

## The Challenges and Opportunities for SAS Statistical Programmers in Two Commonly Used CRO Resourcing Models

Mark Matthews, inVentiv Health Clinical, Indianapolis, IN  
R. Mouly Satyavarapu, inVentiv Health Clinical, Ann Arbor, MI

### ABSTRACT

Managing one's own career and aspirations as SAS programmers in the clinical trial space is always of interest. The Contract Research Organization (CRO) industry provides a diverse opportunity for SAS programmers to pursue and experience their daily work satisfaction and career. Two commonly used models in the CRO space are traditional "Deliverable Based Model" and "Full-Time Equivalent (FTE) Time and Material Model" (also called as role based model). Previously, a presentation at PharmaSUG 2012 identified and differentiated various parameters and methods followed by each model (See PharmaSUG 2012 –Satyavarapu<sup>1</sup>). This paper will take some of those parameters and share actual experiences on how it can provide influence on the career direction of a CRO SAS statistical programmer. The discussion will extend into 3 main categories: SAS and technical skills development, leadership and interpersonal skills development, and the hiring and working conditions. This can give the SAS programmer a better idea on what to expect in these two working environments. Both environments offer challenges and opportunities that can be considered when choosing the best fit for your interests and career.

### OVERVIEW OF THE TWO RESOURCING MODELS

The use of CROs that a sponsor company implements as part of their responsibility has become prevalent in the pharmaceutical industry today. The two models described above that are widely used bring forth some similarities and quite a bit of difference. A straightforward answer that describes the differences and similarities between the two can be as follows: In a deliverable based model, the CRO does clinical research *for* the sponsor company. In a role based model, the CRO does clinical research *with* the sponsor company. However, in any of the two models, the clinical research that ultimately gets done has the common goal of bringing medicines to market. That is, one does the same work in either model.

### SIMILARITIES BETWEEN BOTH MODELS

Let us summarize a few things that remain constant between both models. The phase of study and therapeutic area may not be differentiated between models. Although some sponsor companies specialize and focus on certain therapeutic areas, even they are open to working with both CRO models.

SAS programming and the analysis remains constant as well. The SAS System is the preferred tool of choice for most businesses in clinical trial research. A programmer will also perform the same data derivations, statistical analysis and summaries.

The payment to the CRO is also a constant. The CRO performs the work for a fee for their service. This is part of the symbiotic relationship between the CRO and sponsor company.

These similarities, although few and obvious, may not be enough to play a factor in a SAS programmer's preference to work in any one of the two models. However, it is the differences that can reveal factors to consider when choosing a model that is a best fit for you.

### DIFFERENCES BETWEEN BOTH MODELS

Let us summarize the things that do not remain constant between both models. The sponsor company: In a role based model, the programmer is usually dedicated to that sponsor company, either by contractual arrangements or preference of the sponsor and CRO. In a deliverable based model, although there can be teams dedicated to the sponsor, there is generally the creation of teams that work well together and deliver the same work to multiple sponsors, often simultaneously.

Next, in both models and as said above, even though the sponsor eventually pays the CRO for service, the financial models will be very different. The role based model usually bills by the month or hour, whereas the deliverable based model gets paid by the job – or deliverable, when successfully completed. In the CRO industry, the SAS

programmers are the revenue and thus, there is added pressure to remain utilized. There are different aspects to consider around the overall budget between both models.

As stated above, although SAS programming remains constant, the platforms and various ways companies implement the SAS system is by far, not the same. Generally in a deliverable based model, the CRO uses their own system which allows for the programmer to more freely develop procedures and macros. However, in the role based model, the CRO programmer is usually limited to working in the sponsor's system and abides by their SOPs.

Finally, sometimes the sponsor in the role based model prefers for the CRO programmer to be on-site, or near the office in addition to being an active user in their system. Being on-site and in the sponsor's system enables the CRO programmer to be an integral part of the sponsor's culture. Deliverable based models today are leveraging more telecommuters since the sponsors may not have any location requirements in this model.

