summary reports across studies that are sent to upper management. Macros that report on standards use are periodically run and the output is assessed by the standards group and management to monitor usage and prioritize updates and enhancements if needed.

TOOLS FOR REPORTING ON SOP COMPLIANCE

There are currently six macros that are part of the compliance assessment process. The main macro (%sop0compliance) determines the programs used for a single deliverable and the level of validation required for

each program, and verifies that the expected validation documentation exists. The output datasets generated by the main macro are read into two additional macros which generate tables and graphs displaying the level of compliance found across protocols and time points. Three additional utility macros read in standard excel spreadsheets containing the validation information and convert the contents to SAS datasets. A program flow is shown in Figure 2.

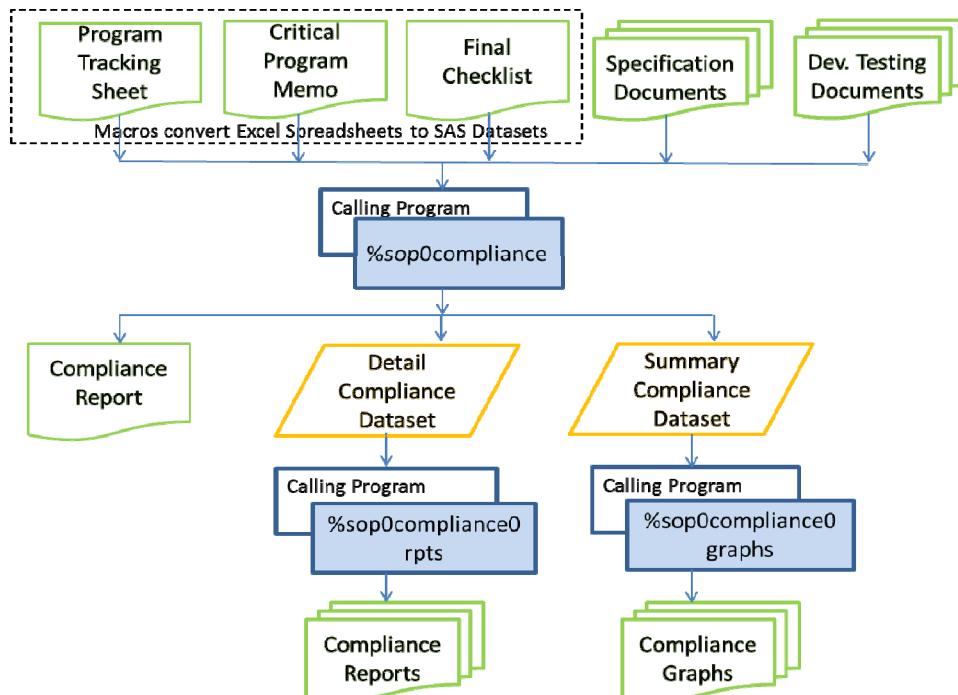


Figure 2, SOP Compliance Flow Chart

INPUT

The input into the compliance macros includes three excel files. The names of the SAS datasets containing the data in the excel sheets are passed into the macro via SAS parameters. Each of these documents is described below.

- **Program Tracking Sheet** – Contains one row for each program that creates an analysis dataset or generates a table, listing or figure for the protocol. All critical programs/macros are designated. The names of the programmers, validators and approvers of each program are documented. Completion target dates for each step are tracked and when a step is complete the status is updated.

| ARGrid001100100.xls [Compatibility Mode] - Microsoft Excel | | | | | | | | | | | | | |
|--|------------------|-------------------|----------------------------|-------------|--------|-------------------|-------------|----------|--------------------|-------------|--------|-----------------------|----------|
| Program Category | | | | | | | | | | | | | |
| Activity / Task / Program | Program Category | Critical Program? | Requirements Documentation | | | Developer Testing | | | Double Programming | | | Statistician Approval | |
| | | | Who | Target Date | Status | Who | Target Date | Status | Who | Target Date | Status | Who | Status |
| 3 Program Name | | | | | | | | | | | | | |
| 4 ecgsum | Graphs | | John Doe | 11/30/2012 | C | John Doe | 12/20/2012 | C | | | | Bill Qie | approved |
| 5 csla | Data | Y | John Doe | 10/20/2012 | C | John Doe | 11/30/2012 | C | Jane Smith | 12/30/2012 | C | Bill Qie | approved |
| 6 adsl | Data | Y | John Doe | 10/15/2012 | C | John Doe | 11/15/2012 | C | Jane Smith | 12/15/2012 | C | Bill Qie | approved |
| 7 formats | Utility | | | | | John Doe | 10/08/2012 | C | John Doe | 11/08/2012 | C | Bill Qie | approved |
| 8 pr0specific0ae | Table/Listing | | | | | John Doe | 11/30/2012 | John Doe | 12/30/2012 | | | Bill Qie | approved |
| 9 p0anova sas | Macro/Analysis | Y | John Doe | 11/15/2012 | C | John Doe | 12/15/2012 | C | Jane Smith | 1/15/2013 | | Bill Qie | |

