

The Highs, Lows and Lessons of a First-Time Study Lead

Bethan Thomas, PRA Health Sciences, Swansea, United Kingdom

ABSTRACT

The transition from being a programming study team member, to programming study team lead can be highly challenging. Many of the tasks involved in leading are very different in nature from team member tasks. Programming of datasets and TFLs is largely logical and analytical, whereas leading also involves planning, resourcing, organising, archiving, budgeting and more. This paper will walk through an overview of several lead tasks, focusing on items which may be particularly challenging and offering some lessons learnt from experience as a first-time study lead, but also sharing in some of the highlights and rewards of leading a study and producing high quality deliverables. This may be relevant to those new to programming and hoping to lead, or those who have recently taken on lead responsibilities and are looking to share ideas.

INTRODUCTION

This paper will be separated into four major sections, exploring some general areas of lead responsibilities and then concluding with some highlights of the overall process of leading a study.

The first area of discussion will be the technical tasks a lead is required to perform. A major part of this is document review. There are many documents that programmers should be used to referencing on a regular basis and will be asked to review, such as the CRF and SAP. There are also further documents which the lead may have never seen before but will still be asked to review for cross-functional colleagues. It can be difficult even to understand these documents, let alone feel in a position to review them. Furthermore, there are documents such as dataset specifications which are the responsibility of the lead programmer to produce. However, for all these tasks, the principle is the same; a lead must be able to answer the question, 'Does this document provide a programmer with all the information necessary to perform the relevant programming tasks?' This section will explore a few examples from the different types of document, some practical thoughts about how to approach them and highlight potential pitfalls and aversion techniques.

The next area covered will be study planning. Tasks such as projecting timelines, managing a budget and resourcing may be completely new to first-time study leads, and therefore very demanding. This paper will summarise some of the key principles, without going into detail of any particular tools, as these will be company and potentially project specific. These items can seem quite far removed from regular programming work and often get put to the side-lines or even be dismissed altogether. However, if these tasks are well understood and implemented, they will enable the programming team to focus on completing work efficiently and avoid encountering unnecessary difficulties further through the process, when there are potentially critical timelines to be met.

Subsequently, the important topic of communication will be explored. Communication is vital for any functional lead, in order to execute a study successfully. It is essential that the lead has all the necessary information from other departments to support the programming work, and it is equally essential that the lead passes on all helpful information to the programming team. There are many aspects of communication to consider; from different methods (e.g. electronic, phone, face-to-face), to different participants (e.g. programming team, cross-functional colleagues, and of course the client if working in a CRO). Some methods of communication may be more suitable for certain people/purposes, whereas other situations may require different communication forms. Therefore, this section will give an overview of appropriate communication and some helpful hints to maintain good and productive working relationships, both internally and externally if applicable.

Finally, the paper will pick out some of the most rewarding parts of being a lead programmer; from producing quality and timely deliverables, to widening technical/study knowledge and professional development. A programmer's first experience as a lead may be very difficult, as it can be quite a big step up from support programming, so it can be helpful to remember the outcomes of the work and the reasons we joined the industry in the first place. Therefore, the last section will also re-focus on motivations and incentives to perform and improve within the industry and as study

PhUSE 2015

teams. Such motivations may be very different, depending on the particular programmer, but in all cases, taking a step back and remembering the bigger picture, can help us become more productive in our day to day work on a study.

TECHNICAL TASKS

ANNOTATING THE CRF

This area of leading is probably the most natural transition from programming, as the tasks tend to have a direct impact on the programming work itself. Programming team members should already be familiar with important study documents such as the protocol, Annotated Case Report Form (aCRF) and the Statistical Analysis Plan (SAP) text and shells, but usually purely as reference documents. This familiarity becomes very helpful when it comes to being involved in developing or reviewing these items. As an example, we will consider the annotation of the aCRF. If a team member is accustomed with frequently referring to the aCRF to understand the input data structure and format as they program a domain, then the thought process involved in actually adding annotations should come more naturally. In short, the aim of annotating the CRF is to aid and explain the programming, therefore the process of adding annotations is simply a case of looking at a CRF page, considering how each field fits into the target dataset (often CDISC SDTM but the same principle applies regardless) and making sure that this is adequately described in the annotation. The completed annotated CRF will then be both reviewer-ready and programmer-ready - any reviewer should be able to look at the document and understand how the CRF data will be transformed into the programmed dataset, and any programmer should be able to view the aCRF and understand how to perform the transformation.

