

Redesigning a Legacy: Techniques of a Quality Partner

S. David Riba, JADE Tech, Inc., Clearwater, FL

ABSTRACT

Most companies have legacy systems, applications that are regularly used but that have become old and obsolete. Often they require extensive maintenance or retooling. The programmers who developed these systems may have moved on to other projects, left the company, or may not have the requisite skills to satisfy current company requirements. Regardless, these systems will eventually become static and fail to meet the needs of the organization.

The examples presented in this paper are two very different systems developed by JADE Tech, Inc. They illustrate how a Quality Partner was instrumental in making legacy systems satisfy the objectives of their organizations. JADE Tech, Inc. is a SAS Institute Quality Partner which specializes in easy to use decision support applications and training in the SAS® System.

This paper will examine some of the issues involved in redesigning legacy systems and will illustrate some techniques for improving the flexibility of these systems. Originally developed internally by the Florida Department of Transportation (FDOT), the Long Range Estimates (LRE) system had grown excessively large and was no longer able to satisfy the internal requirements of FDOT. GAIN, the Global Auditing Information Network, was a PC based reporting system developed in Visual Basic™ and Quattro Pro by the Institute of Internal Auditors. In both instances, the resulting SAS® application systems developed by JADE Tech, Inc. were more efficient and more user-friendly while incorporating the enhancements requested by those using the system.

INTRODUCTION

This paper is intended to address issues confronting two different audiences who may be responsible for the maintenance or redesign of existing legacy software systems. First, the paper addresses the issues of successfully managing these types of projects. It looks at the types of issues that need to be considered and the decisions that need to be made. It also looks at issues of computer consulting projects, in general, and at SAS software consulting projects, in particular. Second, it is intended to suggest some features in the SAS® System that can be useful for the application developer tasked with completing these types of projects. Many of the issues are common to all software systems, while others are specific to systems written in SAS. These issues are illustrated by two case studies of recent projects completed by JADE Tech, Inc.

BACKGROUND

The SAS® System of the mid-1990's has changed dramatically from the SAS software most organizations are familiar with. Gone are the days when SAS was just used for ad-hoc reporting or statistical analysis. With products such as SAS/AF®, SAS/EIS®, the SAS® Screen Control Language, and the many new components to the SAS® System, SAS is no longer "your father's Oldsmobile". The rate of growth and change in the SAS® System continues at a rapid rate.

Over the last half-decade, most organizations have embraced the concept of becoming "lean and mean". The result has been a realignment of organization priorities and dramatic reductions in the size of their work forces. Whether it is called "downsizing" or "rightsizing", the results have been a much smaller work force tasked with meeting the organizations objectives. At most sites today, employees are carrying workloads that were previously performed by three or more co-workers with few additional resources. They do not always have the luxury of keeping on top of changes in technology.

These two trends - enhanced capabilities in the SAS System and a "lean and mean" organization - have created new issues in successfully managing the information management requirements at most organizations. These issues include:

- ◆ Enhanced capabilities in the SAS System often go unused because the organization may not have the resources available to learn and apply these capabilities. They may not have the opportunity to become more efficient or to take advantage of new technology.
- ◆ A smaller organization means that the existing workforce may not have the time or resources to respond quickly to new requirements. They may be too busy to retrofit existing systems.
- ◆ Many organizations who have concentrated on reducing employee headcounts no longer have the luxury of a pool of employees who are full time specialists in specific niche areas.
- ◆ Organizations have found that it is sometimes more cost effective to "hire" an outside expert in specific technologies on an as-needed basis instead of developing and maintaining the levels of expertise internally

The results of these trends has been an increase in the demand for computer consultants in general. The growth in features and capabilities in the SAS System, in particular, has led to the development of many specialized areas of expertise that are in high demand. Expertise in the SAS System is no longer just a matter of understanding base SAS, statistical analysis, the macro language, or SAS/GRAPH®. Specific areas of expertise now also can include interactive applications development, client / server computing, data warehousing, platform specific installations and tuning, vertical market specific, and horizontal functional areas, among many others.

WHAT IS A QUALITY PARTNER?

