Paper 175-2010

Quality Outcomes and SAS:
Structuring, Managing and Working in SAS®-Based Programming Groups

Joe Perry, Perry & Associates Consulting, Inc. Oceanside, CA

ABSTRACT

Compliance with programming standards is part and parcel of managing or working a programming group; whether a C++ or SAS® based group. Managers and programmers find themselves in a constant struggle to find the proper balance between too much and too little structure, between enough direction and oversight but not too little, nor too much.

To obtain quality products or services, programming groups have a wide array of structures in place like: policies, work practices, reporting relationships, personnel and quality practices designed to produce these quality work products. These system structures, when properly designed, and with proper oversight and control result in win-win situations and everyone wins: programmers, management and their customers.

Conversely, misalign any or all of these structures and an organization generally gets, at best, a win-lose or in the worse-case scenarios, lose-lose situations that can, and often do, result in customers, programmers and even management looking around for the exits, both figuratively and literally.

This paper discusses general, quality system design: methods, structures and work practices for obtaining quality outcomes from SAS-based programming organizations. It discusses how to apply and use quality methodologies consistent with the Software Engineering Institute (SEI) and other SDLC related standards to a SAS-based programming organization. It focuses on the actions and responsibilities of *both* the managers and the programmers within these structures; responsibilities that must, necessarily, be met if an organization is to obtain both compliance to the quality structures and the quality assured outcomes they are designed to produce.

BACKGROUND

In addition to my 22+ years of SAS programming and development work; for the last -2- years I have been performing statistical and statistical programming qualification and process audits for sponsor organizations; this experience has provided me with a strong motivation to start a dialogue on the topic of statistical programming work and obtaining quality outcomes at low cost.

Previously, I worked for 9 years in a quality organization within Boeing Commercial Airplanes Group with the responsibility for facilitating the design and implementation of quality systems throughout the wide-bodied plant (mostly are statistically based but, the whole systems approaches to quality was my responsibility.) Heavily engaged with both engineering and software groups like mechanical, electrical, manufacturing and software engineering groups, provided lessons on the application of Quality Engineering principles to the SAS programming world.

Over the last two years of my performing audits, it has become obvious that there is a direct relationship between the observed SAS programming outcomes and the lack of the application of Quality Engineering principles to SAS programming organizations. Issues like: poor documentation of quality checks, poor to no procedural controls and escapes of programming errors in output to the client organizations are generally accepted as ‘to be expected’ in much of the SAS world. The fact remains that in all other programming disciplines in the programming world, in general, these issues reflect controllable outcomes by the design and operation of the quality system.

Though performed in an FDA regulated environments, my other experiences in Aerospace and Telecommunications leads me to conclude that these audit results reflect a general SAS related affliction. That affliction is that quality system design principles are either poorly understood or, understood but thought to be either not applicable to SAS-based programming or, most likely, too costly to use in the SAS organizations through the constraints placed on the quick response times SAS practitioners pride themselves on.

STARTING AT THE BEGINNING

In *The Sciences of the Artificial* by the Nobel Prize winner, Herbert A. Simon, in the section “State Descriptions and Process Descriptions[i]” he stated, “*We pose a problem by giving the state description of the solution. The task is to discover a sequence of processes that will produce the goal-state from an initial state. Translation from the process description to the state description enables us to recognize when we have succeeded.*” Given this definition, it is clear that stating the goal-state, *first*, is critical; not so clear from this statement is that the goal-state must be generally devoid of process-type statements[ii].

Oftentimes, for the activities and environments we are most familiar with; creating a clear, concise statement of goals is, at best, a painful process so, when presented with the need to create these end-state descriptions, a common response is an involved sequence of steps, actually a process description – the business-as-usual model. The business-as-usual decision does not conform to the last step in the above sequence, i.e. we do not translate from the process description to the state description and determine success because no state description exists for this objective.

