

### **Close cooperation between developers and end-users is vital to deployment of CDISC standard software**

Donald Benoot, SGS Life Sciences Services, Mechelen, Belgium

#### **ABSTRACT**

At SGS Life Sciences Services we have been implementing several homemade software products within our organization over the past several years, with the specific aim to provide our customers with a CDISC compliant end-to-end solution. In this paper the different development, deployment, and training steps as well as the important role of clear communication and close cooperation between the developer and end user will be discussed, using real life case studies. More specifically we will focus on the following topics:

- i. Staggered deployment to highly motivated early adopters
- ii. Training users
- iii. User interface (UI) and user experience (UX) influence on workflow
- iv. Empowering users
- v. Software performance
- vi. The origin of suboptimal design decisions and how to correct

Using these methodologies, we were able to successfully implement the software packages in production, identify bugs and shortcomings pro-actively, progressively plan upcoming release updates, and streamline the implementations from lessons learnt during previous cycles.

#### **INTRODUCTION**

At SGS Life Sciences Services we have been implementing several homemade software products within our organization over the past several years, with the specific aim to provide our customers with a CDISC compliant end-to-end solution. As an additional benefit to implementation, SGS has been able to increase efficiency by reducing overhead, and unifying the workflows within our company.

Two important Data Management software packages in our CDISC standard compliant environment are the Data Quality Control (QC) tool and All-2-One (A2O).

- The QC tool is used to perform the quality control on the conversion of incoming source data from (e)CRF, esource, external vendor data,... to CDISC SDTM format, as well as logging and following up on any encountered issues
- A2O is used at the trial level to setup the study rule package (the collection of checks and listings used to clean the database), create new rules, perform the data cleaning, and log and follow-up on review issues. It is also used to maintain a library of validated rules, which can be used in new trials, and a library of SDTM checks, which are run on all trials.

Developing and implementing a new software package is a huge undertaking. Even though the function of the software package was thoroughly investigated, and the findings were listed in the user requirements specifications and the design specifications, taking all the end user interactions and potential usages of the software into account during the planning and development phase is very challenging. Even more daunting is putting the new software package into production, especially when the proverbial show has to go on.

At SGS, the application development is done by the Application Developers (AD) team and the Clinical Data Programmers (CDP). During the conduct of a clinical trial, the software packages are primarily used by:

## PhUSE 2015

- The CDPs, providing study specific programming support (conversion, rule programming) and CDISC SDTM standardization
- Clinical Data Managers (CDM), for validating the CDISC compliant database, defining checks and listings, building the trial specific rule package, and cleaning the data.

As the software is developed and used by these different groups of people, the feedback from all these different levels was - and still is - crucial for the development, deployment and the management/evolution of these software packages.

Using methodologies such as staggered deployment and a tight feedback between the ADs, the CDPs and the CDMs, we were able to successfully implement these software packages in production, identify bugs and shortcomings pro-actively, progressively plan upcoming release updates, and streamline the implementations from lessons learnt during previous cycles.

In this paper several case studies will be discussed touching on important lessons learned, including staggered deployment, UI optimization, correcting suboptimal design decisions, and the importance of speed.

The primary goal of this paper to explain the different development, deployment, and training steps and showing the role of the developer and end user cooperation in this process.

### STAGGERED DEPLOYMENT USING HIGHLY MOTIVATED EARLY ADOPTERS

A daunting task in the implementation of new software is the first deployment into the live environment. For this part of the paper, A2O will be used as an example.

#### FIRST DEPLOYMENT OF A2O

When we initially prepared for the first deployment of A2O in our workflow, we selected a smaller scale phase I trial as a pilot study. As we were working on an actual, live trial, it was very important to make sure work could continue should an issue arise. Therefore, the trial was set up in parallel in A2O, as well as in our previous system as a back-up solution. As an additional precaution, we specified go/no-go points at several steps during the setup of the trial. At each of these points, an evaluation was performed ascertaining whether or not it was still feasible to proceed in the new system.

Of course, running a trial in parallel in two different systems requires extra resources. Apart from extra data management personnel, the application developers were available at all times during the first deployment for help, maintenance and explanation, should issues occur.