DOCUMENT REVIEW

The Lead Programmer usually has actual involvement in developing the aCRF (i.e. creating the annotations), but there are other documents which, although integral to the programming work, may be created by colleagues from other functional groups such as the SAP, in-house database design specifications/blank CRF, vendor Data Transfer Specifications (DTS) and Edit Check specifications. The Lead Programmer may only have input into these documents at the review stage, after they have been initially drafted. It is important to review these documents thoroughly. As a new lead, it may be tempting to assume that a cross-functional colleague has all the necessary expertise in their area to produce the item, so the document that the programmer receives should be essentially final. However, as a programmer you may see things from a different perspective which your colleague has not considered, so it is important to be thorough. For example, if you are sent through a DTS, you will need to consider whether all the necessary controlled terminology has been provided in order to specify a mapping, and whether all the necessary variables and data will be present to create your output. Anything else affecting the statistical programming will also need to be considered, such as the blinding of the data, if applicable.

SPECIFICATION WRITING

Furthermore, there are documents which will very likely be the Lead Programmer's responsibility to create, such as Dataset Specifications. The task may seem daunting and very technical, but it can be simplified by considering input and output. If we consider an SDTM Specification, the input is raw data, which is defined by the CRF and Vendor metadata, the output is SDTM datasets, which are defined by the SDTM Implementation Guide (IG) and other guidance documents. Similarly for an ADaM specification, the input is SDTM data as described by the SDTM IG and datasets specifications, and the output is ADaM datasets, whose requirements are described by the ADaM IG and the SAP Text/Shells. This is obviously a simplification and there may be other study specific documents which will support the writing of specifications, and it is always important to have a good grasp of the protocol alongside all other documentation, but the principle may be helpful. The process of creating specifications is simply writing descriptions of how to transform input into output, in a document format specified by your study/client/company and supplying relevant metadata. Writing specifications can be invaluable in developing the Lead's understanding of the study, standards and processes.

PROGRAMMING DATASETS

Once the specifications have been set up, the programming team can get involved and begin programming. Depending on your time/team restrictions, this may be a step which you are heavily involved in or have very little to do with, which can be a big challenge and will be discussed within the Study Planning and Communication sections later on. It is however, a good idea to be involved in at least the programming of the most complex and key domains, especially given the understanding you will have gained through the document review and specification writing processes. It is also a good idea to work closely with your study Statistician (some programmers may also fulfil the

PhUSE 2015

Statistician role for their study) to ensure the analysis datasets contain everything they are expecting, in the structure they are expecting, so that any statistical models can be implemented correctly. The earlier these conversations occur, the more effective they will be, i.e. it is better for the statistician to review the ADaM dataset and discuss the variables/subsets which should be applied to the statistical modelling, than for the statistician to only review the TFL output and then query the numbers displayed. It would be better still if the definitions were worked out between the lead programmer and statistician during the specification writing stage, prior to commencing any programming.

TECHNICAL TASK OUTLINE

In summary, the best way to prepare for the technical lead tasks is to take an active approach as a programmer. It is much easier to perform a review of a certain item if you have been in the habit of regularly referencing that document. It is also much easier to write a specification if you are used to looking at the source documents and have a good understanding of the standards, than if you have programmed by merely using the specifications as an instruction manual. Most leads are happy to answer questions, provide explanations and generally equip more junior team members, as it enhances the ability of the team as a whole and the quality of the outputs produced, so it is worth making the most of this as a team member and also applying this as a study lead. In general, the technical tasks required of a Lead Programmer can vary in appearance, but the principle is consistent; ensuring the document provides all of the required information to produce the relevant programming output and conforms to the applicable rules and standards related to such an item.