We naively assume that because we have defined a process and that we *must have* achieved our end-goal(s). So, organizational personnel, both management and workers alike, are faced with a conundrum, defined actions without objective measures defining the goal, a conundrum that is typically solved by declaring victory and moving on to the work at hand.

We are still left with the fact is that there is still no way to know whether the process we followed has achieved it goals or, even more importantly, how to change what the people do and how the processes are designed so we can change the situation.

ORGANIZING FOR QUALITY

Rarely arrived at by serendipitous confluences of events and people lasting years or decades, quality outcomes are instead the intended results of conscious decisions and actions to achieve a goal influenced by: clear, concise guiding principles with the necessary associated operational requirements. The requirements address:

1) The structures of the groups and how they are organized for work;
2) Their procedural controls (Policies, SOPs, Work Instructions or Guidelines, and Supporting Documents);
3) How the organization's leadership guides and directs the actions of their personnel;
4) How the personnel are trained and inculcated in the quality expectations and the system of work used in the organization; and
5) How the organization's personnel chose to work within those structures; and
6) What actions to take to harmonize choices made in #5 with the objectives in #4 above?

A corollary to the above is "IF you are not getting the desired quality outcomes from your organization, consistently and within your cost objectives, then the source of those failed outcomes flows from a lack of one or all of the quality system components to function as intended; either due to: environmental, operational or usage deficiencies."

PARADOXES OF CONTROL ACTIVITIES

Related to the failure to manage tasks with vigilance, failed quality outcome goals are simply the inability to get: procedural controls, personnel, and/or personal styles of work to function in the same operational framework; by either managers or workers. To paraphrase Thomas Jefferson "The price of *quality outcomes* is eternal vigilance" but many SAS programming managers, as well as their workers, believe that their responsibilities for quality outcomes ends when policies and procedural documents are first written and implemented and managers and the then hire and train the qualified personnel (or they are hired and trained.)

It bears repeating again, "The price of *quality outcomes* is eternal vigilance" but, paradoxically, **monitoring[1] your desire to be vigilant**, and the level of that vigilance you are willing to affect is by ***your ability to affect change in the organization with your effort***. So, know not what to act on or act on the wrong factors, or at the wrong time or with the wrong energy and the results will be less than desired and, here is the paradox, your willingness to continue acting will diminish.

STARTING WITH THE END IN MIND

I believe that with deceptive ease, high performing quality systems regardless of the type of work performed get quality outcomes through the synergies from the operational interplay between the system components and how they are used. However, generally hidden to all but the most discerning eyes, is the fact that there is parsimony in the design and use of the various Quality Systems' components; that is, we get better system performance when that system is built with and uses no more, and no less than, what is needed to accomplish each function. Conversely, poorly performing quality systems fail to obtain the desired quality outcomes, and do so with expenditures of far more resources than higher performing quality systems. These are not trivial insights into the effectiveness of right choices! My experience shows that correcting some simple factors results in massive changes in outcomes, fail to correct these factors or correct only 98% of what is needed affects your outcomes to the greatest effect.

---

[1] Monitoring –verb "to watch closely for purposes of control, surveillance, etc." from http://dictionary.reference.com/browse/Monitoring

Paper 175-2010

Quality Outcomes and SAS: Structuring, Managing and Working in SAS®-Based Programming Groups

Joe Perry, Perry & Associates Consulting, Inc. Oceanside, CA

During my career when working in the field of quality well as a SAS programmer and developer, a number of quality systems related issues appear endemic in the SAS world. In the next two lists, I have categorized these issues into roughly five groupings and show both sides of each group, for a fuller list of associated with each of these see the section labeled “OPPORTUNITIES FOR IMPROVING SAS-BASED QUALITY OUTCOMES”

QUALITY SYSTEM PERFORMANCE COMPARISONS

Poor Performing quality ‘systems’[2]

1) Operate with General Myths about personnel and structures;
2) Have Poor Operational Controls in Place -- to direct and channel work effort towards quality outcomes;
3) Use of Passive Control Methods, especially in the Tone of Communications;
4) Have Ill-Defined Quality Structures and Systems; and
5) Show Inconsistent Skills Development, including mentoring and other activities focused on moving employees through the levels of skill attainment found in the skilled trade’s classifications of: Novice ➔ Apprentice ➔ Journeyman ➔ Master.

