

Implementation & Maintenance of a Reporting System

Lee Seymour, GlaxoSmithKline, Harlow, UK

ABSTRACT

The GlaxoSmithKline merger in Dec 2000 highlighted the need for standardisation, which led to the development of a new in-house global standard reporting system called HARP (Harmonisation Analysis & Reporting Program).

This presentation covers the processes involved in maintaining a validated reporting system, including prioritisation of changes, backwards compatibility issues, and impact on the end users.

The presentation also covers the processes of development and testing of tools updates.

BACKGROUND

The merger between SmithKline Beecham and GlaxoWellcome in December 2000 highlighted the need for Standardisation across Biostatistics, and HARP is one of the outcomes of this standardisation. The first phase of HARP was to define the GSK Analysis & Reporting processes from start to finish. The second phase was to build the tools required help carry out these processes.

We currently have 4 major sites using HARP, two in the UK and two in the US. Our satellite companies in India, Verona and Canada are also using HARP. Eventually we will have approximately 400 users worldwide.

Within the legacy reporting environment each site was generally doing the same things but in slightly different ways. We had limited sharing of programming code, mainly due to the lack of consistency, and a lot duplication of effort because every team had their own code. We did have pockets of efficiency as most teams had their own set of macros to use. We also faced a considerable 'start up time' for newly assigned staff, which meant that it was very difficult to share projects across sites.

So why did GSK feel they need to create a Standard Reporting System? Well, Core standards were being developed for data capture (paper & electronic), data basing, right the way through to data displays (Tables, Figures, Listings). Therefore it made sense to take the process one step further and develop Standard Reporting System to generate these standard data displays. We also needed to get drugs to the market quicker, which meant we need to report studies faster, but with the same amount of resource. We wanted to improve security inline with GCP compliance, and have a way of making sure people were adhering to the new standards. A standard reporting system would enable use to achieve all these things.

INTRODUCTION

The HARP Reporting tools are a cohesive collection of SAS macros, designed to integrate with the HARP Application that make up a standard, global reporting system.

The suite of macros follow a “building” block design:-

Utility macros - Code blocks that are designed to perform one job and do it well.

Package macros - A macro that contains a series of calls to utilities macros to perform a specific task (e.g. frequency counts, summary statistics)

Wrapper macros - Contain no coding. Pass default macro parameter values into the different package macros to create a specific display. (E.g. The standard displays for summary of adverse events and summary of concomitant medications will both use the same package macro).

The reporting tools are designed with a specific purpose; however enhancements can be made to improve the overall functionality of the individual tools or to remove any bugs.

This paper discusses the processes involved in maintaining and updating a validated set of reporting tools.

WHAT CAN CHANGE AND WHY

Since the original release of the reporting system several updates have been made to the reporting system, these changes can consist of:

Existing tools can be updated to fix bugs and also to enhance them to improve functionality based on business need. Each of the standard tools are designed for a specific purpose, for example, a standard macro will produce the standard adverse event summary, however enhancements to this macro may be made to this macro if the business require a variation of this table.

Code may also be updated to improve the overall code efficiency.

Code may be updated to resolve bugs that may have been noted.

New tools requested are considered a change to the reporting system. Since the first release new tools have been released to report Pharmacokinetic data, tools for producing PK graphics and for summarising in-stream safety data.

The overall reporting system is designed to produce displays based on pre-defined data standards. Tools may be updated if the data standards change.

HOW DO CHANGES GET IMPLEMENTED

The majority of updates to existing tools are raised by the users, but changes to the tools can also be raised by the tool development and tool support groups. Users have the functionality to raise specific changes through a centralized support database.

ASSESSING THE CHANGE: THE DISCUSSION

All proposed updates or additions to the reporting system are initially assessed by a team comprising of representatives from the user community, tool development and tool support.

This team (The HARP Reporting Tools Change Request Team) is responsible for assessing, prioritizing and submitting change requests.