STUDY PLANNING

INTRODUCTION

This aspect of leading is probably one of the most challenging, especially for new leads. This side of lead responsibilities is closer to project management than programming; it can require an entirely different skillset, which a programmer has not previously needed to demonstrate. It can be overwhelming as questions arrive from project management, functional management, cross-functional colleagues and programming team members and you are expected to have answers or at least know what you need to look into. However, it will only become more difficult if planning is not monitored early on in the study. Therefore it is good to have a general idea of the process flow and break it down into smaller, more manageable steps. The guidance provided here is described from the point of view of a CRO, but there are common principles which will apply for employees of pharmaceutical companies also.

PROJECT SCOPE

Begin by considering the scope of the work. The most helpful documents for doing this will be the contract, protocol and SAP so it is worth familiarising yourself with these. If you are working for a CRO, you will most likely have a contract with your client; this should inform you of the tasks that the programming team will be required to complete, e.g. SDTM, ADaM, TFLs, define.xml etc. The numbers of units given in the contract may be subject to change over the course of the study, perhaps when you begin annotating the CRF it becomes apparent that the number of SDTM datasets required may be greater or fewer than initially projected. This is normal and acceptable, you will simply need to keep track of this and keep the project finance team informed so that they can amend the numbers of datasets, TFLs, tasks and deliverables as necessary. The protocol should also help you gain an understanding of the complexity of the study and the number of domains as well as a general idea about the study endpoints. If you are performing analysis or TFL programming, it will also be important to familiarise yourself with the SAP in order to determine the numbers and complexity of unique and repeat tables, figures and listings, thus determining the number and complexity of analysis datasets.

PROJECTION OF TIMELINES

Alongside considering the scope of the work, it is important to contemplate the timeframe of the work. Again, if you have a contract, some helpful information will be in there, such as how many data transfers or Dry Runs are required and whether or not the study will be having an Interim Analysis or DMC (Data Monitoring Committee) delivery. An interim analysis or DMC may lead to more restricted timelines as they will need to occur at pre-specified timepoints, e.g. every six months or once a subset of subjects have completed all of their assessments. Conversely, a Dry Run may often be scheduled to occur once all the programming is likely to be ready. If possible, it is a good idea to have as much input into forecasting timelines as possible and to work closely with colleagues from the wider study team; if a programming representative does not have input at this stage, it is likely that programming will be allocated less time than might be preferable or even realistic. As a new study lead, you may not be sure of how much time to suggest for your assigned tasks so it is important to be well prepared; make use of any tools that your company has available, reacquaint yourself with relevant Standard Operating Procedures (SOPs) and Work Instructions (WIs), ask

PhUSE 2015

Senior Colleagues for rough ideas or even detailed breakdowns of standard timelines. Then when it comes to actually projecting the timelines, two general rules will help; firstly, provide as detailed breakdown as possible, this will justify the time allocated and ensure that steps are not forgotten, secondly, it is better to project a longer timeline (within reason) that is likely to be met than to project an earlier deliverable date which could potentially be missed.

PLANNING RESOURCES

Once you have a good idea of the scope and the timescale, you are able to plan the resourcing. It is likely that your company will have tools which can help you with this, but the following questions are a good place to start. How many programmers will you need on your team? Remember that too many may become difficult to manage, but too few may struggle to meet the necessary timelines. When do you require your resource? You may want to have a larger team initially to get the first-time outputs produced, or you may want to have more of a core team for most of the time and increase the resource in the lead up to a deliverable. What level of experience do you require in your team? Perhaps more complex tasks will require programmers of a minimum experience level, or you will need more senior team members involved in the QC of work. You may need to consider whether you will be able to be hands-on as a lead with the programming or for larger projects. The lead may simply be overseeing and managing the study, in which case, it is worth planning how you will monitor the programming and output to ensure your satisfaction with the deliverable. Some studies may be very similar to others where the programming has already been written, this may significantly reduce the number of programmers necessary. The project budgeted hours may also be restricted, causing efficiency to be a higher priority and the resourcing is affected accordingly.