Display 3, Sample Program Tracking Sheet

- **Critical Program Memo** – The programs used to generate tables, listings and figures to support the primary and secondary analysis for a study are recorded in the critical program memo spreadsheet. This is the official document for defining the programs that require double programming.

- **Final Checklist** – This spreadsheet is used to further enhance compliance and contains a checklist of items that should be completed to follow the SOP and close out the study. The final checklist must be reviewed by both the statistical project lead and the programming manager.

| BARDS AandR Package Final Checklist | | | |
|-------------------------------------|---|-------------------------------------|-------------------------------------|
| | A | B | C |
| 5 | Project/Indication: | mk999 | |
| 6 | Protocol(s)/Package Name: | 042-csr-week6 | |
| 7 | Statistician: | Bill Qie | |
| 8 | Statistical Project Lead: | Joyce Shore | |
| 9 | Date of Review with Statistical Project Lead: | 14-Nov-12 | |
| 10 | Programmer: | Don Row | |
| 11 | Programming Manager: | Rich Chen | |
| 12 | Date of Review with Programming Manager: | 14-Nov-12 | |
| 13 | | | |
| 14 | Check | Statistician | Programmer |
| 15 | Critical program memo is accurate, complete and clear. | <input checked="" type="checkbox"/> | NA |
| 16 | All output, programs, requirements, and validation documentation are complete and stored per the Define, Develop, and Validate SOP and standard | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |

Display 4, Sample Final Checklist

- **Specification Documents** – Individual specification documents are required for programs that have a program category of “Macro/Analysis” in the program tracking spreadsheet. These specification documents must be named in a standard way (programname-spec.doc) and must exist in a specific folder. The filename associated with the folder is passed into the macro in a macro parameter.
- **Developer Testing Documents** - Individual developer testing documents are required for each program listed in the program tracking sheet. These documents must also be named in a standard way (programname-devtest.doc) and must be stored in a specific folder. The filename of this folder is passed into the macro via a macro parameter.

MACROS

The macros developed to support SOP compliance assessment are briefly described in this section.

- **%sop0compliance** – This macro reads the program tracking sheet data to identify the programs included in the protocol and to determine most of the specification and validation requirements for each program. The need for double programming depends on the criticality of the program, which the macro obtains from the critical program memo. Once the specification and validation requirements are identified the macro searches for the developer testing checklist documents for each program along with the specification documents where needed. The program tracking sheet data is used to determine if the remainder of the validation was completed as expected and the final checklist data is read to determine if required reviews were completed. An HTML file is generated as shown below, as well as two SAS datasets, a detail dataset and a summary dataset, which are used as input to the summary report listings. .
- **%sop0compliance0rpts** and **%sop0compliance0graphs** – These macros produce the output summary reports and graphs detailed below.
- **%read0program0tracking**, **%read0critical0pgm0memo** and **%read0final0checklist** – Each of these macros reads an input spreadsheet and converts its contents to a SAS dataset.

OUTPUT

An HTML report is the main output from the %sop0compliance macro. A sample is shown in Output 5. This report lists each program for the protocol and clearly shows whether each piece of required documentation exists. The percent of compliance reached for each type of program review is also displayed, along with the compliance measure

for package review by the programming manager and the statistical project lead. This listing helps the programmer identify any gaps that exist between actual and required documentation.

