

Best Practices in Managing a Global SAS Programming Team

Yi Zhang, GSK Vaccines, Amsterdam, the Netherlands

ABSTRACT

In the pharmaceutical sector, it has become the overall business model that companies utilize mainly contractor programmers or CROs to develop, maintain and support tools used for clinical data collection and cleaning, statistical analysis and reporting, regulatory submission and analysis result data web posting. Programmers could be located globally and within different culture and time zones. Managing a global programming team thus poses various challenges, especially with projects running in parallel with stringent delivery time lines.

INTRODUCTION

This paper discusses the best practices which have been used in managing a global team with 1 permanent and 13 contractor SAS standard programmers in GSK Vaccine (ex-Novartis Vaccine). The effective management has led to the improvements of programmers' soft skills and motivation in learning and taking on challenges. The measurable results are timely project deliveries of CDISC SDTM legacy data conversion, ADaM implementation, ClinicalTrials.gov (CTgov) and European Trials Database (EudraCT) web posting automation and Unusual Pattern Analysis (UPA) automation in less than three years period. In addition this paper also covers the techniques of building a winning team, collaborating, solving conflicts and promoting innovation.

Below I list the top ten best practices used:

1. BUILD TRUST AMONG THE MANAGER AND THE PROGRAMMERS

In my team, more than 70% of the programmers work remotely, thus using time clock systems to monitor programmers' working time is not an option. Instead, performance evaluation based on agreed deliverables and timelines is the key to ensure that tasks are done in a timely manner. To achieve this goal, the first step is to build trust among the manager and the team members. When the manager radiates the trust, the team members perform much better because they know they are trusted to perform at their best. They don't want to ruin this valuable trust. However, trust doesn't mean to totally leave the programmers alone to do their work. Periodical 1 to 1 discussions, either via face to face or via teleconference, are a must. This is to ensure that the manager is always available to support and give advice when needed and the programmers don't feel abandoned. One key team member for a mission critical project once evaluated my support as "In difficult moments I have always felt her support and her encouragement to not give up and to pursue the final results, and never experienced negative pressure". I was deeply touched by her comments, because she was working 100% remotely and was a very respectable technical expert, I was just doing what I thought a manager is supposed to do. We mainly addressed issues and sought best solutions together. Only then I realized that my strong desire for the team to win, my enthusiasm for work and positive attitude towards problems had indeed radiated energy to the team members and kept them going!

2. SETUP CLEAR DIRECTIONS AND GOALS TO BE REACHED

Vague or too high level instructions are one of the main causes of repeated work and task delay. If the team members interpret goals and objectives differently the project will face big risk of failure. It is extremely important that at the start of the project a team is formed in which clear goals and objectives are understood, supported and aligned.

I speak from my project management experiences. Two years ago when I started to lead the ADaM project, I immaturely believed that as long as the team members had great passion and enthusiasm for this project, all would work out. Unfortunately, soon after, I found that there were big differences in terms of how the team members foresaw the end product. I was heading the standard programming team and I truly believed in the concept of building validated SAS tools to make the ADaM implementation as much automation as possible. However, some of the key team members who came from a CRO background didn't share this vision, they would rather promote double programming and study level implementation. These concept differences caused team friction and led to big conflicts. That was my most stressful period in my career. The pain was almost unbearable because we all worked so hard for the project, however we were failing because the team was working towards two different directions and for different goals! After four months' constant and energy exhausting debating and struggling, finally the upper managers made the decision to split the team. Though, at the end the project was delivered by my team, I learned an invaluable lesson about the importance of building the right team before a project kicks off. For the following two critical projects (Ctgov/Eudract web posting automation and UPA automation), I paid particular attention to the formation of the team.

I selected the best suitable team members who shared the same goals and agreed upon the directions. These two projects went smoothly and delivered ahead of schedules. The more important, the team members enjoyed working together very much and we were so proud of our achievements and the great cost saving brought to the company because of the tools we developed.

I would also like to argue about the interpretation of delegation. Managers have quite often been told to delegate because they carry too many responsibilities on their shoulders. Delegation is by no mean totally transferring the work to the team and freeing yourself for other tasks. The delegation should only be started after the team goals and objectives, roles and responsibilities are discussed and agreed upon. The manager delegates the project execution to the team members, however, it is still his/her responsibility to monitor the execution. Any obstacles encountered during the project execution should be discussed and resolved as soon as possible.