FURTHER CONSIDERATIONS

Some other points in study planning to consider are that sometimes the timeline and resourcing parts may occur in the opposite order, so you have to project timelines based on your team, rather than request your team based on your timelines. Also, as implied earlier, things may change over the course of a study, there may be more or fewer tasks than initially stated, new outputs may be requested based on given results, a client may ask for ad hoc programming tasks to be performed or reports to be created. It is important to keep an ongoing log of any changes or things that may affect the overall efficiency and budget, as further into a study, it is unlikely that you will be able to easily recall everything that happened earlier and when it happened. As a new lead, don't be afraid to question additional requests, or propose things as out of scope. For example, you may receive data which is overly complex, insufficient for your purposes, or there may be a design flaw in the database which causes unexpected challenges for programming. Instead of assuming or accepting that this is simply a problem which programmers have to deal with, it is better to work with the project team to establish if the extra programming effort can be recognised and added to the scope of the contract. If not, it is at least worth keeping a record of them so that you are still able to explain the budget deficiencies where necessary.

CHALLENGES AND SUPPORT

As a new lead, study planning may well be one of your greatest challenges. Failing to plan may also be one of your greatest downfalls, as the importance of it may not become obvious until later. However, simply taking time to plan and keeping logs of how the study progresses, will really aid you in this area. Furthermore, lessons learnt will enable you to keep a closer hold of study planning in the future. This is a particular aspect of leading where the support and guidance of the functional manager can be utilized. Managers may have many reports delivering studies and thus have extensive experience of planning timelines, forecasting required resources and tracking budgets. A new lead should never be afraid to ask if they are unsure of something because every question is valuable if it aids or consolidates the lead's understanding of study processes.

COMMUNICATION

PROS AND CONS OF EMAIL

This is an area which might seem familiar and obvious, but effective and appropriate communication can actually make a substantial difference to the productivity of a study team. Items discussed here might seem basic, but they are still worth considering again and reflecting upon. Probably the most frequently used form of communication in this industry is email. Emails are quick and versatile and thus extremely useful; they can be sent to multiple participants, can include attachments and are easily archived. However, they can also be easily misinterpreted or lost. As you transition from being a programming team member to a lead, you will notice a sharp increase in the volume of emails you are both sending and receiving, so it is important to keep them well-ordered. You may even find that you now spend the majority of your working day dealing with emails. Fortunately, there is a great deal of inbuilt functionality for email and it is worth taking full advantage of this to develop a system of handling mail which works for you. Methods

PhUSE 2015

of organisation include creating comprehensive folder structures, categorising emails, using flags and reminders, marking messages as unread and even setting up rules to automate any of these processes for you. This will help you ensure you do not forget to respond when necessary and that you can easily locate emails for reference at a later date. Another helpful idea is to always have a sensible but specific message subject; often email threads can get extremely long and the topic may change significantly over time such that the subject does not sufficiently describe the purpose of the messages anymore, therefore it may be sensible to continue the thread but change the subject. Similarly, when initially writing emails, it is important to give a subject which concisely but clearly explains the content of the email to all recipients so that they can easily organize and prioritize it appropriately.

ALTERNATIVE METHODS

It can be tempting to use email extensively or even exclusively, but once leading you will have cause to be in contact with a wider variety of parties for a greater range of purposes and choosing the appropriate method and tone of communication is important. Within your programming team, there may be times when you simply wish to get a message out quickly so that everyone is notified, in which case a short, informal email may suffice. If your team is local to you, you may even choose to simply speak face to face. For more fundamental pieces of information, such as formatting standards for all TFLs or programming practices to be applied globally, you may decide the best place to communicate this is through a centrally stored document, which team members can regularly reference and check for changes. In order to build up team relationships and keep all team knowledge current, regular meetings – either face-to-face or teleconference – can be useful; during these you can stay up to date with how your team are progressing and pass on any relevant notifications. If it is necessary to make a record of the information shared, minutes can be recorded or a supporting email can be distributed.

