# MS12: Managing a Team of Veteran Programmers
## Some Tips for Discussion

Todd Case, Biogen Idec, Inc., Cambridge MA
Arthur Collins, Biogen Idec, Inc., Cambridge MA

## ABSTRACT

A recent Google search of 'Managing Veteran Programmers' resulted in over thirty nine million results, the first of which was 'Allentown's Mayfair festival to have veteran programmer help book…' While there are many books on 'Management' it only takes a moment to realize that they cover just about everything except managing SAS programmers! This underlies in part the interest in the topic and the necessity of this paper. The purpose of this paper is to open the management dialogue around this topic and discuss managing veteran programmers specifically.

## INTRODUCTION

Managing is part art, part science and all about understanding, communication, dialogue and presenting content. This comes into play in different ways with junior and senior SAS programmers. While junior programmers (as a general rule) tend to ask their manager for more direction and advice, senior programmers may tend to rely more on peers, networks, or user groups they have worked with in the past for technical solutions. In addition, as SAS programmers become more advanced in their careers they become more skilled and learn more advanced techniques which allow them to be remarkably efficient. On the extreme end, a very advanced SAS programmer can have a massive arsenal of techniques (e.g., substantial experience in 'procs' and 'data steps' where they do not to refer to a manual) and efficiencies (e.g., macros developed over time that are very flexible and allow easy generation of validation code, T/L/G code, etc.) at his or her disposal, and perhaps not see or think they either are, or need to be, managed. At this point, it should be noted in this case that being 'managed' specifically means there is a reporting structure and managers by nature will have all types of programmers reporting into them. This paper will present some tips on effectively managing *individual* programmers with significant experience and, even more specifically, it will zero in on effectively managing a *team* of SAS programmers with significant experience (significant meaning each member having as much or more clinical programming experience as their manager). This paper will raise and discuss the following topics in more detail regarding effective management: (1) 'Be the example', (2) 'Be available', (3) 'Remain on the cutting edge of technology or the niche of some technology or movement', (4) 'Network', (5) 'Listen, Respect and learn', (6) 'Know when to give feedback', and (7) 'Lose your ego!'

## NUMBER 1: BE THE EXAMPLE

This is something that we hear (and say!) quite often, both at work and outside of work. For example, be the first one to complete required training and training updates, be the first one to have your self-assessment complete, not asking a programmer to stay until midnight unless you are prepared to sit and stay with them until midnight as well. It also means being in the office during core hours, opening up your calendar or printing it on your office so that your reports (and others) know where you are, remaining in your office when you are not having meetings, etc. Being the example has two major implications, (1) it can lead programmers to emulate your

behavior, and (2) when programmers are exhibiting less than desirable behaviors or not performing adequate work, it is much easier to be mentored by someone who is 'The Example' rather than by a manager who perhaps takes shortcuts, etc.

## NUMBER 2: BE AVAILABLE

When we first wrote this, it said 'BE IN THE OFFICE'. However, with the mobility of programmers and contractors, and the seemingly higher prevalence of being able to work from home (and higher prevalence of managing programmers and contractors in different sites, countries, etc.) this was changed to 'BE AVAILABLE' (with the exception of necessary business trips, illnesses, etc.) - being in the office  as much as you can, responding to all correspondence (email, vmail, etc.) as soon as you can, having your office phone forwarded, and using Instant Messaging software every day, day after day. This has many benefits, some of which include: letting programmers know that you are not 'above' them or taking liberties they may not be able to take, being in the office for any ad-hoc discussions that arise (often these ad-hoc discussions can become very engaging and meaningful to managers), being visible to both programmers and other functional areas, getting to know people from other departments, etc. One of the best days for me to network with my own department and to get to know people from other departments is Friday, when many people are 'telecommuting' or taking long 3-day weekends in order to use the vacation allotted to them. On these days I take the time to stop by and talk with people that I don't normally have a chance to speak to and others stop by my office to informally ask questions and discuss ideas.

