PharmaSUG 2015 - Paper PO05

A Web-Based Approach to Fighting Analysis Chaos

Gordon Fancher, Seattle Genetics Inc., Bothell, WA
Rajeev Karanam, Seattle Genetics Inc., Bothell, WA
Shawn Hopkins, Seattle Genetics Inc., Bothell, WA

ABSTRACT

Review of analyses by cross-functional teams is essential to provide quality output for industry conferences, manuscripts, regulatory submissions, and internal decision-making. Documentation and tracking of reviews are important to ensure all comments are addressed. In addition to comments, documentation typically includes records of the items being produced, team members performing the work, tasks assigned and performed by each team member, the project timeline and progress relative to the timeline, and specifications and metadata pertaining to each item being produced. Datasets, tables, listings, figures, and SAS® macros and other utilities are items that have the potential to be tracked. There is a need to track these analyses over time for multiple purposes. Such purposes include determining common items used to prioritize development of infrastructure, developing standards for timelines and metrics for evaluating workload and resourcing, and analyzing performance on both team and individual team member levels. Both documentation and tracking can be complicated and time consuming, particularly if various portions of the information are captured in different places.

This paper describes the design and development process Seattle Genetics' Clinical Programming group used to create a web-based application that can be used for documentation, tracking, output review, and managing all of the information in one place. The paper also discusses various aspects of the application, such as usage by assigned role, support for company processes and business rules, benefits, and plans for enhancement.

INTRODUCTION

This may sound familiar. Our group, Clinical Programming, had developed a utility for tracking production of tables, listings and figures (TLFs), which included titles, footnotes, and development progress. Our group had developed another utility for storing dataset specifications, which included all of the CDISC information and code lists. As you can imagine, output review was less efficient and typically featured a collection of rich text format (RTF) documents placed in a network folder for the cross-functional team members to review within a given timeframe.

Although this process worked for a time, as the company grew, these processes became progressively more difficult to manage and maintain. The TLF tracker and the dataset documentation utilities were both created in Microsoft Access, and our group was now running into issues with scalability, versioning, managing multiple copies of these databases, and memory constraints. For the output review, there was no way to track who had reviewed which TLFs or document that the review had taken place at all. Additionally, without reliable metrics and tracking, our department struggled with resourcing and managing the "big picture".

Clearly, our department needed a better solution. Our dream was to have a single comprehensive, integrated utility that tracked all of the necessary information and more. The utility needed to be scalable, easily accessible to all potential users, and support our internal processes. What follows is the story of developing this better solution, known internally as the Biometrics Analysis Tracking and Management system.

ROAD MAP

"It's not about ideas. It's about making ideas happen." (Belsky, 2010) The first step towards achieving this dream of having a web based analyses tracker is to lay out a road map to achieve desired results.

Team

Regardless of scope, success is better achieved by harnessing talent, capitalizing on individual strengths and adopting a collaborative approach. This warrants building a team of personnel who can bring required skills, share experience and expertise, represent and address various user groups' expectations and be responsible and committed towards successful project implementation. In addition to the developer, our team was comprised of representatives from various departments which house potential users of the application; and a facilitator to coordinate discussions, solicit feedback, document minutes and decisions, and translate user feedback and requirements to the developer.

Proposal

A proposal was drafted to help garner support and resources from the senior management. The team went through a diligent exercise of highlighting the need for a solution that could aid study teams in carrying out the assigned tasks with greater efficiency, listing the shortcomings of current utilities, evaluating available alternatives that could be developed in-house or acquired from third party vendors and identifying a solution that is well vetted and best suited for the organization given known constraints. The proposal also included estimates on fixed costs and time required for requirements gathering, development and roll-out.

Milestones

A number of project milestones were defined by the team to support the ultimate goal of rolling out the application such that users could start tracking analyses.. A few high-level milestones were Gathering and Evaluating User Requirements, Database Design, Prototype Development and Evaluation, Test Version Development and Deployment, User Acceptance Testing and Evaluating Feedback, Production Version Roll Out and User Training.

DEVELOPING THE UTILITY

Development of the application began once the project was approved and funded, and the developer had been selected. Development involved several important activities: breaking the project down into manageable phases, codifying the processes and business rules to be embedded into the utility, working with the developer to design and implement the utility, and then testing and rolling-out the utility. Each of these activities was led by the core project team, but other participants and subject matter experts were incorporated as needed. The core project team consisted of the authors, the contracted developer, and one representative each from our Medical Writing and Biostatistics groups. This kept the team to a workable size while ensuring all of the major stakeholders were involved.

Determination of Project Phases

Creation of this utility was envisioned to be a huge project, and we were not able to get backing for the entire utility up front. Instead, our group was tasked with demonstrating the value of the application in order to justify funding of subsequent phases. As a result, it became important to identify which aspects of the utility made sense for the first phase of the project and which aspects should be delayed for later phases.

