PharmaSUG China

Managing the analysis programming effort for an NDA submission

Yu Ella Cheng, Eli Lilly and Company, Shanghai, China
Quan Jenny Zhou, Eli Lilly and Company, Indianapolis, USA

ABSTRACT

Preparing for an NDA submission takes a lot of work, and statistical programming plays a key role in making sure sufficient and accurate information is provided to the regulatory agencies. This paper presents the experiences and lessons learned from managing the statistical programming effort to support a recent NDA submission to the FDA. We will present the challenges, and share what we did to reduce risk and ensure efficiency and quality. We hope the topics covered can apply and be beneficial to most other programming projects.

BACKGROUND

The study discussed here is a large (N>1000) oncology phase III pivotal study supporting an NDA submission to the FDA. It was designed to demonstrate a significant survival improvement in a difficult-to-treat patient population where there had been unmet medical needs for the cancer patients group. After the database lock in early 2014, the e-submission package was ready and submitted in 3 months, and an approval was granted 7 months after the submission. By the time of FDA approval, we have generated a total of 52 SDTM datasets, 25 ADaM datasets and 900+ TFLs. Figure 1 shows the key study milestones and the different focus at each stage.

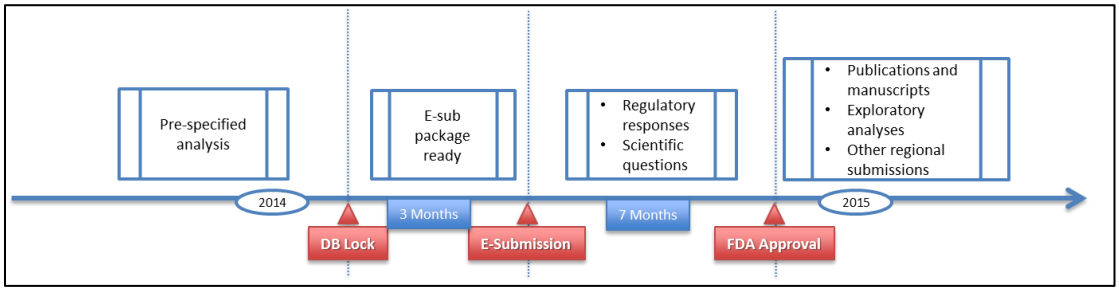

Figure 1. Key Study Milestones and the Different Focus at Each Stage

Pre-specified analyses were outsourced to a CRO, and they developed the programs on their own system following their SOP and processes. Starting from database lock, outsourcing was reduced to the minimum, while the majority post-hoc and exploratory analyses were developed using internal resources. The statistical programming team was located in China, while the rest of the study team was in the U.S. When the study first started, one lead programmer was assigned to oversee CRO deliverables as well as to program for ad-hoc requests. Starting from database lock, additional programmers were assigned. The number of additional programmers depended on the work load, 2 full-time when the work load was at its peak. Figure 2 shows the change over time of internal programming team's work load and resources. You can see from the picture that the programming demands peaked at the time of submission and after the approval, and slowed down in between.

This paper is not intended to elaborate on the regulatory requirements for NDA submission or how to prepare the submission packages. Instead, we will focus on how we managed the project, share the challenges we faced, and discuss our experiences of what made it a successful submission. Furthermore, we will also present some thoughts hoping to trigger discussions for future improvement. We hope these topics can be beneficial to most other programming projects.