The decisive point in the trial came at the start of programming the rules (electronic checks and listings) in A2O. One of the advantages of this system is the ability to store rules in a library, rendering it unnecessary to reprogram and re-validate a rule on every trial. In the old way of working, all rules were created uniquely for each individual trial. However, the rule creation in A2O diverges very much from the old way of working:

- Library-centric rule development: the idea behind the creation of a rules library sped up the trial setup by having a set of validated checks and listings readily available. The rules are created at the trial level, but with the specific aim to add them to a library to use them in as many trials as possible, thus limiting the amount of trial-specific rules that have to be created in the future. To keep the number of rules manageable, strict rule naming, description and programming conventions were created which have to be adhered to. All new rules have to be compliant with these conventions in case they are selected for inclusion in the library.
- The programming of rules happens in A2O itself
- The validation of rules happens in A2O and is therefore linked to the usage of that system

As a result of this library, the decision to start the rule creation in A2O removed the fallback solution of using our back-up trial in the previous system. Although the system displayed some minor problems and issues, and the CDMs processing the trial had to learn and adapt to the new system, it did work and the trial concluded successfully.

#### LESSONS LEARNED

After the successful use of A2O in the first few trials, we scheduled a series of lessons learned meetings to discuss these early problems and optimize the workflow. There were some very important lessons:

- SOP: some of our Standard Operating Procedures were not in line with the new way of working. Planned SOP deviations were created as a temporary solution, but the SOPs were scheduled immediately for revision
- Library: the initial creation of the library was a huge undertaking, but the effort can be reused. This has been proven in later trials, where checks and listings creation is drastically reduced. However, in future trials, we made sure extra personnel was available for all trials which did not have a library (or only a library with a limited amount of rules) available.

## PhUSE 2015

- The availability of dedicated developers during the deployment of the software was crucial. A few issues could have been showstoppers if the developers were not on hand

### GRADUALLY EMBEDDING THE NEW WORKFLOW

For further trials, we gradually reduced the amount of backup plans, go/no-go points and the reserve personnel. We also started to gradually transfer privileges for various interactions with the database, first from the ADs to the CDPs, and then to the CDMs. When first introducing the new software, the need to make sure that everything was faultless in the database before going to a next step was the most important. However, this is a burden on the developers and slows down the work of the end users. As the workflow became well known and the necessary automatic precautions were put in place, we gradually shifted these actions to the end users, speeding up the workflow and relieving the developers from this burden.

An example of this move of privileges is running the checks and listings (rules) in A2O. At first, this process could only be started by the AD. This way, the AD could check whether or not the database was ready to run the checks and listings. This created an extra safety net for the database consistency, but it did hamper CDM work. This was clearly a point for optimization, both for the AD, who could do this for one or two trials but was facing the insurmountable task of having to process all of our trials running in A2O, and the CDM, who could then plan his own working day without the need for AD intervention.

At the moment, all of our trials are running in A2O. The program continues to evolve, now making both easier to setup and review trials, and to automatically create reports and data review guidelines. The continued discussion between developers and end users, during the ongoing development on existing modules and the addition of new modules, has really made this software package an ideal fit for our company.

### TRAINING USERS

Training and documentation are important aspects in the implementation of new software in a live environment. During the first implementation of the QC Tool, we, once again, used a small team of motivated early adopters to use the new system and very critically return comments to the developers.

### FIRST IMPLEMENTATION

Our existing system for performing quality control on the conversion of the clinical database had some important drawbacks. It was time consuming, predominantly paper-based manual work. During an internal audit, focused on streamlining our company's processes, the QC on the data conversion was quickly pinpointed by both the CDPs and the CDMs as one of the most time consuming steps in setting up a clinical trial. The QC tool was therefore developed to eliminate as much of the overhead as possible, make the QC completely paperless and to guide the CDMs through the QC of conversion process in a structured and unified way, and create an environment in which the CDP could start correcting issues while the CDM was still performing the quality control (figure 1).