## NUMBER 3: REMAIN ON THE CUTTING EDGE OF TECHNOLOGY OR THE NICHE OF SOME TECHNOLOGY OR MOVEMENT

Managing programmers based on prior experience or climbing the corporate ladder is not enough. As referred to in point #1, it is critical to be an example to your team. For me, this means not only going to conferences, but presenting at conferences, networking  through professional associations (LinkedIn is a great way to see the groups which other programmers in our industry are participating in), and at times programming in SAS to ensure that you still possess the skills to assist if necessary. It could also be programming in a specific niche such as maintaining the AEs of Special Interest for a specific project or looking across projects for consistency in naming conventions, etc.

## NUMBER 4: NETWORK

With the number of companies downsizing and outsourcing it is critical to network with everyone you may find yourself working with, both inside and outside of your department, organization, or country!  As a manager of veteran programmers, being visible within the industry gives you greater credibility and extends visibility to your direct reports by association. One excellent way to network is to present and volunteer at trade conferences such as PharmaSUG®, SAS® Global Forum, etc. This is also another way to lead by example.

## NUMBER 5: LISTEN, RESPECT AND LEARN

While this just happens to be the 5th point discussed in this paper, it could very well be in the top two in terms of impact. When you are tasked to manage a group of veteran programmers there is usually a very good reason for this (perhaps each programmer has a lot of experience, but not specific experience with CDISC, Submissions, Standards, etc.). So as a manager in this situation it could be very critical to listen to your employees during 1:1s for the first several months. Not only will you learn more about their skillsets, interests, and motivations but it will

also help you learn about the context and culture of the company, the status of projects, and help you begin to have a better grasp of where you can interject with the most impact (i.e., and where you may have built up some credibility)

## NUMBER 6: KNOW WHEN TO GIVE FEEDBACK!

Based on all the above points about listening, respecting and learning there are still times when even the best of the veteran programmers can benefit from and should receive feedback. For example, when there is a newer and/or more efficient way to do something (e.g., perhaps using the enhanced graph abilities in SAS 9.2) it may be a good time to present the more efficient process, or even evaluate whether it really is indeed more efficient. Timing is important though so if a programmer is working furiously to meet a deadline and they are using an older or less efficient approach it is probably better to wait until she or he has met the deadline. As a general rule of thumb it's good to provide constructive feedback *after* both the manager and programmer have worked out a difficult issue, and to provide positive feedback *as soon as you witness it*. Lastly, never wait until an evaluation (either mid-year or end-of-year) to provide constructive feedback!

## NUMBER 7: LOSE YOUR EGO

You don't have to know everything or always be right, veteran programmers all have areas of strength where they are likely to know more than you so the important thing is to show them what you have to offer. Essentially, you have to give them an answer to the question 'what's in it for me?'. This may seem simple on its face because you are the boss after all so what you say goes, but that approach will not work, especially not with highly skilled and experienced programmers. Figure out how you can support them and help them to become even better, to achieve something very difficult, or to get to the next level. Show them why they need you and you can have a mutually beneficial collaboration.

## CONCLUSION

This paper has attempted to summarize some ideas to help managers get the most out of a team of veteran programmers. To use baseball as an analogy, sometimes the team with the best players does not always win, but often the best *team* wins. Modeling the behavior that you want your employees to exhibit, being present, and making yourself available will ensure that you are an integral part of what is going on in the organization and put you in a position to hold others to high standards. Keeping on the cutting edge of technology and networking both inside and outside of the organization will give you greater visibility and enhance your credibility with employees. Learn as much as you can from your employees and show a great deal of respect for their ideas and knowledge. It is impossible and not necessary for you to know everything that your veteran programmers know. Listening closely will enable you to help your employees assess their strengths and identify development needs. This kind of collaborative relationship will help to make them more receptive to your feedback and instill confidence in your ability to help them.

## CONTACT INFORMATION

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

Todd Case                          Art Collins
Biogenidec, Inc.                   Biogenidec, Inc.
14 Cambridge Center                14 Cambridge Center
Cambridge, MA 02141                Cambridge, MA 02141

Work Phone: 617 679 2633
E-mail: todd.case@biogenidec.com

Work Phone: 617 679 2633
E-mail: arthur.collins@biogenidec.com