One result of the growth in demand for specialized consulting expertise in the SAS System has been the development of the Quality Partner program at SAS Institute. Until now, organizations have not had a means to identify consultants specializing in the SAS System who had the expertise to solve their specific problem areas. Too often, projects were unsuccessful because of a lack of qualified consulting resources to accurately complete them.

The Quality Partners program was developed at SAS Institute to meet the needs of the SAS user community by identifying SAS software consultants who have proven expertise in the SAS System. The

North American program is patterned after the very successful European program, which has been in existence for several years.

SAS software consultants who have been designated as Quality Partners have demonstrated expertise in the SAS System and a commitment to high standards of quality. They typically have committed the resources to support changes and new features in SAS software.

In conjunction with SAS Institute, Quality Partners can provide expertise in the leading edge technologies of the SAS System for the benefit of their clients. The commitment to quality and the close working relationship forged with SAS Institute assures that clients are provided with the expertise they need.

ABOUT JADE TECH, INC.

Based in Clearwater, Florida, **JADE Tech, Inc.** was among the first Quality Partners announced by SAS Institute. For over eight years JADE Tech, Inc. has specialized in applications development, consulting, and training in the SAS® System. Their primary emphasis is in the development of easy to use decision support applications; the re-design and enhancement of existing legacy systems; and formal training and support in the SAS® System.

LEGACY SOFTWARE SYSTEM PROBLEMS

Every organization has problems with legacy systems. It is unfortunate, but once a computer system has been written it is immediately obsolete.

- ◆ Decisions that might have seemed valid during the design phase of a project may prove incorrect once a system goes into production.
- ◆ New circumstances that were not anticipated during design arise with use or over time.
- ◆ Users want new features.
- ◆ More efficient programming techniques become apparent after implementation
- ◆ New technologies can be applied to make a system more efficient, improve the user interface, or perform more complex tasks

For these, and many other reasons, systems become obsolete. Organizations are then faced with a dilemma. One option is the status quo - freeze the existing system and not allow any changes. A

second option is to scrap the existing system and purchase or write an entirely new system. A third option is to allocate resources to maintain and improve the existing system.

This paper deals with the issues involved in the final option, the maintenance and enhancement of legacy systems. In the past, many organizations have chosen to allocate the needed manpower internally to support maintenance or redesign of these systems. Recently, organizations have increasingly relied on alternative sources of manpower to provide this support. This decision to use external resources implies a totally different set of project management issues.

REDESIGNING LEGACY SYSTEMS

Once it has been decided that a legacy system needs to be updated, several issues should be addressed:

- ◆ Is there adequate manpower available internally with the necessary skills to successfully complete the project?
- ◆ If not, are the necessary manpower resources available elsewhere?
 - ◆ Contractors
 - ◆ Consultants
 - ◆ Third Party Application Developers
 - ◆ Other resources
- ◆ What is the scope of the project?
 - ◆ Maintenance upgrade
 - ◆ Add new features
 - ◆ Implement new technologies
 - ◆ Redesign the system
- ◆ What is the budget for the project?

It is extremely important to the success of any project that these questions be discussed and resolved. Many of these issues will be re-addressed several times as the definition of the scope of the project evolves. Ultimately, the Project Manager needs to have a clear understanding of the following questions:

- ◆ **WHAT** do I want to do?
(Scope of project)
- ◆ **WHO** do I want to do it?
(Resources needed - internal, consultants, applications developers, etc.)

- ◆ **WHEN** does it need to be completed?
(Timelines for completion)
- ◆ **WHERE** will the project be done?
(Development on-site or off-site)
- ◆ **HOW** much funding is required?
(Budget for the project)

There is one final decision that needs to be made after defining each of the above issues:

- ◆ After reviewing all the issues and decisions, **DO I STILL WANT TO PROCEED WITH THE PROJECT?**

Assuming the decision was made to proceed with the project, a project team knowledgeable in the requirements is assembled and a project manager assigned. Choice of the Project Manager can be critical to the success or failure of a project.

If the decision is made to go outside the organization for the resources necessary to complete a project, it is essential to accurately define the amount of internal support that will be required for the project. If contractors are used, then a work site, computer access, telephones, etc. need to be provided. For consultants or third party application developers issues such as location, SAS software licensing, access to organization data, etc. need to be resolved.