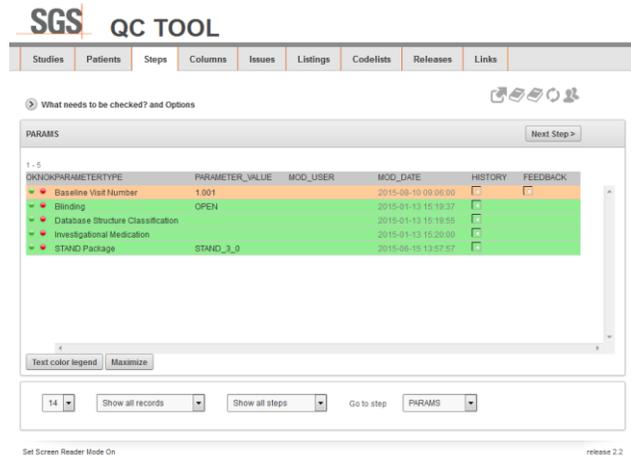


Fig.1 : Going from a purely paper-based QC of the conversion to a fully automated QC using the homemade software QC Tool.

No additional resources were necessary during the first implementation of the QC tool, as the fallback in case of software problems was returning to the paper-based system. The necessary documentation could be printed at any time. Therefore the first few implementations of the QC tool in live trials were done in parallel when new trials became available for QC and the lead CDM of the trial was willing to be part of the early implementation of the QC tool.

### MONTHLY EXCHANGE MEETINGS AND DEVELOPING EARLY TRAINING DOCUMENTATION

Together with the implementation of the QC tool, we scheduled monthly exchange meetings with all involved

## PhUSE 2015

stakeholders; in this case the CDMs and the ADs. During these meetings, all findings, questions and workflow issues were discussed between the users and the developers. Application updates were agreed upon and evaluated during the next meeting. There was a very quick turnover time between the recognition of issues in the early period of this tool and the resolution thereof; driving a quick evolution in the maturation and stabilization of the QC tool.

The downside in the number of trials running with the QC tool in the early deployment phase, the monthly exchange meetings and update scheme was the difficulty to properly document and train on the tool in the early stages of its development due to all the changes. Therefore, we created a 'living' presentation, which was updated, together with the tool itself, after every exchange meeting. The early adopters provided training to the new users within their team and provided the link to the living presentation, as self-training could be possible if the tool changed between usages.

### STABLE SOFTWARE AND CREATING TRAINING AND DOCUMENTATION

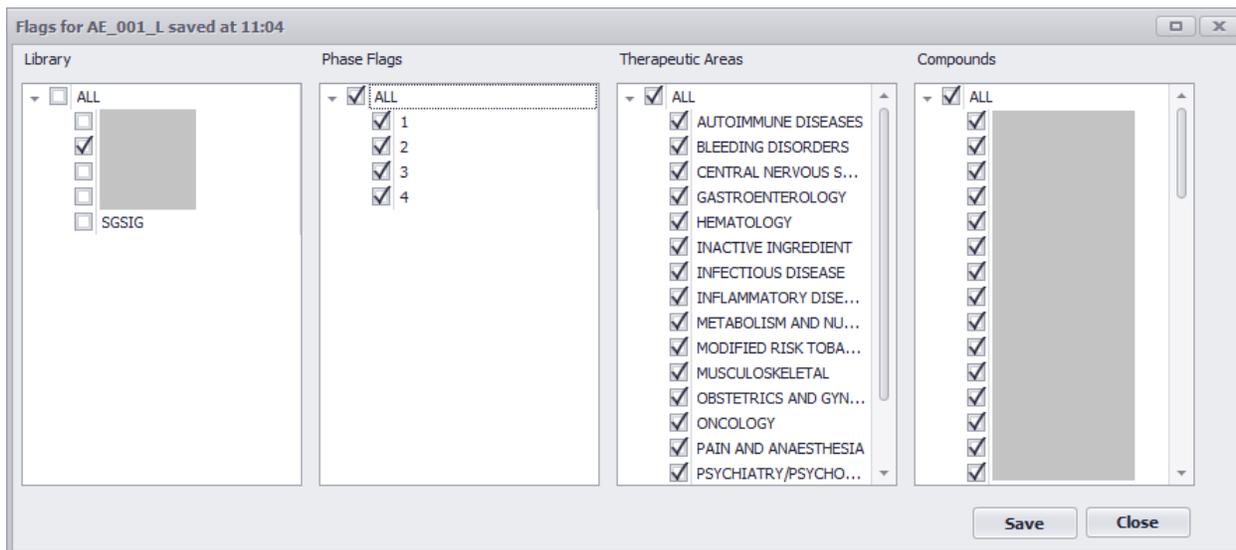
After the initial development cycles, the development stabilized to a more traditional pace. At the moment, the QC tool is in version 2.2 and stable enough to create training documentation. All of the experience and documentation was forwarded to our dedicated training coordinator and an automated self-training module is being developed now.