## **CHARACTERISTICS OF BOTH MODELS: THE IMPACT ON A SAS PROGRAMMER**

No CRO SAS programmer is alike; we all have our own styles, preferences, personality types and endeavors. However, we do have common skills that we must develop in order to remain competitive and happy with our careers. There are also some CRO programmers who do not wish to continuously enhance their skills for a competitive advantage. They would rather do the work and be fulfilled by the sense of achievement. One thing that is perhaps the same for most of us is that we all want to have daily job satisfaction. Many SAS programmers at a CRO are under one of the two models described above. It is here that we discuss further how each CRO resourcing model can provide opportunities and challenges in the three categories of a SAS programmer: SAS/Technical skills, Leadership/Interpersonal skills, and Hiring/Working conditions.

## **SAS/TECHNICAL SKILLS**

**Various SAS platforms/implementation:** One of the similarities between the two CRO resourcing models is that the SAS programming remains a constant; no matter what environment you are in, you will still be using SAS procedures and code to deliver all the things that a SAS programmer is intended to deliver in the clinical trial setting. However, even though the SAS system is a constant, there seems to be different methods of implementing SAS. Fortunately, the SAS Institute does make the system scalable to work in different environments such as UNIX, PC, Windows server, or a mainframe setting. Nonetheless, different companies implement things differently. Some organizations may develop an internal GUI (graphical user interface) to drag-n-drop or point-n-click their way to creating standard tables or data integrations. Some organizations house a central location of reusable macros that are called for any standard code for derivation or other repetitively used processes. Regardless of reusability and GUI, the SAS programmer needs to know the underlying SAS commands and processing logic in order to provide the correct and complete result of your SAS code. The reality is that different companies implement SAS differently. In a deliverable based model where you are operating in your own company's (standard) environment, it is good practice to have a standard of using SAS. Likewise if you are in a role based model, when you are integrated into the client's system, there too should be a standard that exists within that organization. The role based model perhaps provides more variety of SAS installs and implementation. Here, the programmer can in fact operate entirely in the sponsor's system, and also within their own CRO system when it makes sense; role based model can implement a hybrid of using the sponsor and the CRO system. And when the role based model CRO programmer eventually moves from sponsor to sponsor over time, they then are experiencing multiple best practices and implementations of SAS. Many SAS programmers prefer to remain in one constant system so that they can get the most out of their efficiency. However, there are also many SAS programmers who want to experience the breadth of SAS implementation throughout their career. The deliverable based model has more of a constant implementation of SAS, and the role based models offer the variety of implementation. This can be a consideration of a CRO SAS programmer's daily job satisfaction.

**Freedom to program your own methods:** When in the client system, you must abide by their programming SOPs. There may be situations where their practices do not equate to your usual mind set. For example you may be a routine PROC SQL user and the sponsor company has required that the DATA step be used instead of PROC SQL in all but extreme cases. It would not be wise to use your SQL commands as frequently as you would normally would in this case. Another example is suppose the validation requirement on a 100 page listing output requires you to manually inspect 10% of the records, but your practice is to create an independent program that produces the listing and programmatically inspect 100% of the records. That may take a bit longer to do and would obviously inject more quality into that report, but the sponsor may not see that as necessary in some cases. You may not be able to create novel macros or 'invent' a system that can, say, 'build AE reports much faster'. It is often encouraged by the sponsor companies for you to always say something about doing it better and faster, but there are additional barriers to consider before making recommendations. You may not understand the current strategy of the organization; for example the organization may already be developing a system that automates the AE reports and will be ready to be

rolled out in a few months. It may not be wise to invest in your automation when it has already been done or is about to be done by others; that would create double effort. Now when in a deliverable based model where you are using your own systems and SOPs, there is a much different perspective. You are in a better position to understand and influence the strategies of your group. This can open up vast opportunities to freely program macro automation or even in your own style.

**Variety of your programming purpose:** When in a role based model, there are various degrees of sponsor oversight and supervisory responsibility. There can be times when you are working directly under the supervision of a sponsor programmer. If this is the case, then your tasks are at the sole discretion of the sponsor programmer. For example, you may be in charge of providing ad hoc report for an oncology compound. Your focus as a CRO programmer is to perform a quality job so that they keep asking you to return and do more work. In this case, the CRO programmer could be 'stuck' in this position for a long time which can limit their exposure to other opportunities. Conversely, in a deliverable based model, there may be no interaction with the sponsor and you may be part of a large team that is concurrently working on multiple projects at the same time. There would be a larger range of tasks that you could contribute to and it would be up to you to voice your motivation to do those tasks. The tasks could range anywhere from writing the requirements to data set derivation or complete validation. There also could be times that you set aside to build macros for automating some programming for future use.