Critical to the success of any project, especially one that deals with external resources like consultants or third party application developers, is that all parties need to have a clear and mutual understanding of the WHAT, WHO, WHEN issues. An outside resource is usually not privy to the internal issues that may affect a project. Hidden agendas and office politics can only hamper the successful completion of a project. A mutual understanding of the project scope, timelines, required milestones, etc. is essential. Some method must also be implemented to measure progress and resolve problems before they become critical to the success or failure of the project.

Regular progress milestones and periodic update briefings are important to keep communications open between the parties. It is useful to note that the "mushroom" theory of management, which some managers still subscribe to, is counterproductive when dealing with outside resources such as consultants. When paying a consultant for their expertise, it is usually not prudent to "keep them in the dark". The quality of their results is directly related to their understanding of your requirements.

For a successful relationship with an outside resource such as software consultants, there are other issues that also should be addressed. There are contractual issues, rights of ownership, employer/employee relationships, etc. These are beyond the scope of this paper and mentioned only in passing.

The remainder of this paper will focus on two real world case studies that illustrate many of the issues just discussed. They are based on recent SAS consulting projects completed by **JADE Tech, Inc.** Both involve the redesign and enhancement of existing legacy systems, but in totally different environments.

LONG RANGE ESTIMATES (LRE METRIC)

The Long Range Estimates system was a SAS System developed internally at the Florida Department of Transportation. It was designed to provide accurate cost estimates for roadway and bridge construction projects long before the full scope of the project was finalized. It evolved over the years as several programmers maintained and modified the code to the DOT's requirements. The system operated on an IBM MVS mainframe and was written primarily to Release 5.18 of the SAS System. It was in daily use at FDOT offices statewide.

Over time, the system grew to the point of being unmaintainable. The code had evolved into spaghetti code. One PDS member alone consisted of over 5,000 lines of SAS code. There were significant internal inefficiencies in the way the system processed information requests. Internal resources at FDOT were limited and fully allocated to other requirements. There was a backlog of user requests for changes to the system. In addition, FDOT needed to modify the system to comply with a new Federal mandate to metrify existing measurement systems.

To resolve these problems, the Florida Department of Transportation chose to use the resources of an outside organization. JADE Tech, Inc. was awarded a competitively bid contract to redesign and enhance the LRE system. This project was a formally structured project, with pre-defined objects, interim milestones, and a formal review and evaluation process.

Development work was done by JADE Tech personnel on their own equipment in Clearwater. The hardware independent architecture of the SAS System proved an advantage, as development was

done under Microsoft Windows™ and then ported over to MVS. The port of the SAS application to MVS was reasonably effortless.

Each element of the old LRE system was reviewed and analyzed. Obsolete and redundant code was streamlined. The structure of the SAS datasets was analyzed and simplified. The user interface was reviewed. In short, every aspect of the old LRE system was evaluated. No change was made to existing software without a thorough understanding of the implications of the change and the prior approval of the FDOT Project Manager. The necessary modifications were made and the new features that were requested were added.

In many instances the existing LRE system had duplicated similar functions and operations many times, each with a minor change. Instead of several nearly identical data selection screens, a single multi-purpose screen was developed which self-configured based on the parameters that were passed to it. Thus, a change to a common user interface element was only needed once instead of on multiple screens.

There were a variety of other changes that were made to the LRE system to improve the efficiency, simplify future maintenance, and address user interface issues. The changes to LRE included:

- ◆ the elimination of all PDS and external flat files
- ◆ development of a single SAS catalog that contained all elements of the system
- ◆ redesign of all screen displays for a consistent user interface
- ◆ replacement of PROGRAM entries with SCL entries where no user interaction was required
- ◆ a 25% reduction in the number of variables stored by the system
- ◆ implementation of security features to restrict user access to information
- ◆ context sensitive help behind all data entry screen fields
- ◆ elimination of redundant code and a general retuning of the existing SAS software for efficiency
- ◆ utilization of new functions and features in SAS Release 6.10 where appropriate
- ◆ elimination of hard coded values and replacement with easily maintainable parameters

Ultimately after all of the changes were made and the new features included, the LRE system had been reduced from 41 PROGRAM entries and 15 SCREEN entries to only 10 FRAME entries and 7 SCREEN entries. The external PDS member which contained over 5,000 lines of SAS code was reduced to 4 SAS SOURCE entries with a total of only 3,800 lines of SAS code, which included all of the new analysis. Smaller, tighter source code meant that the new system operated faster and user requests were processed quicker.

