

Industry Standard Good Programming Practice for Clinical Trials (Using SAS)

Mark Foxwell, AstraZeneca R&D Charnwood,
Loughborough, UK

ABSTRACT

Good Programming Practices help programmers write code that is clear, easily maintained and reusable. This results in improved efficiency in both the development and running of code as well as reducing the risk of making errors. Over the last year, an editorial board drawn from members of PharmaSUG, PhUSE and PSI have been driving and coordinating contributions to a document of Good Programming Practice for Clinical Trials on the sasCommunity wiki. It is hoped that this will provide a useful resource for both new and more experienced SAS[®] programmers and at the same time provide a location to encourage sharing of ideas on this topic. The aim is that this document will be recognized by the pharmaceutical industry, Clinical Research and Healthcare Organisations as well as Regulatory Authorities. This presentation will look at our progress to date in pulling together a useable document and at how this will be maintained in future. We would also like to share and discuss some of the more interesting proposals for good programming practice or those where ideas differ between contributors!

INTRODUCTION

Guidance and tips for good programming practice or coding standards are used extensively within programming including SAS programs for clinical trials. The benefits of using such standards include.

- Clear code that is easy to understand, review and validate.
- Flexible code that can be easily modified and reused in different environments
- Robust (defensive or data driven code) that requires little or no maintenance
- Code that runs efficiently on execution

Many programmers have their own collection of practices picked up over time. These may come from courses on efficient coding, conference papers on coding tips or conventions or even perhaps mailing list discussions. Many of us are also using guidelines recommended or mandated by our place of work.

This paper describes an initiative to take this one stage further, to develop a consensus document of good programming practice across clinical trials programming. It looks at the rationale for developing such a document and how it might be used as well as the challenges in developing a useful and objective document.

It is hoped that such a document can build on existing guidelines by collecting ideas and promoting discussion between programmers across clinical research, resulting in recommendations that are objective and tested.

WHY HAVE A GOOD PROGRAMMING PRACTICE DOCUMENT?

Use of personal collection of ideas or of company guidelines is a good start, but there is always the risk of guidance becoming out of date. In addition, although many practices are based on long experience and use throughout the industry; others may be more subjective or (arguably) inappropriate for some situations. As an example, company guidelines written, say for SAS version 8 on PC, may be totally inadequate when collaboration with another organization using SAS version 9 on Unix is considered. In another example [1], it has been pointed out that not explicitly stating the input statement in a proc statement (data=) allows that proc to be used without change for other datasets with different names. This is good advice if you want flexibility in a program, but others would quite rightly point out that it leads to a loss of control and risk of error. (Perhaps the flexibility would be better built in through a macro for example.) Macros themselves can make code difficult to debug and maintain for later users, but when well documented and supported could give very easily modified and flexible code. Even when practices are clearly of benefit to the program being written; the relative benefit has to be considered. You could see that spending a couple

PhUSE 2009

of hours maximizing a program for reporting a small pharmacology study is likely to be time poorly spent compared to time documenting and building in design features that will allow the program to be reused on the next study with similar design, whereas these would be much more important for a program reporting laboratory safety results in a very large phase 3 trial.

In all of these cases, it would be beneficial to have access to a set of good programming practices that draw on as many ideas as possible, that have extensive peer review and that are supported with good rationale for how and why they should be used.

Once such a document is in place, one could see several advantages in addition to an up to date and comprehensive set of practices to refer to. Consistency in style will help in sharing of code between organizations, and be of use in supporting programs such as OS3A [2] an initiative to develop open source code for clinical trial coding. Such a document would also be a good basis for writing company guidelines or for those new to SAS programming for clinical trials. Perhaps, it could also form the basis of a general standard for working with regulators in future.

WHAT SHOULD A GPP IN CLINICAL TRIALS DOCUMENT CONTAIN?

Describing the content of the GPP wiki in detail is not an aim of this paper, but I would like to highlight some key issues regarding the content and structure of the document.

CLINICAL TRIALS

Of key importance is deciding whether the focus of the document should be clinical trials and then deciding exactly what this means. I think that a focus on clinical trials is appropriate, but I think that a clear definition of what this means is still being defined. Much of the good programming practice used by clinical SAS programmers is applicable to SAS or even programming generally. However I do think that the nature of clinical trials has some influence on the way SAS is used. Clinical trials could be considered to contain relatively complex data structures and small datasets. Programs are often run only once or no more than a couple of times once validated. In these circumstances, practices that reduce coding time are likely to be of more use than those that reduce execution time. Programs are also likely to be reused on studies with similar (but not necessarily identical) design, so code that can be modified quickly, easily and without error will be at a premium.

Regulations and data analyses specific to Clinical trials may also be important. For example, practices related to 21CFR11 or CDISC data standard will be of interest. There might also be a need to include practices related to the handling of MedDRA, RECIST or SF36 data to name but a few. One example of 21CFR11 compliance would be the avoidance of 'hardcoding'[3]- changing of data in a SAS program rather than the (audit trailed) database.