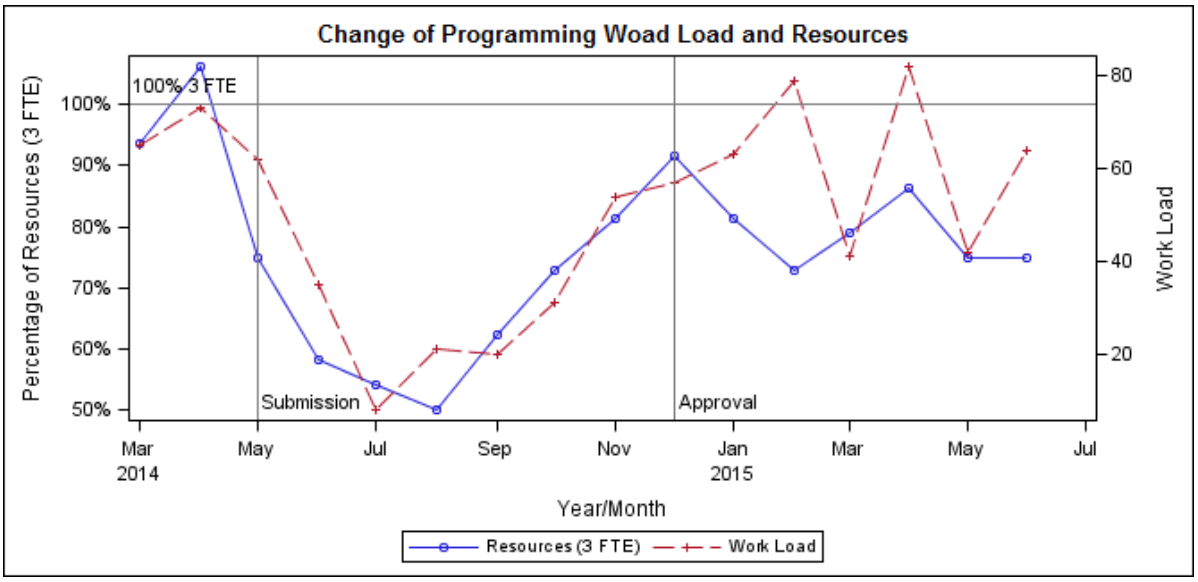

Figure 2. Change over Time of Internal Programming Work Load and Resources

CHALLENGES

Like any other project, managing and coordinating the programming efforts within a team can be challenging. In addition, the nature of a submission and how the team was setup made it more challenging, in the following ways:

HIGH VOLUME OF POST-HOC REQUESTS

Given the importance of the study, and the large sample size that usually means more data mining potential, there have been a lot of interests from other functions within the company as well as external agencies and the medical community on the study results. This means the analysis requests after database lock can be overwhelming.

As mentioned, by the time of FDA approval, we have generated more than 900 TFL outputs, in addition to the SDTM and ADaM datasets. Among the 900+ outputs, about 350+ (40%) were delivered by the CRO, and the majority of the programs were created prior to database lock. The rest of the 550+ (60%) outputs were programmed among 2 or 3 internal programmers, during the 10-month period between database lock and approval. All outputs from the CRO, and about 80% of the outputs programmed internally required independent replication as validation, which means a set of 800 validation outputs had to be created independently. That added up to a total of 1700 TFL outputs.

Figure 3 shows the amount of outputs programmed prior to the FDA approval. You can see the amount of post-hoc TFLs before approval doubled those from the pre-specified analysis.

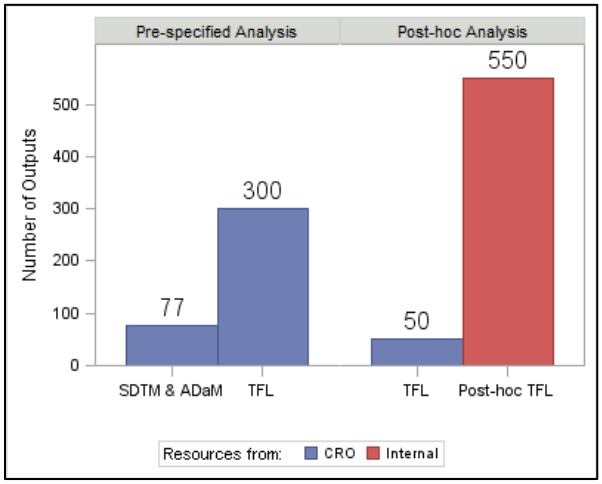

Figure 3. Work Load Comparison by Stage: Pre-specified Analysis vs. Post-hoc Analysis during the 10-month Period between Data Base Lock and Approval

Even though we only showed the amount of programming up to the FDA approval 10-month after data base lock, it doesn't mean the programming support stopped there. In fact, programming requests to support submissions in other regions, publications, and answering scientific question have been coming to the team daily since then.

SHORT TURNAROUND TIME

Different from pre-specified analysis, many questions, especially those that affected the team decision on submission strategy (like go/no go, label negotiation with FDA), and those from the agency, needed to be answered as soon as possible. Since the programming team was located in China, in some extreme cases, we were expected to deliver the results in less than 12 hours to utilize the 12-hour time difference so that the U.S. team can have the results before the next business day. Figure 4 shows an example of how we worked on an urgent request. The programmer usually had no idea what's the request going to be until he/she walks in the office in the morning. There was no time for back and forth discussion on the timeline or issue resolution. The programmers were expected to understand the request and quickly come up with proposals on how to perform the analysis, prepare a complete list of the questions, and discuss this with the project statistician located in the U.S. in a short time period. Then the programmers would need to create the output or validation independently, with no help or consultation from the project statistician, and have the validated output ready before they leave the office.

