PharmaSUG China 2019 - Paper MA-062

Risk Management on Programming Efforts for CSR Delivery

Yiqian Jiang, Janssen Research & Development, Shanghai, China;
Heidi Curinckx, Janssen Research & Development, Brussels, Belgium

ABSTRACT

Statistical programming is responsible for statistical deliverables in the analysis of clinical trials. Prompt and quick decision making is becoming a trend in clinical research and requires programming to provide a fast delivery while ensuring high quality and accurate results. To comply with this tendency, controlling risks and proactively trying to avoid them is highly recommended and very important.

This paper summarizes some risks which were jeopardizing a fast delivery in past experiences, as well as risk mitigations to reduce threats to project activities while ensuring efficiency and quality. We hope the topics covered can be beneficial to other colleagues in statistical programming.

INTRODUCTION

With the rapid development of clinical trials, end-to-end trial execution is speeding up, and decision making is accelerating. Statistical programming plays a key role in providing accurate results to the clinical teams. To meet the urgent needs, risk management becomes more and more important for the programming effort.

ICH Q9: Quality Risk Management defines the concepts of risk management. The basic steps to risk management include: identify and assess risks, mitigate risks, and review risks. Additional steps that are required throughout the process include communication of risks and documentation activities.

In this paper, we will focus on the end-to-end in-house programming business model, which risks held back a fast delivery. We will also discuss our experiences mitigating those risks.

IDENTIFY AND ASSESS RISKS

Risk identification addresses the “What might go wrong?” question, including assessing the potential consequences. Risks which might have an impact on programming activities during the course of a clinical trial are summarized as below.

NO CLEAR RESPONSIBILITY WITHIN PROGRAMMING TEAM

Within a project team, programmers are assigned different tasks and responsibilities e.g. leading the project, generating or validating the deliverables. The programming lead plans the analysis, and the team works together to achieve a fast delivery ahead the deadline. The plan/expectation for the programmer generating the deliverables is to do a first validation, i.e. self-validation. However, in practice -being under pressure for a fast delivery-, this programmer often lacks the time to do a thorough self-validation and relies on another programmer to thoroughly check the results. The validator/verifier will spend extra time and effort to figure out the rationale behind differences etc. At the end, the overall workload and communication cost will increase while the working efficiency might drop.

Programmers are assigned to several projects simultaneously and are working on various tasks within a working day. The expectation for the programmer is to prioritize tasks and complete at least the high priorities and most of low priorities by the end of the day. For junior programmers it is not always easy to prioritize the assigned tasks. Sometimes they are also not aware that they need to escalate potential delay if a deliverable is at risk. The difficulty/challenge for the lead programmer is that he is not always aware of the workload of the projects which are not his’, and therefore he is not always able to give the best advice. The complexity of working on several projects at the same time might lead to a delay in one, or in worst case a delay in all assignments for that programmer which in the end might result in a lot of inefficiency and poor team performance.

UNSTABLE SOURCE DATA QUALITY

SDTM is the source for creating the ADaM data sets for the statistical analysis of a clinical study. For that reason, good quality of the SDTM database is important and key to deliver quality on time and within agreed upon timelines. Herewith an example from a real study: In VS domain, VSTPT was collected as planned time point. In ADVS, ATPT was derived as analysis timepoint. The SAP stated that for the analysis of vital signs the observed timepoint was expected to be used with no need for any extra derivation and conceptual descriptions as Average, Endpoint, etc. The derivation rule of ATPT the ADVS metadata was “set ATPT equal to VSTPT”. However, when the SDTM data was released, VS contained plenty observations with missing VSTPT because some sites did simply not provide timepoint information when collecting data. To identify this issue, additional efforts were needed from both the programmer and the statistician to check data issues. Although querying was immediately processed, to meet the urgent analysis needs for the interim database lock, extra rules to impute the missing timepoints in ADVS ATPT had to be added and as a result an extra validation had to be performed. All these additional efforts, both in time and resources, could have been saved if the source data quality would have been good as expected.

RESPONSIBILITY GAP BETWEEN DATA MANAGEMENT AND PROGRAMMING

Data management and programming are working closely but are separate functions. Given the roles and responsibilities of programming, programmers are not expected to code for data issues or actively search for data issues. That would be a redundant effort because data management is responsible for edit checks and a data validation plan. Per process, programmers are not included in the intensive review for data management related data quality activities. The programmers are also not always heard at the early stage of CRF, database or data validation plan build. The potential risks of data quality related to statistical outputs lack opportunities to be found at an early stage. Often SDTM quality issues remain unidentified until the programmer starts preparing the deliverables (data sets, tables, figures and listings) for the statistician and the physician to be reviewed. Extra programming efforts cannot be avoided, and the workload increases in case of issues found at a late stage.