Since the developer already had a core product that was designed to track analyses, it made sense to use that as a starting point. Our team decided that since output tracking and review would impact many groups and be highly visible, we would also add in the ability to capture detailed metadata about each output associated with the analysis along with the capability to track the development progress and review of each output. This was determined to be reasonable within the limits of our original budget. Our team also felt that this would be a useful and beneficial package, even if we were unable to secure backing for later phases of the project.

Integrating the capture of database specifications and related metadata was relegated to the second phase since it affects mostly Clinical Programming and would have little impact on or visibility to other departments.

Codifying Processes and Business Rules

With the scope of the first phase determined, we turned to the question of how to implement these functionalities. In order for that to happen, we had to dig down into the details of exactly what actions were expected to occur, the order in which they should occur, which user roles were allowed to perform what actions, and how best to guide the users through the appropriate actions at the appropriate times. Our team had a strong feeling that the utility should support our internal processes and standards, while at the same time allowing some necessary level of flexibility.

We began by breaking down the processes that we wanted to include in this phase of the project and mapping them out in detail. (Figure 1) This involved reviewing standards documents and meeting with representatives of various teams and groups. Throughout this phase, we encountered diverse interpretations of standards and variations on processes that had been implemented by sundry teams and it ended up taking longer than expected.

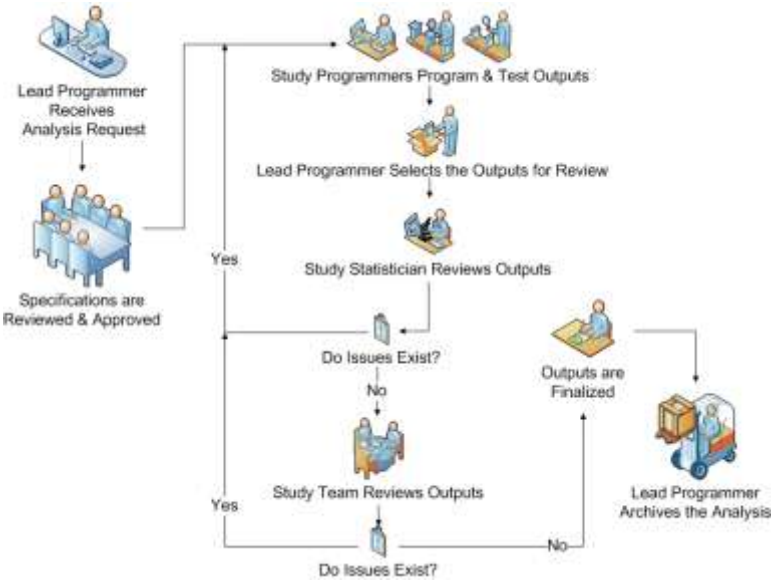

Figure 1: Standard Process for TLF Production

Another major task was defining the business rules with regard to specific roles and responsibilities as they related to the utility. The developer created a grid of possible actions versus process steps, and we then populated the grid with information on what actions could be performed during which process steps, which roles could perform each action during that step, any limitations on the ability to perform the action, and any notifications that needed to be provided to users as a result of that action. This went through several rounds of review and revision with the developer as we clarified terminology and discussed the impact of specific choices.

DESIGN AND IMPLEMENTATION

At this point, we had a pretty good idea of where we were headed and the developer began programming in earnest. In order to catch issues as they arose, we met with the developer on a regular basis to review design concepts and implementation details. This typically involved going over database table structures and relationship diagrams, or focusing on specific user interface screens. Special attention was given to ensuring the database would capture all of the necessary data, and the organization of tables and relationships would support the work processes. Similarly, ensuring that the user interface screens were acceptable affects the usability of the application and encourages adoption by the users.

Between scheduled meetings, we conducted informal unit testing of various modules in the development environment whenever the developer felt they were ready. This allowed us to provide regular feedback and catch any issues as soon as they arose. It also allowed us to try out multiple different implementations in order to identify a workable approach.

System Design

The Biometric Analysis Tracking and Management utility is an ASP.NET and MVC database web application, coded mainly in C# using Visual Studio 2013. The application is hosted in IIS 8.0 on Windows Server 2012. It has a three-tier architecture consisting of a client browser to communicate with the web server, a middle tier that houses the business logic on the web server, and an SQL server data storage layer. (Figures 2 and 3)