High Performing quality systems

1) Assume workers (including those in profession related jobs) are basically conscientious but, efficacious quality systems requires real-time, evidence-based decisions to operate correctly and, in the converse, past performance gives, at best, expectations of acceptable work in the future but EXPECTATIONS are not EVIDENCE;
2) Have Operational Controls in place that have been refined and perfected based on demonstrable evidence that the procedural controls result in compliant outcomes;
3) Use Active Control Methods including: a) Active procedural language without passive forms of verbs, imperative forms of both words like MUST rather than SHOULD, WILL rather than CAN[3] and b) Active supervisory and mentoring activities that detect non-compliant work with robust methods to change non-compliant to compliant behavior;
4) Well-Defined Quality Structures and Systems including active internal and external audit programs with defined and effective quality measures; and
5) Show Consistent, Directed Skills Development, including mentoring and other activities focused on moving employees through the levels of skill attainments found in the skilled trade’s classifications of: Novice ➔ Apprentice ➔ Journeyman ➔ Master.

This categorization may seem arbitrary but I believe that the choices can leads to a fruitful discussion on improving quality outcomes within a SAS organization. Lest I demoralize you before you finish reading this paper, it seems prudent to create a framework for these classifications and, hopefully, a proto-path out of the wilderness of quality at a more general level.

CHANGING THE PARADIGM

During my time as a full-time quality professional at Boeing, I had many opportunities to learn from or interact with some of the early, great, pioneers in the quality world in the USA but one of the most profound experiences was with a man named Paul Steel[4] who had worked with Dr. Armand V. *Feigenbaum* (originator of "*Total Quality* Control"[iii].) At one of the talks he gave in the Seattle area he described the difference between the systems (and consulting paradigm) used by both Dr. Feigenbaum’s and his company when assisting in the implementation of Total Quality Systems.

In Dr. Feigenbaum’s TQS consulting paradigm, the consultants worked from a proven process with (at least) two major components:

---

[2] Generally, the definition for ‘system’ found, i.e. “a coordinated body of methods or a scheme or plan of procedure” does not apply to most SAS organization’s quality activities as few of these components are not coordinated.

[3] Must is operationally verifiable – it’s either HAS IT or DOES NOT HAVE IT, SHOULD is like-wise indeterminate in that if it does NOT then it STILL conforms to the requirements. It has been argued that MUST and WILL are over-determinate but it is clear, from a quality point of view that SHOULD and WILL are indeterminate and any actions satisfy this requirement, therefore the requirement is ineffective.

[4] Paul Steel, President of ***Total Quality*** Inc (a Seattle area based company) and, per the web site http://baldrige21.com/Paul%20Steel.htm, Paul is a 2009 member of the Baldrige Board of Examiners, and an original 1988 Baldrige Award Board of Examiners member

1) They had a standard boilerplate of end-state organizational structure and consensus building procedures including a specific, non-negotiable 3-year implementation timeline; and
2) Second, that an evidence-based requirement that an independent, external, consultant lead the implementation of the quality system in each company.

A review of the above shows that:

a) They ***always*** started with a clear, concise vision of where they were going (though not the form that it would take in each company);
b) A three year company-wide implementation was possible and being accomplished; and
c) Thirdly, there are profound dynamical forces within an organizational that prevent it from changing quality paradigms from within.

