PharmaSUG 2012 - Paper MS09

Coaching the Individual SAS Programmer: Generalist vs. Specialist

Mark Matthews, PharmaNet/i3, Indianapolis, IN
Sandra Minjoe, Octagon Research Solutions, Wayne, PA

ABSTRACT:

Good managers recognize that all programmers are not created equally, and we can best motivate and reward employees when we recognize them as individuals. One major distinction is that some programmers are generalists and enjoy the variety that comes with different types of studies and analyses, while others are specialists and prefer to hone their skills in one particular area. Difficulties arise when trying to force a specialist into a generalist role, or a generalist into a specialist role. This paper will show managers how to determine whether a programmer is a generalist or a specialist, help them find a good fit, and recommend some resources to increase skills in each area.

The two authors have chosen different career paths and share their experiences of generalist and specialist, both good and bad. Mark is a generalist, in management at a CRO (Contract Research Organization), where he has worked with many different clients. Sandra is a specialist, giving training and consulting in one niche market. Each knows the joy found in doing what comes natural vs. the pain from being asked to do what does not, and applies this learning to managing others.

DEFINITIONS:

Let us define what the authors mean by specialist and generalist.

A clinical SAS® programmer **Generalist** is one who supports any or all parts of the clinical drug cycle. We may have heard the phrase ‘turn the clinical crank’, and there are many of us in that space with a seemingly endless supply of analyses after analyses. A group of generalists put emphasis on speed, quality, safety of the patient, and bringing drugs to market. The spectrum of work can range from Phase I to Phase IV and include a multitude of activities such as annual reporting, regulatory response, submission activities, publications, etc. Because of that wide spectrum of tasks, a generalist has the opportunity to engage in a wide variety of work. And yet, because of the great span of that drug cycle, generalists can spend many years on a single therapeutic area, submission phase, or efficacy analysis, to the point where one can argue that they have become specialists. That is somewhat true, but from the authors’ definition in this paper, the generalist is proficient in many different areas.

So what is a specialist? A **Specialist** is one who focuses on one or a small number of areas of the clinical drug cycle. Specialties can include types of data (such as safety, PK, or lab), indication (such as oncology, cardiovascular, or HIV), or statistical analyses (such as survival, regression, multivariate analysis, and simulation). A newer specialization is in CDISC (Clinical Data Interchanges Standards Consortium) standards, including SDTM (Study Data Tabulation Model), ADaM (Analysis Data Model), and ODM (Operational Data Model) applications. Specialists are often added to a project short-term, to help only in their area of expertise, and know very little background of the project and studies involved. Some specialists may act as consultants and/or trainers, or review the work of others who are not specialists.

INTRODUCTION:

We believe that one of the core responsibilities of a manager is to engage staff and understand what motivates them, fostering an environment of opportunity and fulfillment. Managers generally have more exposure to the industry, which can enable us provide the guidance and opportunities for career growth or fulfillment for our staff. We constantly keep our eyes out for industry trends and help our staff see where things are going and where their niche could be.

Refer to the paper “Similarities and Differences in SAS Programming among CRO and Pharmaceutical Industries” (Minjoe and Matthews, PharmaSUG 2011). It suggests that a generalist in the CRO industry has the most variety of work; much more so than the pharmaceutical (or sponsor) companies. Thus, depending on the CRO organization’s specialty as an enterprise, a manager has a very nice selection of opportunities to place their staff to find the optimum balance of career and business fit. Large sponsor companies with a variety of drugs, biologics, and/or devices for different medical indications also offer a nice selection of opportunities to both generalists and specialists.

Managers in the larger companies, be it CRO or sponsor, will have the most opportunities to coach staff into generalist or specialist roles. This paper will focus on these managers.

EXAMPLES OF THE GENERALIST AND SPECIALIST:

We'll begin by considering the type of data that both a generalist and a specialist work with. An example is that of creating laboratory listings and tabulations in a clinical trial where the laboratory values come from a single centralized lab. Also, let's say that all of the laboratory values are expected be that of the conventional unit values as opposed to SI (Standard International) unit values. A generalist would be able to easily produce any table or listing based on the provided specifications of output and layout: seemingly a very frequent and reproducible analysis. In a particular listing, let us say that the specifications for normal range for a lipid test of total cholesterol is <200 mg/dl. What would happen if somewhere along the chain of data flow that the total cholesterol happened to be represented in the SI units (where the typical normal range is <5.2 mmol/L)? If the logic in some programming area does not consider units, then there is a potential for the SI readings greater than 5.2 be flagged as normal in conventional units and perhaps every record on the listing would indicate a normal lab test. Unless the generalist is very proficient and experienced in lab analysis, then this incorrect assessment of the normal range could go undetected into the next stage of the workflow: say the validation of the listing. Most people do know what their total cholesterol should be, however they may not know the units associated with it. So in this case, anyone who conducts a plausibility check on the values may look deeper into the specifications and discover the root of the problem. However, what about all the other commonly reported laboratory values such as amylase, hemoglobin, creatine kinase and bilirubin? A generalist with little experience in labs may not know all the medical terminology; however, they do know what person or resource to find out. A generalist who is very proficient in labs or a specialist in labs would be a good resource for this consultation.