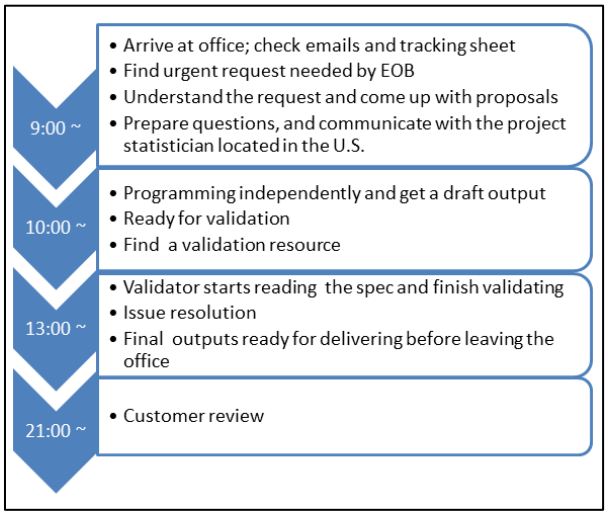

Figure 4. An example of 12 Hour Turnaround Post-hoc Request

NOT WELL-DEFINED REQUESTS/QUESTIONS FROM CUSTOMERS

Unlike the pre-specified analysis, which typically has a detailed programming specification that shows a clear requirement of output, and tells how to perform the analysis, post-hoc analysis requests were not required to have specifications. And in most cases, the requests were simply open-ending questions without a clear understanding of what output they were looking for. We needed to be able to "translate" these requests to analysis and programming specifications. Here are some examples of requests from our customers:

Q1. One of the subgroup variables on efficacy modeling uses x-month as the threshold cutoff point. We were criticized by the agency why x? Can we justify x and look at other threshold possibilities?

Q2. From the CSR AE summary, physicians see certain adverse events that might worth look into given the occurrence different between arms. So we were asked to help investigate these AEs. For example, hypertension: correlation with pre-existing hypertension? Did certain concomitant medication use cause lower/higher rate of hypertension? Time to first onset of hypertension? Duration of events? Occurrence rate related to duration of treatment?

As the owner of the analysis datasets and programs, the programmers would work with the project statistician to design the analysis and outputs to best address these questions. After understanding what needed to be done, the programmer would need to figure out how to get it done. A request sometimes took multiple rounds of discussion with the project statistician and the requester before the question could be completely answered.

OVERSIGHT ON CRO WORK

CRO played an important role in programming for this study, especially for the pre-specified analyses needed for the frequent DMCs, safety reviews, and the final data base lock. Prior to the database lock, we only had one programmer working on the study, to manage the CRO's deliverables and oversee the quality of the products. It required a well-designed oversight strategy to consider the following aspects:

- How to balance between cost and quality;
- How to effectively communicate with the CRO;
- How to follow up with the progress of the deliverables to make sure they are on time;
- How to achieve a reasonably good quality of the products delivered from the CRO and how should we verify them;
- How to maintain a heathy relationship with the CRO and build mutual trust;
- How can we help with motivating and sustaining team members in the CRO;

COMMUNICATION AND COLLABORATION

Effective communication and collaboration is essential in understanding the big picture and improving teamwork. In our case, the programming team was in China, while the rest of the study team was in the U.S., which presented not only language differences but also physical, time, and culture barriers. As the programming team did not sit with the rest of the team, we did not have the benefit of face-to-face discussions where quick decisions could be made, and all communications had to be via telephone or emails. We also found it hard to stay connected with the other functions and locate the right person when we needed help or answers. The 12-hour time difference meant that working in the evenings was normal and expected for phone calls and meetings.

TEAM MEMBER CHANGE / TURNOVER RATE

The project lasted for years, and it was unavoidable for team members to change. Table 1 shows the personnel turnover rate in the programming team for this project over time. With any personnel change, we lost some experience and knowledge. In addition, the productivity would drop at some degree. The experienced team members had to spend time coaching the new members, and taking more responsibilities while the new members were still learning. At the same time, the new ones had to overcome learning curves quickly, and made sure they adjust to the team's practices and processes to ensure consistency within the project. It had been a constant challenge on how we could control the turnover rate in the team, and how to work more effectively.