GAIN

GAIN was a project with a much more modest scope. GAIN, the Global Auditing Information Network, was a victim of its own success. It was designed by the Institute of Internal Auditors to collect information, analyze it, and report back the results to IIA clients worldwide.

GAIN began with a PC based questionnaire written in Visual Basic™ for DOS. The questionnaire generated 150 ASCII delimited files with over 500 data points. The responses to the questionnaire were scrubbed manually and then loaded into Quattro Pro™, where almost 100 individually customized charts, graphs, and tables were generated for each client. This application was developed internally at IIA, but proved very labor intensive. In addition, the amount of time needed to process the results for each client effectively limited the growth of the system.

The Institute of Internal Auditors made several decisions at that point. For GAIN to meet their objectives, it would have to become more efficient. The decision was made to rewrite GAIN in SAS and replace the Quattro Pro charts with SAS/Graph. The second key decision was to use outside resources instead of developing the SAS application internally.

Unlike LRE which consisted of the redesign of an existing SAS application, the GAIN project involved the duplication of a non-SAS application using SAS software. Even though it was a smaller and much less formal project structure, there were several strong similarities that contributed to the success of the project.

- ◆ A single point of contact with the authority to make decisions.
- ◆ A well-thought out and very detailed definition of the requirements of the project.

- ◆ Identification of significant milestones to track progress and identify potential problem areas early in the process.

Development was done off-site by JADE Tech, Inc. using the SAS System for Windows, SAS/AF, SAS/GRAPH, and Screen Control Language. The system incorporated the latest features in the SAS System, Release 6.10. One interesting sidelight was the use of another technology to speed communications. Software under development was sent to the client for review, testing, and comment using e-mail via the Internet.

The result of the project was a simplified user interface that allowed point-and-click selections based on user input. All elements of the GAIN system were stored in a single SAS catalog. User interface screens that reconfigured themselves based on replaceable parameters. All user definable parameters, such as client grouping definitions, were stored in easy to modify ASCII text files. These external files were read at run-time and the program self-configured based on the values in the files.

TECHNIQUES FOR IMPROVING DESIGN FLEXIBILITY

While there are many efficiency techniques that the experienced SAS programmer can apply to improve their programs, several suggestions are particularly applicable when dealing with legacy systems.

Listed below are several ideas that may be helpful:

ELIMINATE UNNECESSARY VARIABLES

Programs that are subject to routine maintenance typically suffer from variable creep. As new features are added to a system, and other features become obsoleted, unneeded variables have a tendency to remain in place. The occasional index variable or temporary placeholder variable may not seem very significant. Yet, each variable increases the storage requirements for the data set, increases I/O and processing time. For example, it takes time to sort a data set. If that data set contained 25% fewer variables, consider the time savings that would accrue every time that data set needed to be sorted.

AVOID STORING REDUNDANT INFORMATION

The same reasoning applies to the structure of the data that is stored. Redundant data results in the same types of problems as storing unneeded variables. Defining a six character variable to store

the text 'FEMALE' takes up five extraneous characters in every observation. Storing the one character 'F' or 'M' can yield the same results with no loss of precision. Consider using the features of PROC FORMAT to convert abbreviated information into its longer equivalent.

SIMPLIFY, SIMPLIFY, SIMPLIFY

Program code rarely gets more compact during routine maintenance. Every bug fix or added feature only adds to the size of existing code. Retrofitting existing code provides a good opportunity to seriously look at all sections of a program. Eliminate obsolete code. Consolidate redundant code. Clean up the program structure. Look at programming bottlenecks are improve the program flow.

IMPROVE THE USER INTERFACE

Like program code, the user interface has a tendency to evolve and become a patchwork. This is a good opportunity to enforce a consistent interface style. Consider the features that would make the system more intuitive for the user. SAS now has the tools for building better interfaces. Apply the ones that are appropriate.