### UI AND UX INFLUENCE ON WORKFLOW

Although the backend of a tool can work flawlessly, the way everything is presented to the user can be less than optimal. These user interface shortcomings can cause delays, confusion, and even irritation. Ironing out these imperfections is often relatively easy, but will be seen as a huge improvement for the end user.

### CLEAR FEEDBACK ON PERFORMED ACTIONS

Originally, when an action was concluded in A2O, a notification was added to the title bar (see figure 2). This was very often overlooked by the CDM using the tool, resulting in CDMs waiting for an action that was actually already done. As a result, we added a slidedown notification (see figure 3), protruding from the titlebar informing the user that an action is done. The notification disappears automatically after a few moments; so the user does not have to remove it manually (i.e. no extra interactions necessary). Because of the addition of this extra visual cue, the tool is generally perceived as running faster.



# PhUSE 2015

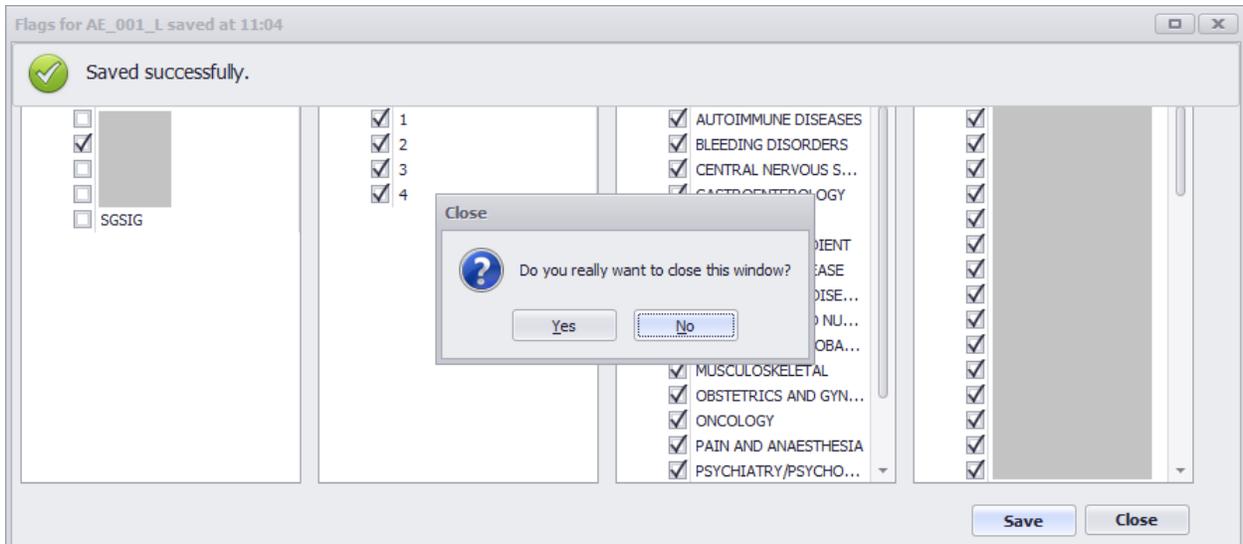


Fig.2&3 : In the original dialog, only the time was added to the titlebar. This was not always obvious. Therefore the slide down notification 'Saved Successfully' was added. Some of the records are censored (gray box).

## DECLUTTER

The original print dialog (see figure 4) for Document Clarification Forms (DCF; a query) forced the user to select the queries to print from dropdown boxes. This required the user to first create the selection, plan the necessary filters and then enter them in the dropdown dialog box. The dropdown box was quite confusing and caused a lot of erroneous printing, which of course had to be redone. This was not a trivial matter; because the DCF, once printed, gets a flag which makes it unprintable (i.e. this DCF is already sent to the sponsor and therefore should not be sent again). The resulting corrections caused a lot of work and frustration for both the developers and the end users.

The CDMs following up on the print dialog issue proposed a new UI for the print dialog. But during the meeting with the stakeholders, a more convenient method was proposed by the AD.