CROSS-FUNCTIONAL COMMUNICATION

Leads will also need to communicate with colleagues from other functions. There is potential for tension to be caused here as each department's role may be very specific and technical and different functions may not be able to easily understand or relate to one another, especially for new leads. This does not need to be the case though and can be avoided by building working relationships carefully and strategically. There are some simple steps which can aid this significantly. Firstly, communicate early; it will not be appreciated if you request something at very short notice from a colleague who you have had little previous contact with, especially if it could have been communicated sooner. You are much more likely to get a positive response if you give your colleague reasonable notice, and even more so if you have already built up a working relationship. Secondly, ask questions; it is acceptable, especially as an inexperienced lead, not to understand all of the roles and processes, but it is much better to ask for the information you need, perhaps from a more senior lead or the person you are contacting directly, before responding. It is much better to clarify what is being asked of you than to guess incorrectly. Thirdly, respond to others in a reasonable timeframe; if you provide others with the information they need promptly, they are more likely to be helpful in return. Fourthly, if possible, use the phone and face-to-face contact as it is much easier to get to know someone and build a working relationship if you do not rely solely on electronic communication. You will probably find colleagues much more cooperative once you have got to know them in this way. Finally, use appropriate language; this includes simply being polite and courteous, being clear and concise, and not using statistical or programming jargon if your recipient is not also a member of the statistics or programming group.

GLOBAL CHALLENGES

Another element which is becoming more important to consider, is global/cross-cultural communication. Many companies now have bases across a number of continents and many are recognising that certain markets will reap rewards and contribute to growth in the industry. Therefore, it is more and more common to need to consider time zone, culture and language with your contacts. For example, the time a meeting is scheduled will need to fit with all attendees across all time zones. If this is not possible, multiple meetings may need to be arranged or a different method may need to be used. Team communication is generally in English, but this may not be the first language of some study contacts. In which case, it is a good idea to let those who are not English first language decide on the contact method; they may prefer to mainly use email as it gives them time to consider what is being asked and how they wish to respond, or they may prefer to mainly use teleconferencing as it gives both parties opportunity to clarify what is being asked. It is likely that both methods will need to be used across the course of the study, but stronger relationships will be built if you attempt to be as accommodating as possible. Simple things like sharing dates of public holidays and appreciating each other's working environment and leadership structure will also make working together easier.

PhUSE 2015

VALUE OF GOOD COMMUNICATION

Communication is an aspect of the lead role which can dramatically change your experience on a study, therefore the importance of effective communication should not be underestimated. Simply taking the time to determine in advance the purpose, content, tone and method of your communication will enable a lead to work successfully with both colleagues and clients.

REWARDS

TRANSITION INTO LEADING

Starting to lead studies can be a huge step up from being a team member and therefore can initially appear to require a lot of extra work. In your first few studies as a lead in particular, you may take responsibility for the work produced very personally and put a lot of pressure on yourself to deliver to a high quality. Whilst a drive for results is a good thing to an extent, it needs to be held in balance. It is important to remember that a career in this industry may be for many years and it is not expected to deliver the 'perfect' study first time, i.e. delivering early, high quality and under budget with no issues to tackle at all within the programming group, cross-functional team or with the client, if applicable. Therefore, the motivation to keep going when it gets difficult must come from something else. Below are some of the rewards that a Lead Programmer can look forward to and focus on.

RECOGNISING ACHIEVEMENTS

The small victories are highly important as a developing lead. These could vary from completing your first programming specification or gaining understanding of a document or process that you have not previously used. Such things give you confidence in your existing abilities, stretch you to learn a little more and help you reflect on what you have already learnt since you started leading. Then there are the bigger achievements such as sending out a deliverable, or receiving positive feedback from internal or external colleagues. These accomplishments cement your development and provide milestones to quantify all that you have worked for.

PERSONAL DEVELOPMENT

Recognising personal development over time and obtaining opportunities to develop further may well motivate you as a lead. This development could be in any of the areas already discussed or something else. Your technical programming knowledge and grasp of CDISC or industry standards may progress. Your ability to plan and manage a project may improve. Your skills in communicating with a variety of people for a variety of purposes may expand. Unfortunately, often your development will be through lessons learnt after problems you've faced or issues you've inadvertently created, but all these contribute to your future studies' success so they are not without value. Furthermore, gaining familiarity with the nuances of a particular trial and getting to know better how to fix and anticipate problems sooner is both rewarding and makes your life as a lead easier.