Table 1. Turnover Rate over Time at Different Stages

Stage	Number of people at each stage				Turnover Rate (%)
	Out	In	Start	End	
1: Data base lock to Submission	2	2	3	3	67%
2: Submission to Approval	1	1	3	3	33%
3: After Approval	2	1	3	2	80%

STRESSFUL WORK OVER A LONG TIME

Last but not least, supporting an NDA submission can be demanding and stressful over a long time period (several years). Achieving a work-life balance and keeping the team members motivated was important to boost the team's productivity and maintain its health.

WHAT MADE IT SUCCESSFUL?

Despite all the challenges, the programming team was able to deliver all the analyses and complete the tasks to support the successful submission. We believe the following efforts were essential in making it a success.

COMPREHENSIVE PREPARATION PRIOR TO THE DATABASE LOCK

The lead programmer started working on the study from the design phase, so she was able to prepare herself during the years while the study was ongoing. Particularly, we felt obtaining the following knowledge and experience helped a lot when working on the submission down the road:

- Know the study well. Familiarize yourself with the study design, protocol, SAP, CRF, etc.
- Obtain good understanding on the FDA submission guidelines. Be prepared to review submission documents with limited time.
- Own the deliverables even though they were created by the CRO. For any output delivered by the CRO, know what it was, how it was created, and why it was done that way. Foresee possible analysis questions and make sure the completeness or comprehensiveness of ADaM. For example: We have AE term stored in ADCM if that ConMed was caused by a certain AE. Even though there is no pre-specified analysis of ConMed caused by related AE or Grade 3+ AE, we expect similar questions might be asked, and so had AEID stored in ADCM so that we can merge with ADAE to get other information about that AE. In fact, we had AEID stored in every ADaM dataset as long as AEs are presented.
- When creating the mock tables, define all possible analysis endpoints and subgroups. For the unique tables, try to include all statistics and analysis results, and combine different categories in one table if possible.

PROMOTE ACTIVE PARTICIPATION IN TEAMWORK - ENCOURAGE EVERYONE TO BETTER UNDERSTAND AND PRIORITIZE THE TASKS

- It is very important for every programmer to understand the big picture. When a request came, understanding what information they were trying to get and why doing it that way would greatly help with delivering the outputs with high quality.
- Having the big picture also helped prioritize the deliverables appropriately. Sometimes we had to negotiate timelines and/or deliverable layouts, so that the higher priority ones could be completed on time.
- When coaching new team members, we made sure to not only show what to do and how to do it, but also explain the rationale so it can be understood better.
- It was common to use programs developed by others. Regardless of the initial author, the programmer was required to take ownership of the programs to understand the details, including the datasets, variables, stats modeling, and outputs.
- When there were multiple requests at the same time, they were assigned to programmers based on the complexity of the request and the level of experience as well as the area of expertise of the programmers. If there was time, we encouraged team members taking on different assignments to broaden their knowledge.
- Everyone was trusted to manage their own work progress, but the lead programmer monitored the timelines and would follow up and interfere if there was a risk of not meeting the timeline.

STREAMLINE THE PRACTICE AND PROCESSES AROUND CODING, CONTINUALITY, AND DOCUMENTATION

While respecting everyone's unique working or coding style, we required certain consistency within the team. That not only ensured the quality but also helped with continuality. Considering the impact of team member turnover, we established and documented certain practices for a smooth transition. These documents served as a good learning resource and guideline when a new member joined the team.

- Each programmer needed to consider the readability of their codes and the reusability by others; always includes an informative header and comments in codes
- Established naming and storage conventions of programs, outputs, and other documents
- Slightly different from common practice, our new programmers started from the role of program validator with relatively less responsibility and a later delivery. Also they usually started from their most comfortable area to help with transition.
- Though macros can be created to improve efficiency, we generally discouraged any extensive use of macros. Complicated macros are typically hard to understand and modify, and using too many macros in a program reduces its readability and reusability. From the past experience, we have been asked by the agency to remove all macros in programs submitted so they can modify and execute. So the programmers had to consider carefully when using macros, and balance between efficiency and readability.
- Use a tracking sheet to document all requests, priorities, along with the programs, outputs, programmers, dates, and status. It not only served as a central place to find all the work performed, but also helped with managing resources and calculating metrics from project management perspective.
- In many cases, we were asked to repeat the same analysis on different patient populations (subgroups). Every new analysis was created keeping that in mind, so that if can be easily used for subgroup analysis
- We encouraged complete and comprehensive documentation on work products and they needed to be inspection ready.