Each team member represents their group and is responsible for:

- Submitting change requests to the Change Request Team for review
- Completing the change request forms for all change requests originating from their site
- Acting as the business contact for the developers processing the change request
- Keeping the originator of the change request informed of the progress of the change request
- Point contact for the group on any Reporting Tools change request issues/questions

As proposed changes are raised, the team is responsible for assessing the proposed changes. E.g. is the change a suitable one, what is the business value of the proposed change and what would be the cost of implementation.

The final stage is to review the proposed changes, assessing the business value versus implementation costs to assess which changes should take higher priority over others.

A low priority change would commonly have a high implementation cost (for example it may require a considerable amount of code changes) but has minimal business value (e.g. the update is only required for a small subset of studies) .

A high priority change would be regarded as one which would benefit the majority of users. High priority requests may require minor or major code updates.

SUBMITTING THE CHANGE

Once the proposed changes have been agreed and prioritized, the formal change request document can be created. This document states what is required to be updated (macro name and version), reasons for the change to be implemented and the business need for the change. These documents are the start of the formal documentation required to ensure the reporting tools are maintained as a fully validated system.

The change request documents are submitted to the tool development group.

ASSESSING THE CHANGE: THE INVESTIGATION

The tool development group will assess the change with the following aim:

To see if any other macros require updates. On investigation some changes may require updates to utility macros not specified in the original request.

To estimate the time required for development and testing. This may be useful for estimating required resources but also for estimating a release date for the updated tool(s).

To assess if document updates may be required. For example will the change require an update to the macro specification document? (The Unit Specification), or to the user documentation?

The result of the tool development assessment is documented in the change request form. At this point the change request document is formally reviewed and the assessment of change signed off and code and supporting documentation can be updated.

UPDATING THE UNIT SPECIFICATION,

The unit specification document contains information on the purpose and design of the particular macro. If a tool enhancement results in new parameters being added, these require adding to the unit specification.

UPDATING THE CODE

Updating the code you would think be the easiest part of the process, but very careful consideration needs to be made when adding or amending code. You need to ensure that the changes made do not have an effect on backwards compatibility. As the code is updated the modification history is updated so that in the macro itself it's clearly identified what changes have happened since the original tool was released into the production environment.

UPDATING SYSTEM TEST CASES

The system test cases may be updated while the code is updated. These test documents are an important part of ensuring the system remains validated.

SOURCE CODE REVIEW

The updated code is reviewed according to a predefined checklist ensuring that the code is adhering to the programming standards.

SYSTEM TESTING

Testing performed by the developers in order to test that the product is doing what it was designed to. Test failure here would mean returning to the code update stage and performing a second round of source code review and system testing.

USER ACCEPTANCE TESTING

Once system testing is successfully complete then User Acceptance Testing (UAT) can commence.

UAT is performed by the product users in order that the end users are satisfied that the changes/updates are appropriate and work as designed. A test failure at these stage could result in returning to the code update stage a redoing the testing process.

After all the change assessment, updating the macros and documentation you would think that it's now time to release, but there are a couple of other things to consider.

UPDATED DOCUMENTATION/TRAINING

It's all well and good releasing improved macros to the production environment, but it wouldn't be worthwhile if the users themselves did not know about the updates. Does the change require updates to the user documentation? Is additional training required?

COMMUNICATION

Communicating the upcoming changes to the users in advance will give them the opportunity to see in advance what updates are being made. Communicating the proposed release date will give them the chance to suggest alternate dates if the one proposed clashes with an important deadline

RELEASING THE MACROS,

After all the stages above have been completed the updated tools can finally be released into production to the delight of the end users. It's then time to work on the next batch of changes.....

CONCLUSION

As documented above it can be a lengthy process to maintain a standardised reporting system. But even if it is a minor change to a small piece of code it's imperative to follow the steps to ensure the validated status of the system.

Keep the end users in mind! You may come up with a neat yet complex series of macros, but if the users do not understand how to use them then there is little benefit.

CONTACT INFORMATION

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

Lee Seymour
GlaxoSmithKline
Research & Development
New Frontier Science Park
Third Avenue
Harlow
CM19 5AW
Work Phone: +44(0)1279 646010
Fax: +44(0)1279 644430
Email: Lee.Seymour@gsk.com
Web: www.biometrics.com