Another consideration might be the background of Clinical SAS Programmers. Many programmers come from backgrounds other than a computing degree such as statistics or life sciences. Over time, I suspect that this difference does not matter. If people in a SAS programming role are interested in developing programming skills they will pick them up. However for a proportion of people SAS programming may not be their primary interest and they may just need enough SAS to 'get an answer'. I think that any discussion of good programming practice has to provide something useful for such occasional programmers as well.

STRUCTURE

For me a key reason to keep a focus on clinical trials relates to keeping the document to manageable size and complexity. For such a document to be of use, a programmer needs to find what they want and also to get a good idea of the content of the document very quickly. On that basis one might need to consider what to exclude if the documents looks to be getting too large.

As an example, general advice such as to use 'where' to filter data being read in to a dataset rather than 'if' to filter data at a later stage might be of use to novice and occasional programmers; but as it is a tip with very general applicability to programming, it may be better in a separate linked document or appendix.

EDITING AND MAINTAINING THE DOCUMENT

SASCOMMUNITY WIKI

The Good Programming Practice Document has been set up as a wiki as part of a general SAS collaboration web site called SasCommunity.org. Most of you are probably familiar with the concept at least in using documents such as Wikipedia for reference, but may not have been involved in editing one. Each wiki page consists of the web page itself and also a discussion tab, a separate page that contains discussions and comments about the main page. Also available is a page containing the raw source code (The wiki uses a basic mark up language – for example for emphasizing text or adding hyperlinks), and a page that shows the edit history- allowing comparisons of text before and after each edit.

In order to edit the document, all you need to do is register with the sascommunity.org web site. Once this is done, you can edit the source code page directly. In addition, extra management functions are provided, you can track your

PhUSE 2009

contributions for example and very usefully; can put pages on a watchlist. This means that you get an email and link to the change history should anyone edit a page that you are interested in.

This format is ideal for authoring a collaborative document with extensive peer review as described above, and we encourage SAS programmers to add ideas for good programming practice and to include rationales as to why these are useful and in what context. You may also have ideas for the structure of the document or may feel that a contribution merits further discussion or scrutiny. In this case we would also encourage contributions to the discussions tab.

We have tried to encourage contributions already by publicizing the site through professional organizations such as PhUSE, PharmaSUG, PSI and ACDM earlier this year, and it is our intention that this is a document written by SAS programmers for SAS programmers, not the editorial board (although we accept that we are SAS programmers ourselves and may have the odd idea to contribute!) This approach should allow sharing and discussion of the widest possible range of ideas, and should allow this with minimal reliance on the time of one or a small number of people. Hopefully as the number of contributions increases, so will readership and in turn further contributions.

EDITORIAL BOARD

Although the intention is that the document should 'write itself', a little guidance and input is likely to be needed. The editorial board exists to provide this. The editorial board mission is currently defined (End August 2009) as

- Review the contributions periodically
- Present the guideline to conferences
- Submit to the boards of PharmaSUG and PhUSE for endorsement and release of a stable version.
- Identify requests for contributions.
- Propose solutions whenever conflicting opinions exist between contributors.
- Provide guidance for contributors to the OS3A program.

Additional role for the editorial board may be to help structure the document over the next few months and maintain structure thereafter. This may help contributors identify areas where content is lacking as well as give a framework for adding their own ideas.

I could also see a need to help make decisions or stimulate discussions on relevancy as the document increases in size and complexity.

CONCLUSION

Good Programming Practices help programmers develop code that is efficient and easy to maintain and such practices are well known and used by Clinical trial SAS programmers. The Good programming practice in clinical trials document on sasCommunity.org provides a way of developing an industry wide collection of the best and most relevant practices. The collaborative nature of developing documents in a wiki allows scrutiny, review and discussion of those ideas to provide something that is objective and up to date. An editorial board is in place to steer development of this document, but its success will also depend on active contribution from throughout the Clinical Trials SAS community.

REFERENCES

1. Aster, Rick (1998). "Coding for Posterity." In the 11th Annual Proceedings of the NorthEast SAS Users Group. <http://www.nesug.org/Proceedings/nesug98/appl/p131.pdf>
2. Open source SAS software applications (OS3A) <http://os3a.sourceforge.net/>
3. Fehrer, Susan, M. I don't look good in Orange or stripes subtitled "Hard coding is not permissible" Pharmasug 2000

The wiki itself can be found at Good Programming Practice for Clinical Trials
http://www.sascommunity.org/wiki/Good_Programming_Practice_for_Clinical_Trials

ACKNOWLEDGMENTS

I would like to thank the GPP editorial board for input into discussions, answering my questions and review of initial drafts. I would also like to mention contributors to the wiki who have already provoked a lot of thought and discussion around the content and direction of the document.

CONTACT INFORMATION

PhUSE 2009

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

Mark Foxwell
AstraZeneca R&D Charnwood
Bakewell Road
Loughborough LE11 5RH
Email: mark.foxwell@astrazeneca.com

Brand and product names are trademarks of their respective companies.