Abstract

Despite highly efficient hiring practices a firm may have, it is important to monitor and evaluate the performance of SAS® programmers. Periodic performance evaluation is necessary to minimize the cost associated with human resource turnover, and to ensure good quality and productivity in terms of programming output. Solid programming support is always a vital necessity for efficient clinical reporting process. Whether working in a service provider or large pharmaceutical company environment, performance evaluation can be an effective tool to measure customer satisfaction and to ensure effective utilization of programming resources.

Primary objective of this paper is to recommend a method for in-depth performance evaluation of SAS® programmers working in clinical reporting. Depending on time constraints of management, a SAS® programmer’s performance can be measured by combining elements of quantitative error rate based method and qualitative personalized feedback method. The approach suggested here is based on assessment of core skills such as programming ability, analytical skills and communication skills, as these are the basic ‘required’ skills a clinical SAS programmer must possess. It is highly desirable if such core skills assessment conducted at adequate timings should have a foundation of SAS® programming metrics. The paper concludes with recommendations about effective uses of performance evaluation results and related possible areas of research.

Introduction

When a pharmaceutical firm or a service provider hires a SAS® programmer, interviewers ensure that the new candidate passes a rigorous screening process. In such screenings, usually interviewers thoroughly evaluate programming and analytical abilities of a programmer. Despite this, many times programmers do not exhibit similar degree of skills and confidence levels in the actual job. The differences in their interview and actual job performance can be attributed to many factors ranging from ‘difficult study design’ for treatment allocation of clinical trial, to ‘different working environment’. In such cases, in order to ensure best possible programming productivity, it is essential to periodically monitor and evaluate the performance of a SAS® programmer. Even though performance evaluation for SAS® programmers is conducted in almost all companies, most of the time such evaluation is left up to the discretion of the immediate supervisor of a programmer. Even if such method seems logical, a standardized method of performance evaluation is necessary mainly in service providers or pharmaceutical companies involving large staff.

Following section will emphasize on why it is important to have a standardized performance evaluation method. Further discussion elaborates on methods and timings of performance evaluations and illustration of supplementary metrics information that needs to be
collected for effective evaluation. Concluding section recommends effective uses of performance evaluation results.

Need For Standardized Performance Evaluation

When a supervisor of a SAS® programmer concludes that the programmer performed below expectations, such performance evaluation is highly subjective in manner. Based on performance interviews of a programmer or even based on past programming work done, a supervisor can have different levels of expectations from the programmer. However, some of such expectations can be unrealistic because of following factors:

a) Difficulty level of Programming: Certain clinical reporting work involves tight timelines, complex clinical trial designs, and multiple clinical domains involving lot of programming.

b) Different types of programming: Programming skills required for summary reporting such as Integrated summaries of safety is different from programming skills required for phase-III efficacy reporting for protocol specific tables. Summary reporting usually demands tighter timelines.

c) Communication: If programmer doesn’t establish a good communication dialog with other personnel involved in reporting process, it causes lot of confusions resulting in inaccurate programming specifications or incorrect interpretation of programming specifications by the programmer.

d) Other technical factors: This includes improperly configured operating system, inadequate performance of reporting system and other technical factors.

Despite the above factors, the question remains about why to standardize such performance evaluation? The answer can be found in the logistics of programming resource planning and management. It is important to weigh the performance of all SAS® programmers on the same scale. In that way, management can strategize the training, mentoring requirements, and can take proactive measures in effective resource allocation. Management often needs to utilize skills of SAS® programmers for time sensitive and critical reporting work. Results of standardized performance evaluation can be a more reliable source for selection of appropriate programming resource.

Methods for Performance Evaluation

While developing a performance evaluation instrument, it is important to assess how much time management can afford to spend in such an evaluation process. This section explains two approaches; i) Quantitative or error rate based evaluation, and ii) Personalized feedback. If management has a setup of data collection processes in practice, then getting such evaluation related data may not require substantial time investment on their part.