The solution to the incomprehensible print dialog was removing the print dialog altogether; the selection is now done on the DCF overview page itself and the selection of visible DCFs is what is printed. A very confusing step was removed and the overall experience was greatly improved. The input from the developers, once they saw how the end users actually used this module, really improved the ease of use.

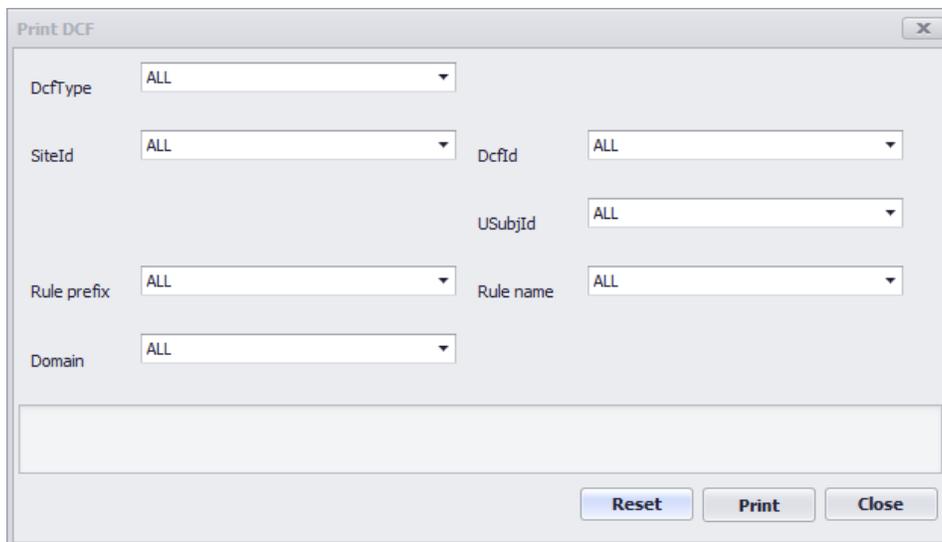


Fig.4 : The old print dialog with the difficult selection boxes.

## REMOVE UNNECESSARY STEPS

During the early development of A2O some cautionary dialog boxes were added to prevent the loss of data or user-specific selections. Examples are the "Are you sure you want to close this window? Yes/No" Dialog or the "Please

## PhUSE 2015

close all windows before action is possible” windows. The dialog boxes do indeed protect the user from performing unintentional actions, but can cause a lot of slowdown and some irritation as well.

In the example of the “Do you really want to close this window? Yes/No” dialog (see figure 5), the number of times the user did indeed mean to close the window is much higher than unintentional closings. With the speed improvements now present, the amount of work to reopen an unintentionally closed table and reset the selections is also less time-consuming. Therefore we are now evaluating removal of this particular dialog box. As this is an old box, the buttons also state “Yes” and “No”. While reviewing user interface now, we opt for actions, in this case, “Close window”/“Cancel”.

In the example of the “All windows need to be closed first” (see figure 5), the user has to perform several actions to continue working. First the user has to close the popup window. Then all windows have to be closed, and then finally the workflow can be resumed. Therefore we are now evaluating updating this to a dialog box with the options: “Close all windows” and “Cancel”.

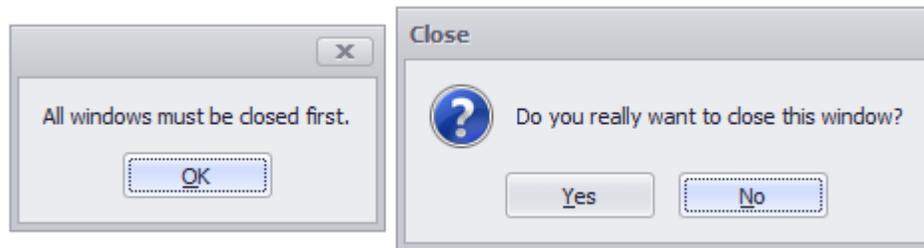


Fig.5 : The offending dialog boxes

### EMPOWERING USERS BY SHIFTING PERMISSIONS/PROVIDING NECESSARY TOOLS

An important impact on the user experience is the gradual delegation of permissions to the end users. As a software package matures, we try to create a continuous workflow for a certain function where the end user can perform all of the necessary actions from the beginning to the end, without having to make a request to the ADs or CDPs.