ELIMINATE UNNECESSARY SCREENS

Along the same idea as the above, there are alternatives to painting a screen for every interaction with the user. Consider the use of pop-up menus for a very quick selection. Painting screen takes time and system resources. For quick information or a selection among choices, a pop-up is very efficient.

AVOID HARD CODING VARIABLE INFORMATION

There is nothing as frustrating, and that increases the potential for error, as searching through reams of source code looking for every occurrence of a value that needs to be changed. There is a high probability that at some time, at least one instance of the value to be changed will be missed. Instead, this maintenance trap can be eliminated by a simple technique. Replace hard coded values in the source code with a macro variable. In a single location, typically at the beginning of the SAS routine or in a separate load module, define the current values for each of the macro variables. Thus, a single change to the value is automatically available everywhere it is needed.

CODE FOR FLEXIBILITY

To continue with the idea of eliminating hard coded information, SAS System applications have the capability to configure themselves based on outside factors. For the LRE system, we were able to take a basic SAS/FSP data entry screen and have it modify its display based on the contents of each observation

and the user's access authority. It takes a bit more coding, but in the long run this technique simplifies maintenance and support by eliminating many parallel entries that are almost exact duplicates. Parameter-driven (or table-driven) programming has many advantages that can be exploited to create a better application.

CODE FOR MAINTAINABILITY

Too often, computer source code outlives the original developer. Whether the developer has left the organization, been assigned to other tasks, or is no longer available, the ongoing maintenance to most systems is typically delegated to others not familiar with the thought processes of the original developer. It is imperative that all source code must be understandable and easy to maintain. Cryptic programming techniques may result in some measure of job security, but they will definitely result in maintenance and support problems in the future. SAS software has several advantages here, since typically SAS programming style results in variable names that are reasonably understandable. This does not eliminate the requirement for easy to follow documentation and understandable program code.

MAXIMIZE CAPABILITIES OF THE SOFTWARE

Recent releases of the SAS System have incorporated new technologies, new functions, and a host of new features to improve the efficiency and functionality of SAS applications. Too often, developers rely on the "tried and true" and do not take the time to understand the tools they have been given to work with. Often, a few minutes of research into some of the newer software capabilities of the SAS System can save hours of coding the old fashioned way.

CONCLUSION

This paper has attempted to illustrate some of the issues that need to be considered when faced with the task of upgrading or redesigning legacy software systems. At some point in the life cycle of every software system a decision is needed whether to maintain the status quo, upgrade the system, or replace it with a newer system. This paper has tried to provide the reader with some useful ideas of the issues that need to be considered when involved with these types of projects. Among the options that need to be decided is whether to perform the work internally or use an outside resource. The role of a SAS Institute Quality Partner was examined and some examples were presented of how a Quality Partner can be instrumental in assisting organizations in this process.

The author welcomes comments, suggestions, and questions by phone **(727-726-6099)** or by e-mail **DAVE@JADETEK.COM**.

TRADEMARK INFORMATION

SAS is a registered trademark of SAS Institute Inc. in the USA and other countries. ® indicates USA Registration.

Windows is a trademark of Microsoft Corporation.

AUTHOR

The author may be contacted at:

S. David Riba
JADE Tech, Inc.
P O Box 4517
Clearwater, FL 33758

(727) 726-6099

INTERNET: DAVE@JADETEK.COM

ABOUT THE AUTHOR

S. David Riba is CEO of JADE Tech, Inc., a SAS Institute Quality Partner who specializes entirely in applications development, consulting, and training in the SAS® System. He is the founder and President of the Florida Gulf Coast SAS Users Group. He chartered and served as Co-Chair of the first SouthEast SAS Users Group conference, SESUG '93, and serves on the Executive Board of the SouthEast SAS Users Group. His first SUGI was in 1983, and he has been actively involved in both SUGI and the Regional SAS Users Group community since then. He has presented papers and assisted in various capacities at SUGI, SESUG, NESUG, MWSUG and SCSUG.

Dave is an unrepentant SAS bigot. His major areas of interest are efficient programming techniques and applications development using the SAS® System, particularly using Screen Control Language with SAS/AF® and FRAME technology.