Quantitative or error rate based evaluation:
Often programming deliverables such as tables, listings, analysis datasets, and graphs require some degree of corrections after the programmer submits them as final. If the cause of such correction has a root in the code written by the programmer, then such correction can be considered as a ‘programming error’. Such programming errors can be very basic like missing
label of a derived variable, or like a format, which is not meeting required specifications. In certain cases, these programming errors can be serious and can impact the timelines of clinical reporting. Regardless of seriousness of such errors, management needs to capture these after every programming deliverable. Such errors are always communicated to the programmer so that the programmer can rectify such errors and resubmit the deliverables. At the same time it is important for management to keep the records of such errors from the performance evaluation point of view. After getting these data, management needs to weigh the total number of errors committed against the total number of outputs produced by programmer. ‘Error rate’ can be a good measure of performance evaluation. Errors and the re-work done by the programmer can result in lost productivity, and can result in increased cost for clinical reporting. For each deliverable, the error rate can be computed as follows:

\[
\text{Error rate} = \frac{\text{Total errors committed by programmer in all outputs of a deliverable}}{\text{total number of outputs produced in a deliverable}}.
\]

Another important factor in understanding the error rate is the required skill set ‘level of programming’. For example, programming skills required for creating summary reports using statistical procedures is usually more than that required for basic listings from the dataset. It is important to keep the level of programming in mind while interpreting the error rate of a programmer.

**Qualitative or personalized feedback method:**
In general, management often develops different types of survey instruments to gather such personalized feedback. While developing such instruments for SAS® programmer’s performance in clinical reporting environment, it is important to focus on core skill sets required to perform programmer’s duties. Following considerations can be helpful while developing performance analysis instruments and while providing rating to the programmers.

**Programming skills:** As discussed earlier, this could be subjective. ‘Good’ programming skills rated by one supervisor could be equivalent to ‘average’ programming skills rated by another supervisor. While analyzing programming skills it is important to consider the degree of supervision required to monitor the programming activities of a programmer and the extent of ‘generic’ programming, which SAS® programmer is capable to do.

**Analytical Abilities:** Many times a SAS® programmer needs to understand the programming specifications, translate them into algorithm and then develop the code. Obviously, analytical abilities are critical for the success of a SAS® programmer. Often the errors committed by a programmer may not be directly related to the code development, but may be attributed to the misinterpretation of specifications. Based on these factors supervisor needs to provide a rating on the analytical abilities of the programmer.

**Communication skills:** In order to provide good quality programming deliverables, a programmer must get required specifications and valid data within appropriate time frame. If the programmer doesn’t get necessary input within required time frame, it seriously impacts the quality and timing of programming deliverables. It is often the responsibility of a programmer to ensure that he or she has necessary and valid input. Programmer needs to communicate with the
concerned personnel to get such valid input. Lack of these skills can be analyzed through basic factors like responsiveness to e-mail and phone communication, ability to raise ‘valid’ questions to appropriate personnel in a timely manner.

Other factors: Other important factors include knowledge of a firm’s processes, and systems that is acquired by the programmer. These factors can be considered as success factors especially for the contracted SAS® programming resource. If programmers know the systems and processes, they are likely to be more productive and efficient. Another important factor is programmer’s knowledge of clinical data. If a programmer ‘knows’ the clinical trial data for a certain compound or drug and posses necessary domain knowledge, then he or she is likely to provide better programming output. Most of the other factors include the behavioral attributes such as hard-work, dedication, customer focus, and good attitude of a programmer.

Effective Use of Evaluation Results

As discussed earlier, the quantitative or error rate based evaluation strictly focuses on numbers like level of programming and programming error rate. However, decisions based on such evaluation may not be ideal because such evaluation lacks ‘opinion based’ or subjective input. Management should combine the elements of both approaches while judging the performance of a SAS® programmer. Qualitative feedback could be a periodic process, and can be taken on a quarterly, bi-annual, or annual basis depending on the policy set by management. Error rate based feedback needs to be captured and archived after every major deliverable submitted by the programmer. Based on need, management can get the error rate based data of each programmer under consideration and then seek out the qualitative feedback from the supervisors of the programmers. Such performance evaluation method is extremely helpful in re-assigning the programmers to different tasks. Consider the following example:

Programmer-A: Assigned to develop analysis datasets and generate summary tables in accordance with specifications.
Programmer-B: Assigned to develop survival analysis related summary tables based on already generated intermediate datasets.
Programmer-C: Assigned to develop graphs and integrated summary tables.