When working in a deliverable based model, the SAS programmer has the opportunity to work with multiple, and thus more, sponsor companies. This does allow for more therapeutic areas and phases of trials perhaps. However, unless you are integrated into the sponsors systems and a SAS user, then you will not necessarily see the various ways that they deploy SAS to undertake the clinical data flow and analysis. If a SAS programmer has an interest in having a versatile and broad use of SAS implementation, then a role based model may be the best fit. On the other hand, if the programmer has an interest to become an expert in a constant process, then the deliverable based model may be the best fit.

## LEADERSHIP AND INTERPERSONAL SKILLS

**Financial knowledge requirement:** As indicated before, the CRO SAS programmer must tend to their utilization targets since you are the revenue to your organization. In a deliverable based model, not only do you need to be aware of your utilization, but also the realization impact of the project. That is, you can be 100% utilized on a project but if you take twice as many hours to do the job that was budgeted, then you only have a 50% realization: which is not good if you want to remain competitive and profitable. The lead programmers in a deliverable based model in most cases need to take time away from the project deliverables and account for the overall budget, usually monthly. Also, at the beginning of the project the lead programmer is usually part of the initial budget discussions, calculating the predicted time needed to perform the job based on various known factors such as volume or complexity. Some SAS programmers really enjoy the financial side of the work and of course, there are those of us that do not. When engaged in the financial aspects, there is broader exposure to other groups such as finance, or even the business development group. You can be called upon to answer some fundamental questions about your operations that other groups may not know about. It is a good opportunity to create an external focus outside of your core programming work. In the role based model on the other hand, if it is truly time and materials, then for every hour you work is billed. There are also cases where there is a flat rate monthly bill. In these cases, there may not be an emphasis on realization. Thus, in the role based model, there is less added pressure to understand the impact of the finances. If the SAS programmer wants to extend their responsibilities and broaden their exposure in the financial side, then the deliverable based model may provide more opportunity and also a bigger challenge. The financial control is always volatile. Conversely, if the programmer does not want to be bothered by the financial responsibility and focus more on the programming, then a role based model may be the best fit.

**People skills:** There is a broad range of people skills such as good character, relationship building, approachability, and communication styles. Regardless of the model, it is important for a SAS programmer to have a well rounded set of personal skills. There have been multiple articles written about this in the recent years at PharmaSUG. The emphasis is usually to improve your skills to be more effective, or to get to the next title level, or sometimes part of a corrective action plan as a result of recent performance. Remember that there is high competition in our field. Also, the CROs as a whole are in high competition with each other. Thus, a good strategy would be to always put the top performing programmers in front of the client at all times so that each CRO can claim a competitive advantage based on the skills of their people. People skills are highly considered as a decision whether or not to be placed in front of the client; it is not just the analytical or SAS skill. The role based model offers more interaction between the CRO SAS programmer and the sponsor than the deliverable based model. Especially with the CRO SAS programmer is on-site and under the direct supervision of the project manager or statistician. Thus, it is very important for the CRO SAS programmer in this model to have the appropriate people skills in addition to their other skills that make them a SAS programmer that the sponsor wants to keep around for a long time. If in a deliverable based model where you

do not interact with the sponsor, then the need for people skills for the sole purpose to dazzle the client is simply not needed. However, as mentioned above without argument, it is important in every work situation to have the appropriate people skills. In spite of both models having some degree of sponsor interaction, the bottom line is that when you are interacting with the client, then you better have your 'best behavior hat' on.