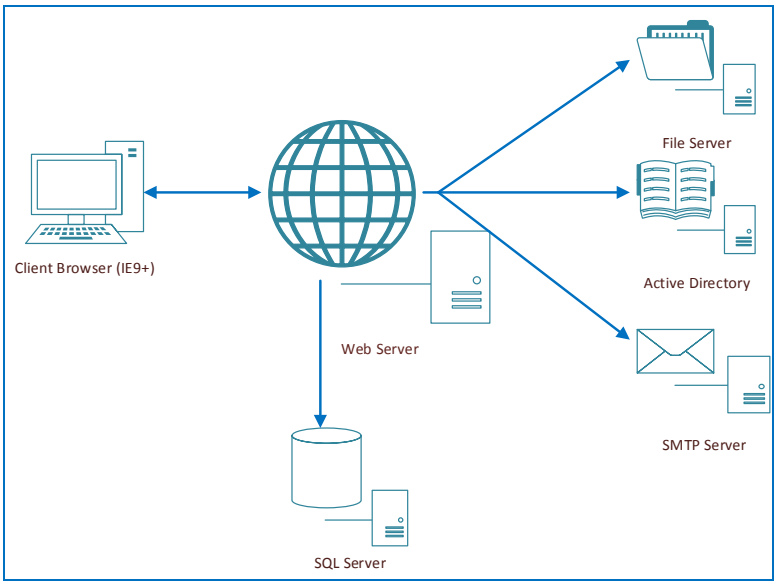

Figure 2: Application Architecture

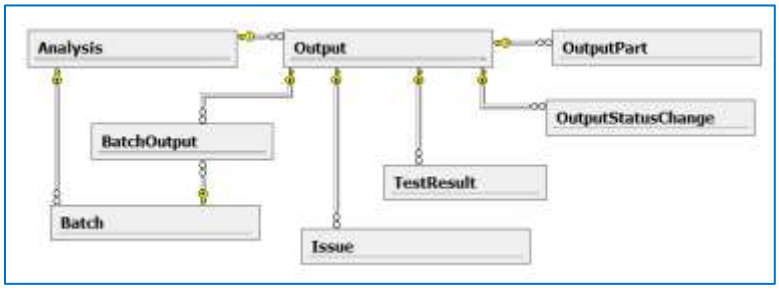

Figure 3: SQL Server Data Storage Map

Security for the application is provided via Window Integrated Authentication, allowing users to connect to the application without needing to enter a password. The application retrieves their credentials from Active Directory and defines their role-based permissions based on membership in one of several security groups.

The application impersonates the user's credentials to access the secure Clinical Programming file share and upload files to the web server for dissemination to the much broader study team.

The original intent of developing this utility was to provide a cross-department solution to managing and tracking clinical study workflow and metadata. However, as the design and development matured, innovative functionality was added to automate a variety of activities within the application. Several additional functions are currently in development or planned for the second phase of the project.

- Titles and footnotes – the application captures the title and footnotes as attributes of each output. This metadata is used to generate a SAS macro that integrates with the reporting utilities in Clinical Programming to automate title and footnote generation in outputs.
- Analysis archival – when an analysis is complete, the application creates a single output that records the workflow, from development to final team review, for each TLF and data set, which is signed and retained with the study documentation.
- SDTM and ADaM dataset specifications – using the dataset metadata, the application will create a SAS macro that enforces the CDISC standards defined in the application in the output dataset. (Phase 2.)
- Format catalogs – the application uses the CDISC controlled terminology for a study to generate a format catalog that allows programmers to map values between the raw data and standardized values. (Phase 2.)
- Define.XML – using the SDTM and ADaM metadata, the application can create Define.XML to support submissions. (Phase 2.)

Process Implementation and Tracking

An important aspect of this utility was that it needed to support our processes and help us track progress through the process. The two most important processes that we wanted the application to model and track were progress through an analysis, and development progress for individual components (i.e. datasets and TLFs) of that analysis.

In order to do this, we created a series of status indicators – or values – for each process. For analyses tracking, the status values are:

- **Started** – analysis is newly created in the system
- **Specification** – metadata, including general timeline, about the analysis has been entered and approved as complete
- **Active** – programming and review activities are ongoing
    - Includes one or more Review Periods that are divided into **Stat Review** and **Team Review**
- **Archived** – analysis is closed and archiving activities completed

Per our process, the lead programmer initiates the analysis in the system. This sets the status to 'Started' and begins the 'Specification' phase. They then assign team members, enter the timeline, and add initial metadata for the components. The biostatistician for the analysis then reviews this information and, if complete and accurate, closes the 'Specification' phase and moves the status to 'Active'.

Once the analysis is 'Active', the first Review Period is defined. The programmer adds TLFs to the Review Period and then biostatistician opens the 'Stat Review' phase. When they complete their review, they then close the 'Stat Review' phase and open the 'Team Review' phase. Once the study team has completed their review, the biostatistician closes the 'Team Review'. This process is conducted as many times as needed to ensure that all TLFs are ready for release. Following final release of the analysis, the lead programmer conducts the appropriate archiving activities and the analysis is set to 'Archived'.

