

Are You An Indispensable SAS Programmer?

R. Mouly Satyavarapu, inVentiv Health, Cary, NC

ABSTRACT

Job security is a serious concern for so many SAS® programmers in today's workforce, especially knowing that many positions are either being outsourced to low cost regions like China and India or being cut/ revamped. Does that mean working towards or becoming an indispensable SAS® programmer is the solution to this concern?

In fact to become indispensable, lot of the programmers try to - monopolize a particular skill; be willing to go the extra mile; have a good attitude; be a thought leader; stay current with technology and trends; be a team player; offer solutions; be committed; do the work that matter most, but not the work that's easy; continually improve oral and communication skills; and be consistently reliable and trustworthy.

I wholeheartedly agree that each of us should become the best we can be and that our work should be developed and refined to the point that it's viewed as an art, and we are seen as the contributor or architect behind the product. By all means, each should be a great employee and always seek to become a more valuable contributor, but we shouldn't elude ourselves into the thinking that we are "the only or the best" contributor in the team. In this paper, I will outline the perspective of a programmer and a manager for being an indispensable SAS® programmer. And for every perspective as there are challenges, therefore I will key those challenges out for the programmer and the manager.

DISCLAIMER

This paper represents my own personal opinions. By figuring out these guidelines the "hard way", I have been able to learn and progress in my career quite well. Below suggestions do not guarantee success, nor are they officially sanctioned by the company where I am presently working or at any company where I have previously worked.^[1]

INTRODUCTION

What are the definitions of an indispensable resource?

- *Dictionary*: Is a strong adjective for something that you couldn't do without.
- *Programmer's*: Learn a hard skill, tool, or task that needs to be done by the team/ organization and ensure that the individual (who is the programmer here) is the only person who knows how to do it.
- *Manager's*: Team member who represents as a top 5% (or tier one) see themselves as part of the group seeking to make those around them more successful.
- *Organization's*: A corporation's employee must be able to acquire knowledge (learning), transfer knowledge (out of the head and into an information system), apply knowledge (from the information system into an actionable event), manage knowledge (execute with focus, timing, and precision) and secure knowledge (keep it from evaporating or even worse walking out the door to a competitor).

Above mentioned definitions by role or entity clearly puts us into the thinking whether the notion of being indispensable - Is it the solution to job security or should we understand the various perspectives and its challenges from all critical stand points?

PROGRAMMER'S

In any workplace, you'll find employees who are absolutely critical – that is to say that productivity would grind to a halt if they were suddenly no longer there, and it's not just C-level executives who hold this distinction; it could be a SAS® programmer at any level in the organization. In other words, the thought of being indispensable at times means that your supervisors count on you so much that without you, the productivity of the team might suffer, or at least that's the perception. Due to the above, any employee including SAS® programmers form their own definition and approach in becoming indispensable.

PERSPECTIVE

- **Writing Complicated or Messy Code** – The very first thought that comes to a mind when you put both programming and being indispensable together is someone writing complicated SAS® macros/ programs to make it efficient and also to make it difficult for others to read. Furthermore, there could be instances of programmers writing messy codes with no documentation.
- **Stay Current with Technology and Trends** - As Technology and Industry trends constantly change, to be an invaluable asset to the team and company, SAS® programmers continually learn the updates to the newer versions (for example SAS® 9.4) and keep up with Pharmaceutical industry trends. By working towards this – If you're not already, you will become the go-to person on these matters learned which makes your knowledge and skills even more valuable compared to those who are not staying current.
- **Monopolize a Particular Skill or Tool** – There is always a possibility and/ or tendency of the SAS® programmers to find a particular task that needs to be done by the organization and this programmer is the only person who knows how to complete it, which helps to develop a unique skill and if it's a tool then become the expert of this tool's execution. By doing so, you will have the chance when somebody who already has this unique skill leaves the company, and you step in to replace them. Or, you may get a chance if there is an initiative in the organization which requires the similar skills.
- **To Go the Extra Mile** – Most employees just execute the assigned tasks to meet the expectations, but if the individual is one of the few who is constantly coming up with new ideas and is taking on the new responsibilities that aren't required from him/ her, this will go a long way in making the person indispensable. Volunteering for projects that other employees don't want to work on will also solidify their position in the team and organization.