**Ability to influence change:** The author believes that a good leader is one who can positively influence change. In fact, the higher level of a SAS programmer you are, the more influence to change becomes an expectation out of your core responsibility. Recall above when it was shared that the deliverable based model allows for more freedom to program your own style. It is the same with implementing your own ideas to process, people and technology as well. In a role based model, it would be much more challenging to change the behavior of a sponsor programmer that you were working side-by-side with than that of a deliverable based model where your peer programmer is your fellow employee. Let's say that in a role based model, your peer sponsor programmer tends to talk on the phone about personal matters on a daily basis to the point where you can hear the conversations. And the quality of their work is not as good as the rest of the working team. In order to bring this observation up to that individual or their manager would need to be handled a bit differently if that same peer programmer was your fellow employee. In order to provide corrective feedback to the sponsor programmer, the CRO programmer would need to exercise caution. Oppositely, if it were in a deliverable based model where the low performer was a fellow employee, then most likely there is a clear path and expectation within your company practices to provide your feedback. In any event, we should all be empowered to provide constructive feedback. It just so happens that when dealing with the sponsor, it should be done conservatively. The scenario above was a simple example of influencing positive change with people. Imagine you are to influence a significant process or technology change within your company, such as upgrading to a higher version of SAS, or to upgrade to better performing SAS installed servers. Again, we all should be empowered to attempt to influence such changes, but obviously, the change at your own organization, and not the client's, has a much higher probability to succeed. With a deliverable based model, you can have much more opportunities to influence positive change within your day to day work environment.

## HIRING AND WORKING CONDITIONS

**Flexible work schedule:** The CRO programmer role is generally competitive and thus a hard working job. This can be perhaps why there seems to be a high turnover of SAS programmers in the CRO industry. The CRO programmer constantly needs to find ways to remain energized. A good way to stay energized is a flexible work schedule. This way the programmer can have that home/work life balance they need, or simply find time to rest after a long project. Regardless of which CRO model the programmer is engaged in, they both can require that same amount of work and time. In the deliverable based model, you generally are working with your own internal teams and abide by your company policies and culture. If a company emphasizes that people come first, then there is generally a flexible work arrangement, as long as it is not abused, does not impact the deliverable, and it is equitable to others across the company. The schedule that you work often times come down to the agreement of your team(s). Examples of 40 hour flexible work week arrangements include working Monday through Thursday, extending your daily lunch break to 2 hours a day so that you can tend to personal matters, working 40 hours per week but not on Fridays or just spontaneously taking a day off without using vacation time because you are exhausted from either work or home matters. It is more likely for you to engage in this flexibility in a deliverable based model because you do not have to worry about what the sponsor company rules are. The sponsor company too can also have flexible working arrangement for its employees. However, programmers in a role based position are not employees of the sponsor company. They can be viewed as contractors and cannot partake in the sponsor company perks; in general they would only be subject to the rules that pertain to the entire company such as an end of the year company shut down in December.

**Location of the office:** It seems that more CROs are allowing telecommuting jobs for SAS programmers. It makes sense because not everyone is willing to relocate and it can limit the CRO's ability to tap into the full resource pool of SAS programmers. Also, a SAS programmer in a Midwest state may not cost as much as a programmer on the East or West coast. Moreover, the CRO should wisely consider the use of developing countries. One can argue that when the team is collocated in a central office, then they interact much more efficiently. This may be true, but this generation is becoming more seasoned in telecommuting and thus can gain 'good enough' skills to effectively network with their peers and client. In a deliverable based model, there are benefits to the sponsor company to not have the CRO use their systems and physical location on a long term basis. Thus the sponsor company may not care where the CRO programmer is located; as long as there is sufficient communication capability. When the sponsor company leverages a role based model, there may be a need for the programmer to be located on-site with a cubicle even. The sponsor company can also request that some of the programmers be near site, so that a face to face visit to the office will not be costly. The sponsor company can also request that the teams that are formed in a role based model all be co-located at the same office. These cases of role based work obviously restrict the programmer's office location. The sponsor company as well may not care where the programmers are located in

both a role based and a deliverable based model, however, in deliverable based models there is more of a tendency to have less work location restrictions and rules imposed on the CRO.