Next let's consider the statistics used for analyzing mean change from baseline to endpoint efficacy analysis, where a programmer can begin to build proficiency in applied statistics even without an advanced degree in statistics. In a real life example, two generalists (a programmer and validator) programmed a typical mean change from baseline analysis with a linear model yielding a p-value. The biostatistician then asked them to center the baseline of the independent variable (that is, take every value and subtract it from the mean). That was an easy SAS formula adjustment for any level programmer. However, as these programmers looked at the resulting p-value, they observed it did not change – it was identical to the p-value prior to centering the baseline. Those generalists who have not experienced such a scenario would then need to research, or consult with a specialist or statistician as to why this is happening. A specialist in the space of inferential analysis would immediately recognize that a linear transformation using type III sum of squares on a linear model would not change the resulting p-value; only the model assumptions would change. Unless the SAS programmer has an advanced degree in statistics, they may not fully know the mathematics behind why a linear transformation on a linear model will not affect the p-value in that case, but can be specialized enough in applied statistics to know what can be expected and know when to pursue and inquire about such perplexities. This is another example where you can spend many years in inferential analysis and develop your expertise, or specialty in applied statistics across a wide variety of analyses.

Finally, let's consider an example of developing CDISC ADaM datasets. ADaM is becoming more prevalent in the industry, and many programmers have learned about the ADaM structures. There are a variety of published documents, including the Implementation Guide and some appendices, with a lot of information and examples. CDISC and other companies offer training to provide additional information. A generalist with some experience or training on ADaM can usually develop ADaM-compliant data. Being able to follow these rules is all that is needed for the generalist. However, sometimes questions arise that aren't well covered in the ADaM documents. One interesting example is when a result captured as a character needed to be mapped to a category used in numeric analysis. The generalist realized that AVALC, the character field, was needed for one analysis, so putting the numeric category into AVAL would fail the required 1-to-1 relationship between AVAL and AVALC. Because a categorical variable such as AVALCAT1 was a character, it could not be used for the needed numeric analysis. He brought this case to the company's internal ADaM consultant, who was a member of the ADaM team and had more experience in this area. The consultant reviewed the options put forth and agreed that neither AVAL nor AVALCAT1 were appropriate in this situation. She knew that the ADaM model allows for adding variables that are based solely on AVAL and don't invalidate the parameter description, and this situation met that criteria. An ADaM variable naming rule is to add an "N" to the end of a variable name to create the numeric version, shortening the variable name if needed, so she advised the generalist to create a numeric categorical variable named AVLCAT1N.

We shared here three examples, each demonstrating that there are various degrees of specialty that a generalist can experience; be it various types of data, depth of analytical methods, or industry standards. These three examples also show the relationship between generalists and specialists. Generalists are involved in the day-to-day workings of the project and specialists are pulled in as needed to help with specific issues. The reader can probably think of other examples of tasks and types of situations that a generalist and a specialist can experience.

RECOGNIZING THE GENERALIST AND SPECIALIST:

When we experience something new, we generally need to follow a set of rules. Think of the analogy of putting together a ping pong table for the first time. You want to do it right the first time so that you do not get all the way to the final step #56 and realize that you need to take everything apart and go back to step #5 and replace bolt A(2) with Hex bolt H(4). Doing it the first time may take quite some time to read the instructions, verify the instructions and fully understand what is to be done before you do it. The saying "measure twice and cut once, or else you will measure once and cut twice" simply means we have to take the time to do it right the first time. Initially, going through all the steps very slowly may be frustrating. However, after repetitive assembly, we become much more efficient to the point where it may take us only a fraction of the time to complete the assembly. And if our friend wants us to help out in the assembly of his ping pong table, we can either do it for him or, even better, coach him how to do it. We may even provide feedback to the manufacturer, suggesting some assembly improvements that could be made.