| Program Name | Program Category | Critical Program Flag | Requirements Criteria Met | Developer Testing Criteria Met | Double Programming Criteria Met | Statistician Approval Criteria Met |
|---|------------------|-----------------------|---------------------------|--------------------------------|---------------------------------|------------------------------------|
| adsl | Data | | _missing_ | _missing_ | | _approved_ |
| csla | Data | Y | _missing_ | _missing_ | _completed_ | _approved_ |
| ecgsum | Graphs | | | _missing_ | | _approved_ |
| gline0ecg | Graphs | | | _missing_ | | _missing_ |
| gr0ich0ae0listing | Table/Listing | | mock-up TLFs document | gr0ich0ae0listing-devtest.doc | | _missing_ |
| makedemo | Data | | _missing_ | makedemo-devtest.doc | | _approved_ |
| pr0ns0ae | Macro/Analysis | | _missing_ | _missing_ | | _missing_ |
| pr0ns0graf01 | Graphs | | | _missing_ | | _approved_ |
| pr0so0drug0compliance | Macro/Analysis | Y | _completed_ | _missing_ | _missing_ | _approved_ |
| pr0specific0ae | Table/Listing | | mock-up TLFs document | _missing_ | | _approved_ |
| Compliance Measure | | | 42.9% | 20.0% | 50.0% | 70.0% |
| Date of Review with Programming Manager: | _missing_ | | | | | |
| Date of Review with Statistical Project Lead: | 06/04/13 | | | | | |
| Compliance Measure | 50.0% | | | | | |

Output 5, SOP Compliance Report

Output 6 shows a sample summary report which displays the rate of compliance for each type of testing for each protocol over one quarter. Overall compliance is also displayed.

BARDS Compliance (Define, Develop and Validate Programs)

1Q2014

| | Overall B-S004 Compliance | Requirements | Developer Testing | Double Programming/Independent Validation | Stat Approval |
|-----------------------------|---------------------------|--------------|-------------------|---|---------------|
| Departmental Average | 75% | 56% | 99% | 98% | 44% |
| MK-xxxx-999 | 100% | 100% | 100% | 100% | 100% |
| MK-xxxx-999 | 61% | 38% | 98% | 95% | 13% |
| MK-xxxx-999 | 63% | 30% | 100% | 100% | 20% |

Requirements - % of programs having expected requirement documents

Developer Testing - % of programs with developer testing checklists

Double Programming/Independent Validation - % of programs where required validation was completed

Stat Approval - % of program requiring Statistical approval which were approved

Overall B-S004 Compliance – average of 4 items above.

Output 6, SOP Compliance Summary Report

Output 7 shows a similar sample summary report displaying the level of compliance for the two required package reviews for each protocol over one quarter. The departmental average is also displayed. Outputs 6 and 7 are used by management to determine which programming groups are non-compliant and enables them to remediate where needed.

BARDs Compliance (BARDs Operations)

1Q2014

| | Stat Project Lead Review | Programming Manager Review |
|-----------------------------|--------------------------|----------------------------|
| Departmental Average | 67% | 100% |
| MK-xxxx-999 | Not Done | Done |
| MK-xxxx-999 | Done | Done |
| MK-xxxx-999 | Done | Done |

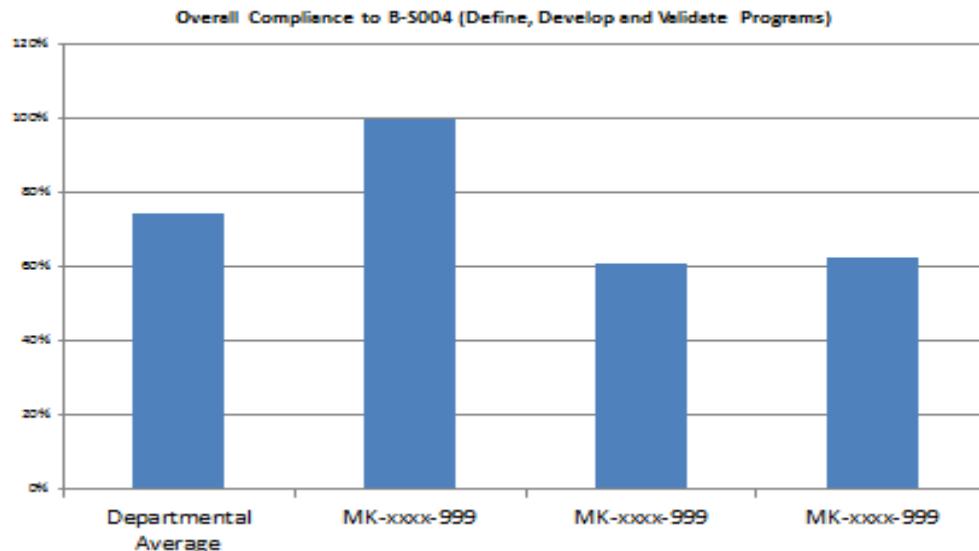
Stat Project Lead Review – Done or Not Done populated to indicate if stat section of the BARDs A&R final checklist was completed