**Employee qualification:** A programmer needs to start their career somewhere, and at one time we were all considered entry level. Clearly a working team cannot have all entry level associates. There needs to be some degree of expertise in the group. However, if there is a large enough team and the environment is right, then it is wise to hire in entry level programmers for support work. This allows for the CRO to get billable work done with a higher direct margin of profitability. It also provides opportunity to build the next generation of SAS programmers. In a role based model, the sponsor generally has more requirements on where the programmer is located as well as what level the programmer is operating at. There seems to be more of a preference for the sponsor companies not to use entry level or level I programmers, and require the use of senior or principal programmers. Whereas in the deliverable based model, the sponsor company generally does not care as long as the work is done on time, within budget, and with no defects. Thus, a deliverable based model is where you would most likely find more entry level programmers. There are some programmers who really enjoy coaching and developing junior programmers, and there are some who have less patience than others and only prefer to work with senior programmers. It does not mean that all deliverable based models have entry level programmers, although there are general tendencies to that and thus can play a factor in a programmers preference of resourcing model.

The working conditions of a CRO SAS programmer is a fast paced environment with many times more work to do than there is time available. Both models can offer a good home-work life balance. The degree of flexibility ultimately comes down to the organization’s culture and policies. However, if you are looking to choose the greatest flexibility between the two models, then your best bet may be that of a deliverable based model.

## CONCLUSION

There are some similarities between the two CRO resourcing models presented above; SAS programming remain as one of the more significant constants. With the deliverable based model, there is more opportunity to meet more people and experience more therapeutic areas. Moreover, there is perhaps a better opportunity to demonstrate leadership and influence change in this environment, including working with entry level folks who will become our next generation of SAS programmers. With the role based model, there may be less opportunity to freehand SAS code, influence change, and engage in a variety of work, it does offer a larger variety of SAS implementation and a lower emphasis on financial responsibility that can enable the programmer to focus on the core science of what we do. In either model, the CRO SAS programmer is known to be a dedicated, hard working individual where their commitment for excellence, perhaps due to the competitiveness of our role, is always a focus. We all have a desire to have job satisfaction, whether it is career progression or simply a pat on the back as a sense of accomplishment. As we continue through our career life, there may be perhaps the role based model or the deliverable based model that you will experience. Both offer the same niche of SAS programming and they both offer different opportunities of fulfillment factors that we all seek. Below is a table summary of some of the areas discussed in this paper and the impact that each model offers.

<b>Parameter</b>	<b>Deliverable-based Model</b>	<b>FTE-based Model</b>
Systems Used	CRO	CRO, Client, or a hybrid of both
Freedom to program	High – leverage own SOPs and best practices	Low – must follow Sponsor SOPs and best practices
Macro creation and use	High; encouraged	Low; situational
Variety of work; as in client, therapeutic area, phase, etc	High; potential for both depth and breadth	Low; less potential for breadth but high potential for depth
Variety of programming purpose	High, less limitation within your own organization	Low, limited to the sponsor request
People skills; such as good character, relationship building, approachability, etc	Not necessarily needed	Highly emphasized; when working directly with the client.
Knowledge/pressure of financial knowledge	High – both utilization and realization in demand	Low – utilization only in demand
Employee hire qualification	Entry level more likely	Experienced employee likely
Location of hire/Home based	Anywhere when feasible	Can have restrictions
Work schedule	Can be very Flexible	Less flexible: Core hours with client

## REFERENCES

<sup>1</sup> Satyavarapu, R. Mouly (2012): "A Comparison of Two Commonly Used CRO Resourcing Models for SAS\ Statistical Programmers." <http://www.lexjansen.com/pharmasug/2012/MS/PharmaSUG-2012-MS11.pdf>

## ACKNOWLEDGEMENTS

The authors would like to acknowledge Evelyn Guo for her contribution to this paper. Evelyn has been in each CRO resourcing model along with, and side-by-side with the authors.

## CONTACT INFORMATION

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

Name:	Mark Matthews	R. Mouly Satyavarapu
Enterprise:	inVentiv Health Clinical	inVentiv Health Clinical
Address:	4745 Haven Point Blvd	5430 Data Court, Suite 200
City, State ZIP:	Indianapolis, IN 46280	Ann Arbor, MI 48108
E-mail:	<a href="mailto:mark.matthews@inventivhealth.com">mark.matthews@inventivhealth.com</a>	<a href="mailto:ramesh.satyavarapu@inventivhealth.com">ramesh.satyavarapu@inventivhealth.com</a>

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.