TEAM SUCCESS

It can also be highly fulfilling to build up relationships with other team members. There may be programmers on your team who you are able to learn from or those who are able to learn from you. It may be that as a study progresses you find you need to give less direction because your team has also grown more familiar with the details of the study and the processes to produce a deliverable. This is positive reinforcement of the combined effort of the programming team and study lead. Working with colleagues from other groups can also be satisfying as you gain a better understanding of each other's roles, are able to support them better and receive better support from them in return. This can also broaden your perspective the entire conduct of a trial.

THE BIGGER PICTURE

Seeing the end product, whether that is datasets, TFLs or other documents and recognising the work that went in and the quality that has come out, can help you realise that the hard work is worth it. But ultimately, in the industry we work in, the end product is safe, effective drugs and devices being brought to market and potentially saving or improving real people's lives. When your team is in the middle of a project and there is a lot of work to do it and pressure to do it well and efficiently, it can be difficult to remember that our work actually has a huge impact and could potentially make a difference to our own lives or the lives of our loved ones but this is a reality and a great motivation to strive ahead.

ACKNOWLEDGMENTS

Thanks to Lindsey Thomas and Clare Jones for their management and support throughout my own transition from programming team member to programming team lead and also for reviewing the abstract and paper.

PhUSE 2015

Thanks also to Edward Jones and Marion Thomas for proof-reading the paper and sharing ideas.

CONCLUSION

The lead role can be very challenging but also rewarding, in a variety of different areas. Depending on your previous experience and personal strengths, you may find that certain aspects of leading studies come more naturally than others. It is important to accept that most study leads will not master all elements straight away and it is usual to build up competence over time. Therefore, remember that one of the greatest resources for a new study lead is the expertise and experience of more senior programmers within your company. It is invaluable to draw from their working knowledge by asking questions, seeking out constructive criticism and then acting upon this advice or guidance. Some companies may also have built-in infrastructure to support new leads; internal or external training courses as well as mentoring schemes may be available or sometimes a new lead has the opportunity to start taking on more responsibility whilst working alongside a more experienced lead on a study.

Whilst bearing in mind that you are unlikely to master all aspects of leading initially, it is still important to keep quality a priority. Good communication and project management can help your study to run more smoothly, but accurate data tabulation and analysis will be the lasting product of any study, and could affect regulatory authority approval. Therefore, as a new study lead it is best to focus on the quality of your outputs and deliverables. The lead tasks which most affect quality are specification writing/review, and programming/review of datasets and TFLs. If you are not personally involved in much of the actual programming, then you can still pursue high quality by ensuring that the team is fully aware of expectations and standards, and regularly checking that these are being implemented in the outputs.

To summarise, there are many highs, lows and lessons to be learnt as first time study lead. Some of the highlights may include achieving high quality deliverables, building up good relationships with your team and developing your own skillset. There may be many more highpoints, but they will be dependent upon each lead's own challenges, strengths and the teams or clients they work alongside. It is important to celebrate all of these things and build upon them in order to consolidate a lead's learning to date, because inevitably there will also be low points. Lows may vary from communication difficulties, discovering mistakes (both from within and outside the programming team) and the sheer steepness of the learning curve from being a team member to team lead. These may be hard to handle and as a lead you often take a lot of the responsibility upon yourself, but it is important to use them as an incentive to learn and to develop as a lead. Finally we consider the lessons learnt, again these will be extremely varied depending on the lead and the specific study. They may be small things, which could still make a big difference to the study, they may be things which only affect the programming lead or they may affect the wider study team. They may be lessons in programming techniques, industry standards, processes, organisation or communication to name just a few. The important thing is to recognise each challenge as a lesson worth learning and to remember that it is worth learning because well conducted clinical trials can have an immensely positive impact on global healthcare.

CONTACT INFORMATION

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

Bethan Thomas
PRA Health Sciences
Llys Tawe
Kings Road
Swansea
SA1 8NE
Work Phone: +44 (0)1792 525710
Fax: +44 (0)1792525601
Email: thomasbethan@prahs.com

Brand and product names are trademarks of their respective companies.