An example we have already touched upon is the running of the checks and listings by the CDM. Another example is loading of help tables. Some rules in A2O run generic code which is trial non-specific, but then subsequently use help tables to describe the trial-specific aspects. In the earliest incarnation, the help tables were loaded in the database by the CDPs. If an update was necessary, the CDM always had to make a request to the CDP to make the update and to load the new data in the database. Now, editing help tables is a module in A2O and can be done by the CDM, without outside intervention.

### SOFTWARE PERFORMANCE

#### OBJECTIVE SPEED INCREASE

When we first started using A2O in production, we preferred using smaller Phase 1 trials as previously explained above. However, when upscaling to more and bigger Phase 2 and 3 trials, the system started to run too slow. The datasets which are loaded are very large and can slow down the system when not handled efficiently. Various upgrades were introduced in rendering the data to screen, only calculating certain properties of a record (column width, row number, status colour,...) when a record is displayed on the screen. Furthermore, the efficiency of database connections was upgraded, performing a sequence of actions within one database connection.

#### PERCEIVED SPEED INCREASE

On top of the objective speed increase, we also strived to make the program work as fluid to work with as possible.

Wherever possible we provide visual cues when an operation is ongoing and done, limiting the amount of time a user is needlessly idling in front of the screen. We also tried to optimize the menu structure and make the screen as uncluttered as possible, making it possible to quickly find a module and focus on its most important tasks (see figure 6).

# PhUSE 2015

row	siteid	usubjid	armcd	aeterm	aeout	aestdtc	aeendc	aeerf	message	state
1				FATIGUE	UNKNOWN	2015-07-16T20:01		AFTER	Adverse event outcome = unknown, but an adverse event stop date or ongoing at the end of the ...	CHANGED
2				HYPONATREMIA	UNKNOWN	2015-08-13		AFTER	Adverse event outcome = unknown, but an adverse event stop date or ongoing at the end of the ...	NONE
3				CHILL AND COUGH	UNKNOWN	2015-08-07T08:01		AFTER	Adverse event outcome = unknown, but an adverse event stop date or ongoing at the end of the ...	CREATED IN EDC
4				NUMBNESS ON THE RIGHT HAND	UNKNOWN	2015-08-09T19:00		AFTER	Adverse event outcome = unknown, but an adverse event stop date or ongoing at the end of the ...	NONE

Fig.6 : The rule output is presented as a clear table.

A batch processing dialog was added (see figure 7), making it possible to perform the same review action on a group of filtered records.

Set message, state and comment for all selected records

Message, State and Comment for all selected (filtered) records will be updated as specified below, by clicking the Apply-button.  
IMPORTANT: When no filter is applied all records will be updated.

Message:

State:

Comments:

Clear Message

Clear State

Clear Comments

Apply Close

Fig.7 : The batch processing dialog.

## SUBOPTIMAL DESIGN DECISION

### A2O LIBRARIES

Part of the optimization of our internal workflow was the creation of a library for our checks and listings. The idea behind the creation of a checks and listings library was, of course, speeding up the trial setup by having a set of validated checks and listings readily available. Although the trials we process are all SDTM compliant, several of the sponsors we work for have created their own implementation. Because of these sponsor-specific implementation guides, there exist some subtle differences in database setup due. To handle these, we created different libraries for these different implementation guides. Later on however, this way of working proved to be inefficient.

### COPING WITH THE SUBOPTIMAL DESIGN

At first, working with different libraries per SDTM implementation guide worked very well, right up until we started to get new implementation guides, or higher version of sponsor implementation guides. At that moment, we had to start recreating a library from scratch, rendering all the previous work worthless. To bypass this, we documented a plan to copy checks and listings from an existing library to the newly created one.