3. BE KEEN IN UTILIZING PROGRAMMERS' STRENGTH AND ASSIGN THEM WITH THE MOST SUITABLE TASKS

Everybody has strengths and weaknesses, so have our SAS programmers. The manager's job is to make the best use of the programmer's strength, particularly for time critical projects. It could be disputable that individuals' interests (e.g. love learning and utilizing XML language) could be transformed into the strengths for their work. However I have indeed noticed that when programmers work on tasks in which they have great interests, the tasks are done in a more effective, timely and higher quality manner. They are even self-motivated to dive in more technical details beyond the working hours and deliver the tasks to satisfy both company needs as well as their own learning satisfaction. To create this win-win situation, it has become my routine that every time a new programmer joins the team, I will have a 1 to 1 chat with the programmer, let him/her tell me his/her interests in career and daily life. I will observe those interests during the work. After 3 months to half year, I will ask them again to see whether the same interests are still there. I might also friendly ask whether they have a five year goal to reach. Personally I truly believe that setting up realistic and reachable goals has a great impact on personal skill and life quality improvement. Those who always have goals for their life enjoy life more than those who don't have.

To retain talents, managers should demonstrate their care for their team members by not only looking at their delivered work, but also paying attention to their personal development and interests, when possible, assign the tasks which interest them the most. Even when it is not possible, it doesn't hurt to say that "this time you got this a bit boring task, but I am still aware of your interest in ...and...". You imply that you are very considering, you will still assign interesting work to them whenever you can as long as they commit to any task assigned to them.

I treat contractors almost the same way as permanent programmers, except that they are in different payroll system. The concept behind this is that training a contractor could cost as much as energy, time and money as for a permanent hire. I am not in favor of quick turnover for contractors. When they sense your caring and non-discrimination (contractor vs. permanent), they feel that they belong to the family, they will be more willing to take the ownership of the assignments, be proud in a winning team and contribute as much as they can.

4. BRAINSTORM AND INSPIRE INNOVATION

In the development field, we design new tools to improve efficiency and quality. Many new ideas could come from the programmers. Therefore it is ultra-important that they are given an environment that encourages innovation, exploration and creation. The manager's job is to be a good listener, open minded and good at collecting useful ideas, summarizing and leading the team to reach the final agreement. Of course, the manager's own knowledge and experiences, helicopter view of the project objectives and good understanding of correlations with other projects are also essential in helping making good team decisions. The manager inspires and embraces brainstorming and innovation, but the manager should also know the business well, be able to speak the same languages as the programmers and gain mutual respects.

5. EMPHASIZE INDIVIDUAL OWNERSHIP AND ACCOUNTABILITY FOR THE ASSIGNED TASKS

Programmers should be trained to follow the defined processes and procedures (e.g. a process flow chart). They will independently perform the tasks without being chased by the manager once direction and objectives are made clear to them. Moreover, they should know well the ins and outs of their tasks and the steps to follow. To achieve this, I have been emphasizing a lot in the team for setting up the right mindset, taking the ownership of the tasks assigned to them and being accountable. We have a track record of meeting timelines, the key factors for success are the commitments and sense of urgency that every programmer has for the assigned tasks. We have reduced many email exchanges and made communication more effective and avoided unnecessary confusions. As the manager and project team leader, I have been more and more able to delegate, focus on managerial tasks and be away from hands-on programming as a programmer.

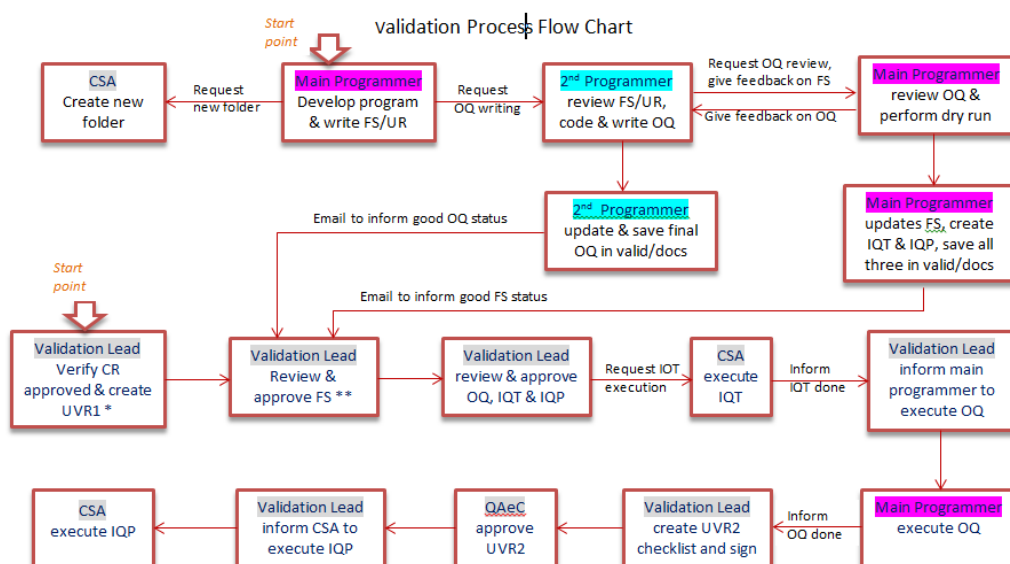
6. SOLVING CONFLICTS IN AN OPEN AND SINCERE WAY