Programming in our field can follow the same analogy. There is always a point in time where we are in a learning mode, a proficiency mode, and then a coaching/mentoring mode.

An experienced manager has most likely personally experienced all three situations, and it is this experience that we capitalize on to guide and mentor our staff. It is important as a manager to understand what motivates and fulfills our staff by the type of work they engage in. We need to listen, understand the various situations, and determine how that impacts the individual's fulfillment. Does the programmer get frustrated or lose patience easily when they need to slow down during a learning curve? Do they become complacent when they are in the position of performing the same tasks, even some very complex, so efficiently that there is a mundane, no-thrill feel about it? Some individuals want to be in the state of efficient programming performance, some want to be in that constant learning curve, and, of course, others want a balance of both. Some experienced programmers are motivated to act as a consultant, providing guidance to others, as an industry expert in their field. The answers to these questions often depend on a combination of experience, stage of the career, and personal preferences.

A good way to find out the motivation of each team member is to engage them in a one-on-one discussion. To understand where each person is in along their career path and what makes them tick, we will have different dialogue with a 20-year seasoned SAS programmer when compared to a rookie that is 6 months out of college. We caution against assuming anything from your 20-year veteran who may, in fact, not want to be in the efficient mode because there are some new things that they want to experience. And that 6-month experienced college graduate may already have a good idea of what they want to achieve in their short term and long term career. Always listen and respond to their career questions and objectives.

When we feel that our direct report doesn't seem to know what they want, then sharing personal examples of our own career experiences helps to facilitate the discussion and trigger ideas. As a manager, we may not have all the experiences to share, but we can also share a colleague's or a mentor's experiences, keeping their stories true but their identities secret. This also works well when we see that what a programmer says that they want is in direct conflict with how they perform. Sometimes a programmer believes that they should become a generalist because they're fast and can do a variety of tasks well, but they actually most enjoy when they are given a more specialized task. Others may think that they should be a specialist, but we've seen that they have a tendency to get bored when focused on one particular task. A good manager asks probing questions, such as "What task was the most fun for you this week?", or "If you could take one thing off your plate and never do it again, what would that be?" to help us determine whether the programmer is a generalist or specialist at heart.

We may tend to think of generalists as those with little experience who are learning the industry and specialists as those who have been around awhile, but this isn't always the case. A project lead who is coordinating many programmers is usually a generalist with a lot of industry experience, and a new programmer fresh out of school with courses in survival analysis could be the specialist in that area. As managers, we need to keep in mind that the number of years of experience in the workforce does not always correlate to the area of generalist vs. specialist.

FINDING A FIT:

The employee needs must align with our company needs. Sometimes we have a programmer who isn't the right fit in their current role; for example, a programmer who has been put into a specialist role but would prefer to do a larger variety of tasks, or a programmer who has been given generalist duties but has become an expert in one area. Instead of trying to "force the square peg into the round hole", it's often more practical to move the employee into a better fitting role, one that allows the employee to excel and the company to benefit from their added productivity.

In a large company we usually have the flexibility to move an employee into a different role that is a better fit. It's worth searching for this fit, even if in a different department, because an employee who isn't performing well or isn't happy with the work they're doing is likely to be tempted to leave the company and go elsewhere. The costs of moving programmers within a company are minimal compared to the cost of replacing them. A high turnover rate not only makes it difficult to meet our timelines, it can also make it difficult to hire new employees who are fearful of working for a company that so many others before them didn't seem to like.

Unfortunately, at a small company, or at any company that doesn't have the type of position our employee is best suited for, we might need to coach this employee to find a better fit elsewhere. For example, if the employee excels at PK analysis and wants to specialize in this area, but the company does very little of this work, it won't be possible to move them into a full-time PK analyst role. Sure, the employee can continue in the more generalist role, but probably won't be as productive as others would be in that role, nor as happy. We need to have an honest discussion with the employee about the company needs vs. the employee's desires, and coach them to consider other alternatives.

DEVELOPING SKILLS:

So now we have determined the fit of each of our programmers, and moved them into the appropriate role where they can excel, but we're not yet done. It is important to help our employees continue to grow. An excellent method of increasing their skills is hands-on (on-the-job-training). There are a lot of options each programmer can use to learn new skills or increase the proficiency of current skills.