Copying the checks and listings from one library to the other was a good workable solution for the short-term, but it had a few downsides:

- Labour intensive: the correct checks had to be selected, copied and tested. Copying could only be done by ADs, taking up a lot of their time.
- Error prone: a lot of human interactions (selecting, copying)
- Not futureproof: has to be redone for every new implementation guide/version
- Dependent on name availability in copy-to library: if a check or listing with the same name already exists, a copy cannot be made and the rule had to be reprogrammed and revalidated

## PhUSE 2015

- Loss of version control: if an update is performed on the original check/listing, the copied check/listing is not updated

### CORRECTING THE SUBOPTIMAL DESIGN

To correct for this suboptimal design, we merged all existing rules in one master library, but we added several metadata flags to the systems (see figure 8). Now the creation of a sub-library is only the work of adding the correct flags to the checks and listings. Furthermore, we have even more control over the specificity of a rule:

- They can be part of multiple sub-libraries
- A rule can be part of a sub-library, but only available for specific compounds or specific clinical phases
- Version control is maintained throughout the library

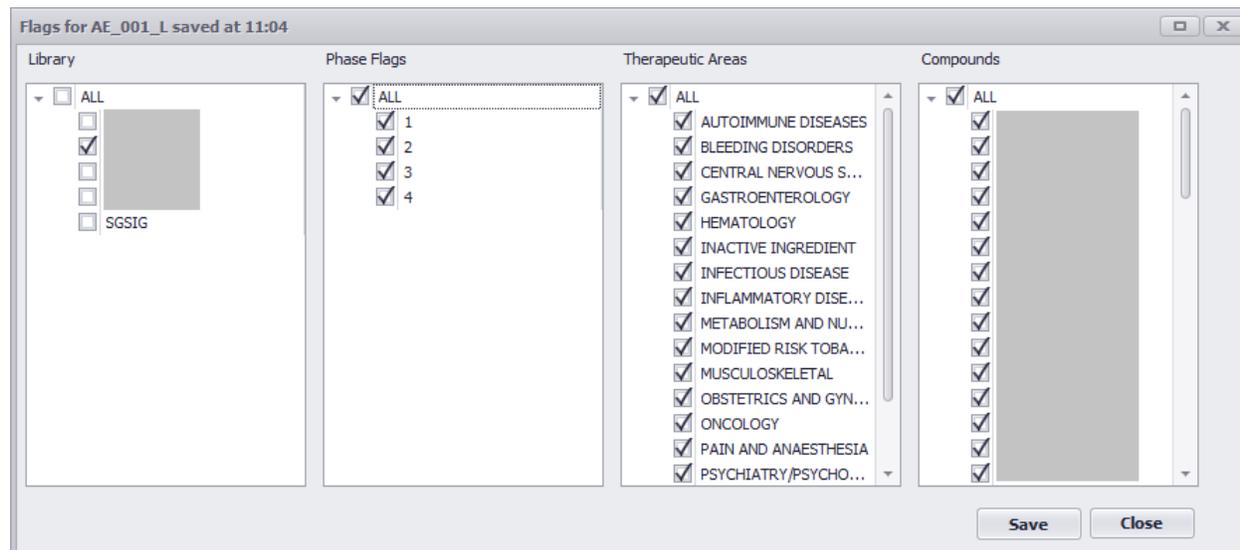


Fig.8 : The library metadata flags. Several levels of metadata flags are present: library (implementation guide), clinical phase, therapeutic area and compound. Some of the records are censored (gray box).

Although the merging of the libraries itself was a huge undertaking, we certainly ended up with a system that is much more versatile, efficient and future proof. The maintenance of the library and the creation of sub-libraries is mostly done by CDMs, freeing up the time of the ADs and CDPs.

### CONCLUSION

The close interaction between the ADs, the CDPs and the CDMs within our company has made the process of designing, developing, maintaining, and evolving our homemade software packages easier and a lot less frustrating for both developers and end users. In particular, for the software that our CDMs depend on daily and spend quite a lot of their time using, it is a refreshing experience to actually be able to work together with the developers in creating the best possible software, which not only works well, but also works with the end users in easing the actual workflow. The feedback loop from the end users to the developers is not just a one-way 'upgrade this' command, but the developers are actively involved in looking at the workflow of the users, and often come up with even more clever, or fundamental, solutions to the initial problems.

### CONTACT INFORMATION

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

Donald Benoot  
SGS Life Sciences Services  
Generaal de Wittelaan 19A  
B-2800 Mechelen  
Work Phone: +32(0)15 283 057  
Fax: +32(0)15 29 93 02  
Email: donald.benoot@sgs.com  
Web:

Brand and product names are trademarks of their respective companies.