Working in a team is like living in family, surely there will be disagreements and conflicts. Each individual has his/her own character, ideas, beliefs, strengths and weaknesses. It is the matter of how to solve conflicts in a most

constructive way. With a team consisting of programmers from 7 different countries working mostly remotely from different locations, I have been quite caught in the middle of technical and process disagreements. Being aware of cultural and individual character differences, I have been using open and sincere ways to handle conflicts, I focus on matter, e.g. the root cause of the conflict, but not on person, particularly avoid finger pointing to a person or generalizing issues. Once, I met a high level manager who had an India background. During the chat, he told me that when he worked for a Dutch company, the Dutch colleagues had very harsh debates in the meeting room. But right after the meeting, he saw the same colleagues drinking coffee together next to the coffee machine. He felt that this was very interesting and different because this would not happen in India. I learned this mentality difference from his comments. In the office, I tend to discuss disagreements with Indian contractors in a more gentle way while with Dutch colleagues I am more open and direct (the Dutch way). I am a big advocate of 1 to 1 and pre-meetings. In 1 to 1 or small pre-meetings, team members tend to be more open and willing to speak out their thoughts, concerns and wishes. I have collected some quite good ideas from these small meetings and then presented them and agreed upon in the group meetings. I am also very much in favor of using small pre-meetings to prepare proposals and think through all the details, then using group meetings to present proposals and getting the agreement. This has been proven very efficient and effective in using meeting times.

7. STANDARDIZE PROCEDURES AND PROCESSES TO IMPROVE EFFICIENCY, CONSISTENCY AND REDUCE REDUNDANCY

This is particularly important for the program validation. Validation is very process driven and time consuming. Without standard SOPs and/or work instructions to follow, each programmer could end up doing what he/she thinks is right, causing poor documentation, incompliances and lost time. The following validation flow chart is a good example of how a well-defined process could streamline the process and improve productivity. What noticeable is that programmers and validation leads like the process flow chart more than pages of work instructions in word documents. With this flow chart printed and in hand, it appears that the process is followed and remembered better.



8. TRACKING TASK STATUS WITH A VALIDATION TRACKER

It is a very challenging task for a programming manager to keep track of the status of all projects and tasks running in parallel and what each programmer is busy with. We do not always have the luxury to have project managers to plan and coordinate the projects. To effectively track the work progresses and make them transparent to not only the programming team, but also to stakeholders, we have been using a Validation tracker, which has been proven to work well. This tracker is illustrated below:

PROGRAM CATEGORY	PROGRAM NAME	CHANGE DESCRIP.	VER.	TASK ASSIGNMENT	UVR1	UR/FS WRITING	PROGRAM STATUS	IGT	FS/UR & CODE REVIEW + OR WRITING	OR	UVR2	IGP	UVR3	PLANNED DELIVERY DATE
CDM PROGRAM	METADATA_RECON_REPO_RT.SAS	Create reconciliation listing table	1.0	FS, Pgm: John OQ Writing: Mike OQ exe.: John VTL: Paul	Approved	UR Approved	Programmed	Executed	Done	Executed	Approved	Executed	Approved	31-Jul-15
B&SR STANDARD PROGRAM	CDR_INTERMEDIATE_POOL_DATA.SAS	Different bug fixes	1.0.1	FS, Pgm: Anna OQ Writing: Andy OQ exe.: Anna VTL: Paul	Approved	In-Progress	In-Progress							4-Sep-15

In each row, an individual program is listed, roles and timeline are agreed. From the UVR1 to the UVR3 column, the validation status is recorded.