CHALLENGES

- **Non-Maintainable Product** – A program written by not following the best practices and by building non-maintainable codes might help with the job security, but in the long run, when you put such programmers on a project it gradually becomes difficult to maintain, because they make it complicated as they go, which becomes a problem for other programmers in the team to read and execute the code.
- **Improper Code Documentation** – Leads to inefficiencies due to the amount of time other programmers will have to spend in interpreting and then executing the program. Also there is a possibility of missing on the initial logic/ algorithm shared during the first round of requirements. Furthermore, lack of documentation would perhaps even lead to a situation when original programmer is taking time to remember how the logic was first applied if there is an instance when modification of the code is needed.
- **Complacency** – Webster's dictionary defines complacency as: self-satisfaction especially when accompanied by unawareness of actual dangers or deficiencies.

Complacency is perhaps one of the biggest problems we face in completing our day to day tasks. We are used to things being a certain way each time, unless the obvious comes right out and hits us. We tend to be oblivious to all of it. This state of mind can affect many things such as productivity, quality, and efficiency. For example, if a SAS® programmer monopolizes a particular skill or tool, the chances of this individual getting complacent is pretty high and would lead to a risk rather a boon.

- **Make your Supervisor's Job Easier** – As managers need help and support to do their jobs, in becoming their go to person, s/he is building a trust between the two. If this individual is always thinking of ways to help his/ her supervisor and do make their job easier, they are certainly going to count on this resource more and more.

In the process of going above and beyond for some of the tasks, it could be perceived or maybe even turn out to be true to "sucking up" to the supervisor. Be careful on how the support and contribution is provided to the supervisor.

MANAGER'S

An individual can't be really indispensable for any job, unless s/he is the founder or OWN's that organization or has 51% or more of the organization's shares. Even in that case it might not be true if you take instances like Steve Jobs. If this doesn't apply, one should always be prepared to be let go anytime. No matter how good the programmer is or the connections s/he has with the upper management, etc. – an executive management's decision that changes the corporate strategy can get you fired in a snap at any time.

PERSPECTIVE

- ***Is writing complicated or messy code the answer to our question?*** – Absolutely not, in fact this practice will certainly put the programmer on the short list for layoffs. Why? Firstly, messy code that nobody understands is obviously not following the company programming guidelines (or general-accepted guidelines), which is enough reason to get you fired. Secondly, if this strategy would work, it would give the programmer power over their manager (e.g. when s/he asks for a raise, the manager can neither say no nor fire the individual when there are budget reductions, etc.). So any smart manager will want to stop this behavior as soon as possible.
- ***Reusability/ Scalability*** - The programmer's mentality should be to write good code and teach/mentor others how to replace you, and then when you get replaced, repeat the whole process again in another team or organization. This approach scales forever.
- ***Prima Donnas interested in their own success*** – From my perspective there is a monumental difference between real tier-one talent and a prima donna who thinks of themselves as indispensable. Employees who represent true tier-one talent see themselves as part of the team seeking to make those around them more successful. Contrast this with those prima donnas who are interested solely in their own success without regard to those around them. Any organization that bestows a prima donna with recognition as somehow being indispensable is a company about ready to experience a completely avoidable disaster.
- ***Be Productive, but don't make it a race*** – Most employees equate being successful and productive with doing the most work and doing it the quickest. As it's not always the case, the programmer should take their time to complete the work correctly and thoughtfully. By doing so, you uncover better and more efficient ways to get the job done and when you bring those to the table, everyone benefits from it. You're viewed as more indispensable to the manager and the team/ company.

CHALLENGES

- ***To Identify True Tier-One Talent*** - A good manager is expected to be adept at how to marshal the resources of balancing the time, cost, and personnel. To acquire that skill and be proficient in doing so, s/he should clearly - assign the responsibility for tasks and decisions, set clear objectives and measure the work assigned, monitor the process and progress with being able to measure the results, and as required design work-in-progress feedback for effective empowerment and engagement of the staff.

As every team and a leader needs an associate who is constantly and consistently one of the top performers. Hence the manager should always be on the lookout for a potential staff member who steadfastly pushes self and others for desirable results. This individual being bottom-line oriented and someone who could be counted on to exceed the goals successfully would qualify to be categorized in top 15% of the team.