Status values for individual components are:

- **New** – the item is newly entered into the system
- **Dev** – programming of the component is underway
- **Ready** – the item is ready for testing
- **Tested** – testing of the item is complete with no remaining issues
- **Reviewing** – the item is currently in a Review Period *(not used for datasets)*
- **Team Reviewed** –the item has been reviewed by the study team *(not used for datasets)*

When the programmer enters metadata for a dataset or TLF, the system sets the status to 'New'. The programmer then moves the status to 'Dev' when programming for that item begins. When the item is ready for testing, the programmer sets the status to 'Ready'. If the tester finds issues, they enter those as Test Results; the programmer then returns the item to 'Dev', makes the required changes, and resets the item to 'Ready'. On the other hand, if no issues are found, then the tester sets the status to 'Tested'.

If the item is a TLF and its status is 'Tested', it is eligible to be included in a Review Period. When the 'Stat Review' is opened by the biostatistician, the system sets the status of all included TLFs to 'Reviewing'. Once the 'Team Review' is complete, the system advances the status of the TLFs to 'Team Reviewed'.

Testing and Training

As the first phase of the project neared completion, we instituted more rigorous testing. This was approached in two ways: formal integration testing and user testing. The integration testing was based around a written test plan that incorporated testing of all of the planned requirements, including those identified in the process analyses and business rules. Each role was assigned to one member of the core project team, who then performed each of the tests assigned to that role in the test plan. Any issues that arose were reported to the developer, and then retested once a fix had been implemented.

For user testing, we assembled groups of users from various functional areas. Each participant was assigned the appropriate role in the system for their job function. They were then given a brief introduction to the system and allowed to explore on their own. The objective was to give them the opportunity to try things we had not thought of, as well as getting feedback on how intuitive the system was.

Upon completion of testing, the application was ported to the production environment and made available to all users. Clinical study teams were encouraged to use it for any upcoming deliverables, which led us to providing training for the users. Since people sometimes do not retain training well if they do not use the skills right away, we decided to take a two-prong approach to training.

First, we worked with the clinical study teams to plan and schedule training sessions to coincide with the first deliverable for which they planned to use the system. This "just in time" approach meant that the training would be fresh in their minds at the time that they would be using the system on a real project.

Second, we wanted to encourage self-initiated use and exploration of what we hoped was a fairly intuitive system. To support that intent, we created a series of activity-specific "quick start guides" – one page summaries of how to perform a specific action or function within the system. The system then provides links to these guides to the user based on the user's role in the system.

CONCLUSION

As our company grew, our Clinical Programming group found that the jumble of tools used to track various types of metadata were insufficient to the task. Additionally, our ability to track and manage analyses, both individually and in combination, was severely lacking. There was a clear need to create a better means of collecting and managing all of this information. A plan was made to develop an integrated, scalable, comprehensive solution, which we named the Biometrics Analysis Tracking and Management system.

The first phase of this project was rolled out to production in August of 2014 and clinical study teams were encouraged to use it for their upcoming deliverables. Our organization has had significant successes in using the utility and the project team continues to collect feedback on how it can be improved. Small improvements are triaged and implemented as appropriate. Major enhancements are documented and tracked for later inclusion. Meanwhile, the contracted developer is also training one of our internal applications programmers to eventually take over maintenance of the system.

Due to the positive response from users and clear benefits, the second phase received backing and is moving forward. For this new phase, we are implementing tools to capture dataset specifications, including all of the information needed to generate CDISC-compliant Define.XML files for SDTM and ADaM. This phase will also include a variety of usability enhancements that were requested by the users. The team is currently in the middle of development with informal testing and evaluation of draft modules being conducted. Roll-out is anticipated to occur in late first quarter or early second quarter of 2015.

REFERENCES

Belsky, Scott. 2010. Making Ideas Happen: Overcoming the Obstacles between Vision and Reality. London, England: Portfolio Trade / Penguin Books

CONTACT INFORMATION

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

Name: Gordon Fancher
Enterprise: Seattle Genetics, Inc.
Address: 21823 – 30th Drive SE
City, State ZIP: Bothell, WA 98021
Work Phone: (425) 527-2328
Fax: (425) 527-4209
E-mail: gfancher@seagen.com
Web:
Twitter:

Name: Rajeev Karanam
Enterprise: Seattle Genetics, Inc.
Address: 21823 – 30th Drive SE
City, State ZIP: Bothell, WA 98021
Work Phone: (425) 527-2812
Fax: (425) 527-4209
E-mail: rkaranam@seagen.com
Web:
Twitter:

Name: Shawn Hopkins
Enterprise: Seattle Genetics, Inc.
Address: 21823 – 30th Drive SE
City, State ZIP: Bothell, WA 98021
Work Phone: (425) 527-2340
Fax: (425) 527-4209
E-mail: shopkins@seagen.com
Web:
Twitter:

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

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