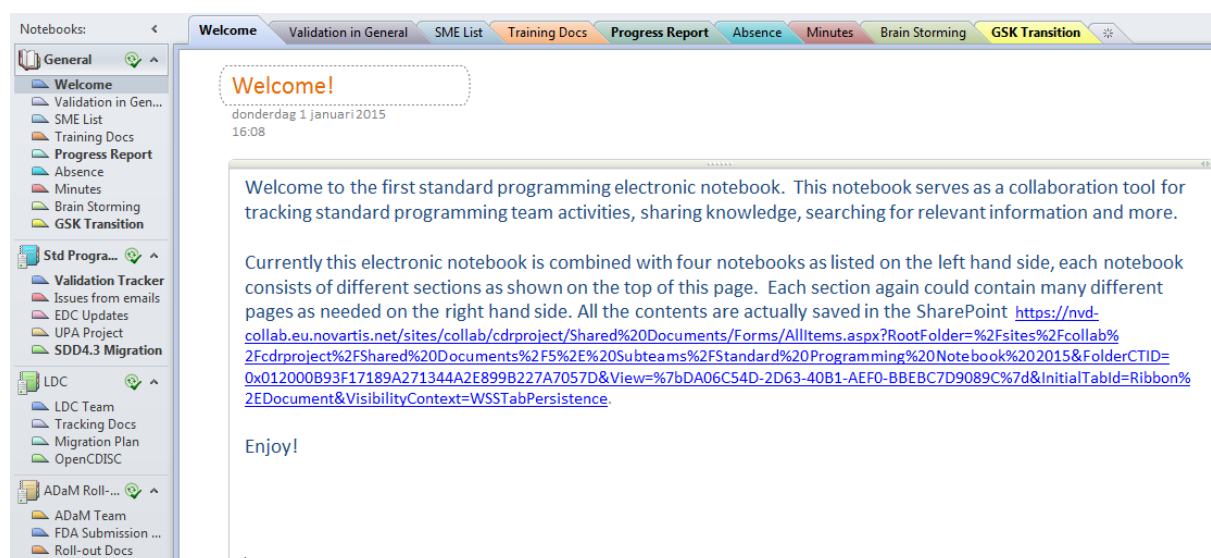
For batch of programs in a project, they can be included in the same tracker, but in a separate sheet. In addition, program enhancements or bug fix requests could be also included in a separate sheet in the tracker. This tracker should be updated daily by team leader, validation lead and programmers.

This tracker is used as the master piece for all standard programming related activities. It also helps upper managers and stakeholders to have a overview of the project status and resource allocation.

9. USE TOOLS TO IMPROVE COMMUNICATION AND WORK EFFICIENCY

Improving communication among the global team has been always one of my focus points. Besides encouraging face to face communication, if possible, and utilizing Microsoft Lync and teleconference, I have been also promoting the use of Microsoft OneNote as the team collaboration tool.

The screenshot below is the welcome page of our OneNote collaboration tool. There are four notebooks presented on the left hand side. The first notebook is "General", which contains all the information that should be shared among all programmers. E.g., in "Absence" section, all programmers enter their planned absences. The Validation Progress Flow Chart mentioned above is in "Validation in General" section. "In "Progress Report" section, programmers post their weekly activity reports including expected tasks in the following week. By one click, instead of receiving 14 reports weekly, I am able to see how much work the programmers currently have finalized and whether there are enough tasks for the coming weeks. In "Brainstorming" section, good ideas and suggestions are encouraged to be posted there. All other three notebooks are categorised per project. The project related info and useful links are posted there. The Validation Tracker mentioned above is in the second notebook.



With the help of Onenote, team members can not only focus on the tasks assigned to them, but also have a good overview of the activities of other team members as well as ongoing projects. This gives them a better sense of engagement in the team.

Besides using Onenote, we are also using monthly programmers' meeting to give technical presentations. The programmers take turns as the presenters. This is to encourage knowledge sharing in the team, so that team members could be prepared for other possible challenges besides the work which they are familiar with. This is also for the purpose of improving programmers' soft skills, such as good presentation and communication skills.

10. CELEBRATE THE ACHIEVEMENTS AND PRAISE THE PERFORMERS

As a manager, celebrating achievements and acknowledging good performance shouldn't be missed. This is to show your gratitude and appreciation for the hard work from the team and encourage continuous good performances. For every project achievement, we use team lunch, dinner or other team activity to embrace the team spirit. I also use 1 to 1 meetings to express my personal appreciation to the team member for his/her great contribution. Being generous in acknowledging achievements is much more effective and motivating than criticizing. For underperformers, face to face, hart to hart discussion about the root cause of poor performance is the key for a constructive conversation and asking for performance improvement. Being soft on the person and hard on the matter will help to reduce the tension during the conversation.

CONCLUSION

Managing a global of SAS programming team proves to be challenging, however, with the right mindset, methodology and hard work, we can turn the team into a pool of intelligence, a team with high energy and high capability to deliver any product needed in a speedy and quality manner.

CONTACT INFORMATION

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

Author Name: Yi Zhang
Company: GSK Vaccines
Address: Hullenbergweg 83
City / Postcode: Amsterdam 1101CL
Work Phone: 0031-20-5460332
Email: yi.w.zhang@gsk.com
Web: www.gsk.com

Brand and product names are trademarks of their respective companies.

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.