NEEDED CHARACTERISTICS

As they say – nobody, I mean nobody is indispensable. It really doesn't matter who you are, what role you play, or what your title is. Therefore, in this section I will list out the characteristics and required strategies that will give a programmer a chance at being perceived correctly as indispensable, which is exactly what anyone should really care about rather getting confused between indispensable and irreplaceable.

FOR THE PROGRAMMER

- ***Clarity over Cleverness*** – While writing the code, the programmer should consider clarity over cleverness. In fact, simple code that gets the job done in an obvious way beats obscure tricks, most of the time. As they know the obscure tricks, if there's a good reason, the programmer should use one of those tricks but leave helpful comments within the code explaining how the trick works and

also provide links to any useful references explaining the idea and what the code is intended to do. With this approach, the next programmer/ team member who would utilize the program should feel smarter just from reading their code.

- **Interpersonal Skills** – A programmer should be good at person-to-person communication, as s/he should be able to explain what they're working on in a way that is more appropriate to their audience (SAS® programmers). Therefore, strong interpersonal skills result in a practice of writing code/ program that is easier for others to read.
- **Build Value** - Do your work right. Come up with brilliant solutions. Be friendly with your co-workers and help them. Be customer-oriented (if you get in contact with the customers). Accept challenges and finish them. Don't be afraid to admit your mistakes, but work hard to fix them.
- **Positive Attitude** – I know it sounds obvious, but it's easy to find people at work place who wouldn't appreciate their job at all and much harder to find someone who would make the office a pleasant place to come. Typically everyone likes working with individuals who seem like they are happy to be there, and if manager were to choose between the two employees who are equally good at their jobs, s/he is more apt to axe the grumpy one.

FOR THE MANAGER

- **Eliminate Unnecessary Dependencies** – Here's the thing, a well-managed company or its manager does not allow becoming dependent upon the performance of any single individual. Those individuals who attempt to hoard knowledge, relationships, or resources to attain job security should not be valued or viewed as indispensable, but should be admonished as ineffective and deemed a liability. Corporate talent that cannot be shared, duplicated, distributed, or leveraged is not nearly as valuable as talent that can be. Therefore, as a manager, s/he shouldn't allow a programmer to create ultimate domain over anything that is considered key or mission critical.
- **Offer Solutions** – There is always something to complaint about, it might be the poorly designed eShare platform, the way the warehouse is organized, or the computer system that has some really whacky glitches. Whatever it is – instead of commiserating, the manager should offer ideas to fix it. S/he should work closely with the team to get the work done and stick around through the completion of the task.

CONCLUSION

In today's programming world, manager should encourage SAS® programmers to be a valuable part of the team and maximize their contributions to others, but under no circumstances allow anyone to become the proverbial cog in the team or organization. By doing these things the manager will add breadth and depth to the team, and increase the overall level of talent in the group and across the organization.

All of these create a culture in the enterprise that values transparency, knowledge management, mentoring, coaching, process improvement/ management, and innovation.

REFERENCES

- Satyavarapu, R. Mouly (2014): "A Guide for Front-Line Managers to Retain Talented Statistical SAS Programmers." *Proceedings of the Pharmaceutical Industry SAS® Users Group. June 2014.* <http://www.lexjansen.com/pharmasug/2010/MA/MA06.pdf>
- ^[1] Minjoe, Sandra (2006): "How to Get Promoted: Planning for Career Growth." *Proceedings of the Pharmaceutical Industry SAS® Users Group. May 2006.* <http://www.lexjansen.com/pharmasug/2006/Management/MA03.pdf>
- Myatt, Mike: "Seth Godin's Linchpin Theory: Sound Advice or Career Suicide?" <http://www.forbes.com/>
- Smith, Jacquelyn: "17 Ways to be Indispensable at Work." <http://www.forbes.com/>

Are You An Indispensable SAS Programmer?

CONTACT INFORMATION

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

R. Mouly Satyavarapu
inVentiv Health
1001 Winstead Drive, Suite 200
Cary, NC 27513
Phone: 919-355-2214
E-mail: ramesh.satyavarapu@inventivhealth.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.