FREQUENT RECOGNITION & MOTIVATION HELPED PROMOTE TEAM ENGAGEMENT AND JOB SATISFACTION

- We recognized team members' contribution monthly in staff meetings, held team building events and celebrated milestones. We also encouraged knowledge sharing within the team.
- Build good relationship with the CRO through regular recognition emails. We participated face to face meeting with key study members to connect, brainstorm, and share.
- Any critiques had to be constructive feedbacks – use the sandwich format: start with positive, then constructive, and end with positive.

EFFECTIVE COMMUNICATION AND ACTIVE COLLABORATION EMPOWERED US TO ACHIEVE OUR GOALS

We can't say enough how important communication and collaboration is. Despite the language, physical, time zone, and culture barriers we faced as mentioned earlier, we felt the following communication methods helped us stay on the right track.

- Deliver expectations clearly to the customers, team members, and the CRO
- When in doubt, don't hesitate to ask. This was especially important for programmers who just joined the team.
- Know your team members and ask the right questions to the right person. This could be quite hard for us to work with team members from other functions, especially those who never met. It required extra effort in actively listening during team meetings and taking the initiative to reach out to them. It was also important to talk and listen to them knowing their perspective.
- Be open to evening meetings
- Sometimes a phone call is more effective than emailing back and forth
- Fostered an environment for the team members to communicate questions, concerns, and suggestions openly
- Be aware of other team's similar work. For example, when we needed to create the reports submitted to the FDA Office of Scientific Investigations (OSI), we reached out to another submission team who had done it before, and were able to learn from their experience and use their programs with little modification. That saved us a lot of time and effort and made sure the quality of those reports.

QUALIFIED TEAM MEMBERS ARE KEY TO THE SUCCESS

We believed it's all about the team: starting with the right people is fundamental to the team's success. In addition to being a good communicator and a team player, here are some of our expectations.

- Basic job skill set: CDISC standard, SAS programming, statistics understanding: these skills take a relative long time to develop and thus are treated as the required qualifications for an urgent project candidate
- Ownership is important; Self-driven and work oriented; able to lead a task independently and solve the problems by exerting his own energy
- Know when to ask questions and how to get help from consultants and study lead
- A good learner; understand the CRFs, SAP, clinical study report outputs, therapeutic area efficacy safety knowledge etc.
- Be able to develop his own programs; the quality of his delivery will be improved gradually
- Be open to use other people's codes

The consequence of all the efforts we made is that we successfully deliver all requests under the high turnover rate. When some people had to leave the project because of other assignments, we quickly brought in the new programmer and made the project move forward smoothly. After approval, from figure 2, you can see thatwe've been able to use fewer resources at second workload peak.

FURTHER THOUGHTS

Some further thoughts about how we can improve the experience in the future:

- How to sustain team members? It would be a huge loss if previous experience is unable to be kept. Use a continuality plan to guarantee the stability of our study team.
- How to balance between efficiency and easy-to-use/read (e.g. macros)? Macro in our case is not always helping with efficiency. Too much debugging work. But a completely macro-free program means too many future updates, bad for reuse. Our learning is that relatively independent macro developed specifically for the study might be a balanced choice.
- How can we utilize / take over CRO's programs? We nearly can't use CRO program directly because of environment change if the program itself include several CRO settings. It's also not efficient if the program itself consist of all open codes with hand writing labels footnotes, etc. Currently we only use CRO program to read the programming logic. Clearly is not the best way for efficiency.
- Be prepared for submissions in other region like EU or Japan. Plan your analyses and subgroups in advance, and have programming resources ready.

ACKNOWLEDGMENTS

The authors would like to thank all of our colleagues who reviewed this paper and provided insight comments. The authors would also give special thanks to all of our colleagues who worked on our study team.

CONTACT INFORMATION

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

Name: Yu Ella Cheng
Enterprise: Eli Lilly and Company
Address: Lilly Suzhou Pharmaceutical Co., Ltd Shanghai Branch
City, State ZIP: Shanghai200021 P.R. China
Work Phone: 86-21-23021558
Fax: 86-21-33315093
E-mail: cheng_yu_ella@lilly.com
Web: www.lillychina.com

Name: Quan Jenny Zhou
Enterprise: Eli Lilly and Company
Address: Lilly Corporate Center
City, State ZIP: Indianapolis, IN 46285 U.S.A.
Work Phone: 86-21-23020703
Fax: 86-21-33315093
E-mail: zhou_quan_jenny@lilly.com
Web: www.lillychina.com