Programming Manager Review – Done or Not Done populated to indicate if programming section of BARDs A&R final checklist was completed

Departmental Average – Done = 100% and Not Done = 0% - average of all trials in the table

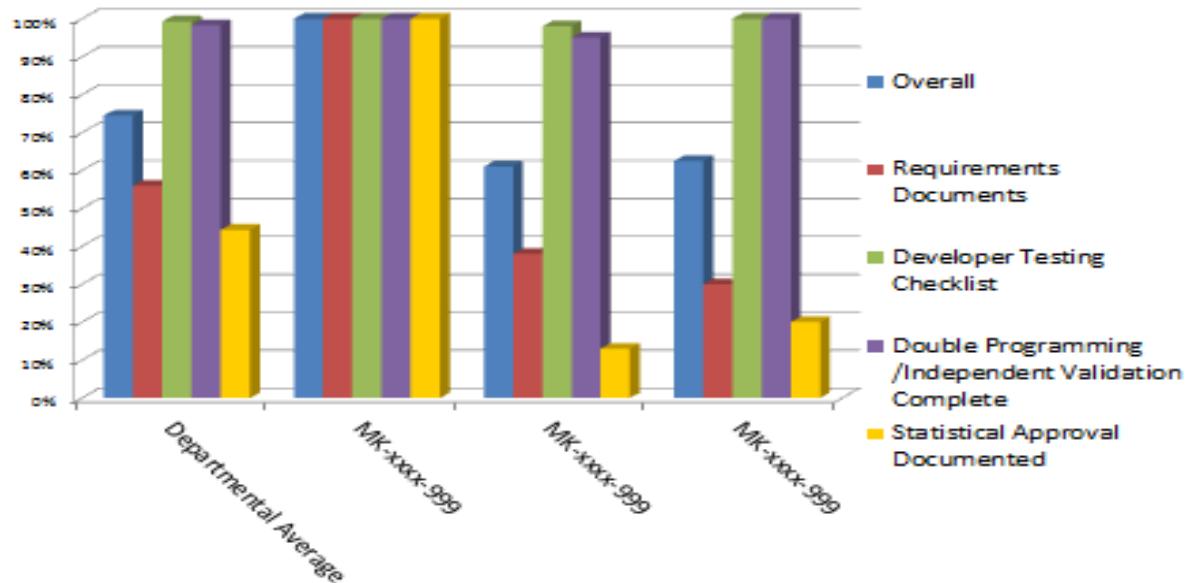
Output 7, Final Review Checks

Two graphs are produced, one to show overall compliance for each protocol (Output 8) and another to show the level of compliance for each type of validation, for each protocol (Output 9). These figures are also used by management to assess departmental compliance and address as needed.



Output 8, SOP Compliance Summary Graph by Trial

Compliance to B-S004 (Define, Develop and Validate Programs) Specific Measures



Output 9, SOP Compliance Summary Graph by Trial and Validation Type

CONCLUSION

SOP compliance is extremely important in the pharmaceutical industry and must be adhered to when programming and validating the programs that generate deliverables to be included in regulatory submissions. Often these deliverables must be completed under tight timelines which can result in overlooking required steps in the process. Applying automated checks to key steps in the process and generating reports showing areas of non-compliance can be valuable to programmers and statisticians, and assures management that the process is being followed as expected. Ensuring standard macros are used by all project teams is important for reducing the risk of error, decreasing programming and validation time and increasing SOP compliance. Developing tools to track and report on standards use across a programming department can be helpful in assuring standards are being widely used. The compliance assessment effort also minimizes the probability of any findings that could adversely affect a trial in the event of an audit or a submission.

ACKNOWLEDGMENTS

The authors would like to thank their management and colleagues who reviewed and provided feedback on this paper.

CONTACT INFORMATION

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

Name: Ginger Redner
 Enterprise: Merck & Co., Inc.
 Address: 351 Sumneytown Pike
 City, State ZIP: North Wales, PA 19454
 Work Phone: 267-305-7634
 Fax: 267-305-6474
 E-mail: virginia_redner@merck.com

Name: Eunice Ndungu
 Enterprise: Merck & Co., Inc
 Address: 351 Sumneytown Pike
 City, State ZIP: North Wales, PA 19454
 Work Phone: 267-305-6866
 Fax: 267-305-6474
 E-mail: Eunice.ndungu@merck.com

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.