COMPLEXITY OF ADAM DATA QUALITY

The quality and correctness of ADaM data sets are key to performing a correct analysis. Programmers are responsible for generating ADaM data sets per the requirements in the SAP. ADaM quality assurance includes two steps: an accurate translation of the SAP definition to an ADaM specification (or metadata); and translating the SAP definition in programming code which will exactly does as expected to generate ADaM data sets.

Programmers generate ADaM specifications from a template or a reference study and customize those to meet study specific requirements. The expectation of a good specification is that it provides clear and logical step by step derivation guidance, concise and with enough and complete information to accurately reflect the SAP definition. To give an example: the SAP does not provide analysis guidance for ongoing status data. Although, in the end, each subject will be with a discontinuation date or a completion date, the date is not available for every subject when the study is ongoing. This situation is mostly not covered in the SAP and requires extra guidance in the ADaM specification for e.g. how to derive phase end date, whether a missing date will be acceptable etc. These extra efforts add extra difficulties for ADaM specification quality and a timely deliverable.

The main responsibility of a programmer is programming. The expectation of good ADaM programming code is that it can completely implement the guidance in an ADaM specification. However, the reconciliation of ADaM specification and ADaM coding is always complex and prone to error, for the simple reason that ADaM coding is a manual process. The consistency between codes and specifications sometimes can be easily ignored when the results are correct.

NOT WELL-DEFINED ANALYSIS

The DPS (detailed programming specifications), contains additional information and instructions for programmers when preparing the analysis, together with mock outputs, information on footnotes etc.. Sometimes, the DPS refers to previous studies or templates instead of being study specific. The definition

of derived variables is not always clear, the variables used in mock up tables are not always specified, the statistical methods in SAS procedure are not always provided etc. When it is not clear for programmers how to derive variables or which variables are necessary to be used in tables, then he needs to double check with the statisticians and/or lead programmer which increases the communication efforts within the team.

ADAM STRUCTURE NOT APPLICBLE

CDISC is the standard that we are using and gives guidance on how to generate data sets with high quality but given the design of the study it is not always the correct solution to develop all of the data sets. In some early development trials, the team might focus only on the interaction between different factors, in that case and for those analyses the ADAM BDS structure is not suitable. If we adhere to the BDS structure in case of ten or more factors, and build the parameters vertically we need to transpose the parameters into a horizontal structure when performing the statistical analysis which increases the complexity and efforts when performing the analysis.

TEAM MEMBER CHANGE

If the project lasted for years, the composition of the team changed unavoidably resulting in loss of experience and knowledge. On top of that the productivity dropped at some degree because the experienced team members had to spend time coaching the new members while at the same time had to take on more responsibilities as well because the new members were still learning and adjusting to the team's practices and processes to ensure consistency within the project.

COMMUNICATION AND COLLABORATION

J&J is a global company, the team members are located in US, EU and China. Time zone differences have an impact on urgent request and on-time feedback. We have the experience of study with a statistician based in US, data management in EU and the programming team in China. With only minimal overlap within the cross-functional teams, the communication cycle time increased. Especially the programming team did not have the benefit of face-to-face discussions where quick decisions were made, all communication had to be via telephone or emails.

HIGH EXPECTATION FROM CLINICAL TEAM

Clinical teams are more and more relying on programming support to deal with urgent or unmet needs. Having a scientific mindset, understanding the big picture is an emerging expectation from the perspective of a clinical team in the current pharmaceutical industry research area. When an issue is identified, although it might be out of the programming work scope (and not directly associated with the CSR), the clinical team may request analyses for clinical review without any thorough planning in the SAP. When providing extensive scientific support like this, the overall operational excellence might be challenged due to an above estimated resource allocation and limited resources for the CSR.

MITIGATE RISKS

After identifying risks and possible consequences, actions are needed to mitigate the severity and probability of harm. The below summarized plans might help reducing the risks.

SET CLEAR EXPECTATIONS AND PRIORITIZE THE TASKS

When holding a kickoff meeting, the lead programmer should clarify the role and responsibility for each member, emphasis the importance of the self-validation and set up a clear timelines and expectations for each task. Every programmer in the study team needs to know and understand the study well and be familiar with the study design, protocol, SAP and CRF.