An effective method of systematically increasing their skills is demonstrated with a development plan. A good development plan will have clear and measureable objectives for both short term and long term goals. Granularity of actions to take toward development includes the elements of "what", "how many" and "by when". In other words, the objective needs to be clear and measureable with a reasonable timeframe around it. Our 6-month entry-level associate who wants to get promoted within the next year may need more independent coding without coaching, and one way to get there is to give them opportunities to experience a variety of SAS procedures. Thus, a clear objective could be in the form of "use at least one new SAS PROC or OPTION in a SAS PROC, per month, for the next twelve months". Our 20-year veteran may want more exposure industry CDISC standards so they can contribute and influence the general scientific community. A goal example may be "get acquainted with two volunteer representatives of the CDISC organization and learn at least one new pioneering effort throughout the year".

Training and conferences can be great skill-building tools; however attendance alone doesn't meet the criteria of a good objective. Be sure to include specifics of what can be gained from the training or conference, how it will be applied, and the benefits that it will provide. For example, the object might be "attend the PharmaSUG conference, find at least one technique or tool that can be incorporated into project X, and present to the team the applied result".

One important factor of a good development plan is that it can be used to show how these increased capabilities have made an impact on the larger business segment objectives. The 6-month entry-level associate who is able to produce deliveries without coaching is not only more productive but also frees up the coach to either work with a new entry-level person or spend more time programming. The 20-year veteran who has learned of a new pioneering effort in CDISC has brought that information to the company and allowed management to plan for future needs. The conference attendee who created a simple, well-documented macro that is now used by the whole team to find date variables that contain partial dates and may need imputation has saved hours of manual review plus may have even saved the biostatistician from creating specs for dates that don't need imputation. In all of these examples, showing the programmer how their work has impacted the company to do business more efficiently will likely increase their job satisfaction, since we all like to hear that we're useful.

THE GENERALIST ↔ SPECIALIST SPECTRUM:

Some SAS programmers seem happy to fill either of these roles, generalist or specialist, or prefer to move back and forth between them. For example, the 20-year veteran may have started as a generalist, become an oncology specialist focusing in time-to-event analysis, gone back to the generalist role after getting bored with that, and moved into the specialist role of developing define documents from specifications. We can think of the generalist and specialist as two ends of a spectrum, with a lot of overlapping areas between.

Often our roles come with some generalist and some specialist components. One of the authors, Sandra, is a specialist when it comes to ADaM, but a generalist when it comes to applying that specialty to any clinical trial. Mark is also a generalist in terms of the types of studies he works on, but a specialist in process metrics, coaching others, and building teams. Thus we can act as both a generalist and a specialist at the same time.

CONCLUSION:

There are SAS programmers who excel as generalists, who enjoy getting to do a lot of different things and challenge themselves with speed and efficiency. Others are specialists, who enjoy continually learning about and contributing in one focused area. There is a need for each type of programmer in the industry, and the trick is to find the best fit for each person.

Below is a chart summarizing the information presented in this document, to help the manager recognize the generalist and specialist, plus give suggestions for how to help these employees with their career development.

	Generalist	Specialist
Happiest When	Producing a lot of deliverables quickly Getting a variety of tasks to do	Solving complex problems in one area Giving recommendations and professional opinions

	Generalist	Specialist
Experience and Competency	Little needed Experienced programmers often lead work of others SAS Certification is recommended	Specialized experience or education/training is required Some way to demonstrate competency (e.g., degree, years of experience) in area of focus
Overall Project Knowledge	Required	Not needed
Tasks to Assign	Tasks across many areas	Tasks focused to a specific area
Training to Provide	As needed in order to gain new skills	Ongoing to keep current in area of expertise
Transferable Skills	Many: can be quickly productive on a new project in any area	Few: will be quickly productive only in areas related to specialty

As managers, we help our programmers to facilitate their own career track by providing them opportunities to be successful. It's the employee's career, but with our experience in the industry and the bigger picture of work that we see, we have tools to help them.

REFERENCES:

Minjoe, S. and Matthews, M. "Similarities and Differences in SAS Programming Among CRO and Pharmaceutical Industries" http://www.pharmasug.org/proceedings/2011/IB/PharmaSUG-2011-IB05.pdf (28March2012).

Information about the Clinical Data Interchange Standards Consortium (CDISC), including the models described here, can be found at http://www.cdisc.org/.

CONTACT INFORMATION:

Your comments and questions are valued and appreciated. Authors can be reached at:

Mark Matthews
PharmaNet/i3
4785 Haven Point Blvd
Indianapolis, IN 46280 U.S.A.
mmatthews@pharmanet-i3.com

Sandra Minjoe
Octagon Research Solutions
585 East Swedesford Rd
Wayne, PA 19087 U.S.A.
sminjoe@octagonresearch.com