Assignments of all three programmers are about to end and you need to choose which programmer needs to be assigned to the upcoming programming assignment involving development of safety reports for a clinical trial having cross over design for treatment allocation. As a resource planner, you get the following evaluation of the three programmers.

<table>
<thead>
<tr>
<th>Programmer</th>
<th>Type of Work</th>
<th>Error rate*</th>
<th>Level of programming**</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Analysis datasets, and summary tables</td>
<td>0.03</td>
<td>6</td>
</tr>
<tr>
<td>B</td>
<td>Survival analysis tables.</td>
<td>0.03</td>
<td>8</td>
</tr>
<tr>
<td>C</td>
<td>Graphs, and integrated summary tables.</td>
<td>0.096</td>
<td>4</td>
</tr>
</tbody>
</table>

* For example 0.03 or 3% error rate means 3 programming errors committed by programmer while delivering 100 outputs. The outputs can be either tables/listings, graphs, or datasets.
** Level of programming: On the scale of 1 to 10, 1 being lowest or least difficult. This can be decided by the task assigner.

The feedback received from the supervisors of these programmers is as follows:

<table>
<thead>
<tr>
<th>Programmer</th>
<th>Programming skills</th>
<th>Analytical ability</th>
<th>Communication skills</th>
<th>Other factors (integrated)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>4</td>
<td>4.5</td>
<td>3.5</td>
<td>4</td>
</tr>
<tr>
<td>B</td>
<td>4</td>
<td>4.5</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>C</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

* Rating based on the scale of 1 to 5, 1 being lowest or poor rating.

Above information is very valuable, while deciding resource allocation for next assignment. Based on both feedback scenarios, programmer B seems to be more ideal than programmer A. This is evident from better communication skills feedback and based on the fact that programmer A has the same error rate as that of programmer B for the lower difficulty level of programming.

It is difficult to judge which programmer is better between programmer B and programmer C. At this time, it is important to understand the level of programming required for the new assignment. If the new assignment requires level of programming of up to 6, then it is more desirable to assign such task to programmer C. Assigning programmer B for such task would be insufficient usage of a skilled resource. Management may be able to leverage skills of programmer B for some other challenging assignment. Whereas for assignments requiring programming level of more than 6, programmer B may be a better choice. Even though the task assigner can specify the difficulty level of programming, such level can be understood from the various technical details of an assignment. Some of these details include programming specifications, type of output deliverables, nature of data, and method of treatment allocation in the clinical trial.

In addition to performance evaluation results, prior experience of programmers in certain clinical domains, reporting methods, internal data standards of a firm, and knowledge of firm’s reporting system are also other major considerations while deciding programming reassignments.

**Conclusion**

As illustrated above, combining the elements of quantitative error rate based method and personalized feedback evaluation method provides very useful information for resource planning and allocation. Capturing and retrieving such data requires minimal time commitment from management. Resource allocation based on such methods not only helps the reporting team to deliver good quality programming output but it also helps management in leveraging the skills of programmers. Such performance evaluation also uncovers the areas where a programmer may possibly need mentoring or training support. Inappropriate allocation of resources may adversely impact the quality of programming output and at the same time it may not provide necessary job satisfaction to the programmer. Periodic performance evaluation by combining the elements of the above two methods can help management reduce the turnover.
The methods discussed in this paper are more applicable to a large pharmaceutical company environment or in service provider involving high volume of staff. Above discussed methods are based on the metrics, such as level of programming, and error rate of a programmer for each programming deliverable. Collection and analysis of appropriate metrics for SAS® programming in a clinical reporting environment, is always a challenging task. Further research in integrating the programming metrics with performance evaluation could certainly help the pharmaceutical industry.

Acknowledgement
Author would like to acknowledge the guidance and mentoring received from Mr. Antony Goncalves, Director, eClinical Solutions, Eliassen Group.

Reference

Contact Information: Your comments and questions are valued and encouraged. Author can be contacted at pshiralkar@eliassen.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.