Programmers knowing and understanding the big picture will help prioritizing the deliverables appropriately. Lead programmers should really coach the new members: not only show what to do and how to do it but also explain the rationale so that it will be understood better. Programmers who are assigned tasks in different projects need to proactively communicate with the lead programmers and

share the status, difficulties, challenges and worries. When a junior programmer is struggling, the lead programmer should also coach that junior programmer on how to prioritize by giving clear priority levels.

DATA QUALITY CHECK

In case of unstable data quality, the programming team puts extra effort in identifying data issues and summarizing key areas for data management to pay attention while doing the quality check. The lead programmer decides whether extra mapping rules are necessary to impute missing timepoints. The programming team can develop tools to help checking issues instead of manually finding issues.

INVOVLE PROGRAMMING IN DATA EARLY STAGE

The programmer is not always heard at the early stage of CRF, database or data validation plan building while this can be very useful as he can provide his perspective on how data quality will impact the analysis. Especially for the data validation plan, the programmers' opinion can really add value as he has a very clear idea of what the quality expectation of variables should be, and what the consequence might be of less quality.

DPS AND ADAM SPECIFICATION READINESS

A comprehensive DPS and good ADaM specifications are key to performing the analysis. With the ongoing status of data, extra analysis guidance covering possible situations should be documented if not pre-defined. It requires the lead programmer to write comprehensive ADaM specifications to ensure the good quality.

To ensure the programming code is completely performing the guidance in the ADaM specification, the QC programmer needs to check the consistency between the ADAM code and the specifications. The lead programmer needs to use Pinnacle 21 to ensure compliance with the CDISC standards.

DESIGN THE ANALYSIS DATA SET PER NEEDS

When assigned to a new project, the lead programmer needs to have a clear picture of the analysis requirements. It is highly recommended to have discussions with the team and to get to know and understand their expectations. When writing the data specifications, the lead programmer needs to translate the needs correctly and not just applying the current CDSIC or company standards.

QUALIFIED TEAM MEMBERS REFILL IN

It is unavoidable that programmers change projects for whatever reason. For a lead programmer it is essential to know both the needs and specificities of a study, and the qualifications of his team members. When somebody leaves the team, the lead programmer should be able to quickly find another qualified person to take over and by that decrease the learning curving and mentoring time. Basic job skills such as CDISC standards, SAS programming, statistics knowledge, etc. take a relative long time to develop and, hence are considered the required qualifications for urgent project candidates. On top of that, ownership is important, a possible candidate should be self-driven, work oriented, able to lead a task independently and solve the problems by experience and exerting his own energy.

EFFECTIVE COMMUNICATION AND ACTIVE COLLABORATION

Despite the language and time zone differences, we successfully stay on the right track and deliver within agreed timelines. Regular meetings with data management, statisticians and the programming team are crucial and phone calls are more efficient than emails going back and forth. In case problems cannot be solved by email, a joint meeting with all stakeholders should be set up to discuss questions, solutions and suggestions to finally come to an agreement. Afterwards meeting minutes should be send out to ensure that everybody is on the same page. The lead programmer should foster an environment of proactively communicating concerns or worries and should encourage everybody to spontaneously keep the team up to date, instead of only doing that when asked to.

CONCLUSION

To minimize the risk of holding back a fast and successful delivery, the programming team should consider all aspects to reduce possible delays of any project activity while ensuring efficiency and quality. Prior to the start of a project, roles and responsibilities should be set up clearly. During data collection, a close collaboration and a proactive communication with data management should be set up and maintained to ensure data quality at an early stage and enforce giving input for optimal data checking. For the full project execution period, a full and clear understanding of the big picture and the analysis needs, getting familiar with the data, a final and well-documented the DPS, good ADAM specifications, and performing CDISC compliance check are key for success. As important are a proactive communication with all functions involved as well as a close and transparent communication within the programming team. We always need to keep in mind that high quality is as important as meeting timelines.

REFERENCES

ICH HARMONISED TRIPARTITE GUIDELINE. 2012. “QUALITY RISK MANAGEMENT Q9.” Current Step 4 version dated 9 November 2005. http://www.ich.org/fileadmin/Public_Web_Site/ICH_Products/Guidelines/Quality/Q9/Step4/Q9_Guideline.pdf.

ACKNOWLEDGMENTS

The authors would like to thank all our colleagues who reviewed this paper and provided insight comments.

CONTACT INFORMATION

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

Name: Yiqian Jiang
Enterprise: Janssen Research & Development
E-mail: ajiang4@its.jnj.com

Name: Heidi Curinckx
Enterprise: Janssen Research & Development
E-mail: HCURINCK@its.jnj.com

Any brand and product names are trademarks of their respective companies.