CHANGING OUTCOMES

As Cesar Milan demonstrates in the cable show ‘The Dog Whisperer’, act on the right factors at (approximately) the right time with the right energy and the results will meet or exceed expectations and (using the principle effect from the monitoring construct) your willingness to continue acting will increase. Therefore, the key to a high performing and well-functioning quality systems lies in knowing the following:

1) What to act on;
2) Who should act;
3) How to act;
4) When to act;
5) With what level of energy; and
6) When *not* to act.

The greatest benefit to the organization comes from transiting through these six states of control in the order shown in the list. The answer to final question, #6, Deming would say lies in knowing when a process is in statistical control (i.e. in a state of equilibrium and hence predictable) and, that the existing controls are adequate within the norms of: activity level; complexity of work; and the current personnel mixture.

Most quality systems found in SAS-based programming groups assume that #6 above, i.e. a self-regulating system is the current condition of their system, and ignore the activities required to journey through these stages of control; this results in few effective controls on the system and generally poor quality outcomes in the SAS programming world.

LESSONS LEARNED AS A SAS-FOCUSED QUALITY AUDITOR (a partial list)

1) It is a myth, from a practical point of view, that people will follow rules without an enforcement mechanism, there is little motivation to follow rules consistent with others in a group without monitoring and enforcement mechanisms;
2) Quality documentation is the only evidence that counts as evidence that something has been done, and done correctly (paradoxically, a substantial number of audits of a company’s vendors require that conditions, audits of the company insist this is not a necessity);
3) Auditable quality documentation is a collection of documents that contains:
    a) A full enumeration of the end-state requirements[5] for WHAT was to be done;
    b) The EVIDENTIARY PROOF that each requirement was met (if practical, the full evidence or, if not practical, a representative sample of the reviewed documentation that proves the claim);
    c) Signed and dated attestations that the work was completed, when stated and fully meets the quality expectations for the product[iv];
    d) It is notable that, ***auditable documentation does not include documentation that requires a duplication of a) through c)*** but that the documentation created during the QC process is retained and arranged such that the a conclusion can be determined based solely on the QC documentation;
4) Procedural documents are no substitute for monitoring and mentoring of employees, whether they work in a management role (like leads) or perform programming tasks (like programmers) – this fact alone gives rise to the Myth statement in #1 above;
5) Monitor people’s work, do it early, and often; issuing simple, informal corrections are superior to procedural tomes (SOPs/WIs/etc.) on what is expected and allowed, IF this gets the results.
6) If documentation is required, make parsimony of information a by-word of the documents, put no less and no

---

[5] See the discussion on Process Descriptions and State Descriptions above.

Paper 175-2010

Quality Outcomes and SAS: Structuring, Managing and Working in SAS®-Based Programming Groups

Joe Perry, Perry & Associates Consulting, Inc. Oceanside, CA

more specifications in the documents than is needed to obtain the desired results;

7) Formally spot check the steps in the procedural documentation, including compliance with these procedures while they are being used;
8) Revise your SOPs on SOPs to require a dynamic review interval rather than the standard 2-year mandatory review requirement, especially in the initial implementation window for the SOP[6];
9) Change the SOPs on SOP to ensure measures of process efficacy have been enumerated and will be reviewed on a regular basis;
10) Measure quality outcomes, especially missed quality issues at each meaningful stage of the work.
11) Internally measure, really measure, the efficacy of the procedural documentation against objective measures of quality;
12) Ensure that a second, independent QC organization use quality audits to confirm and monitor your quality outcomes;
13) Cover programming styles during staff meetings until you get compliant outcomes and reinforce the reasons for, and mandatory nature of, their use and, if necessary,
14) Follow-up all training with a judicious use of spot checks on work performed and individually correct and you will get *more* consistency in departmental personnel's programming styles.

CHANGING YOUR ORGANIZATION

Nine years of facilitating client efforts to improve quality outcomes within Boeing Commercial Airplanes Company (both programming and otherwise) suggests that if you want to change your own quality outcomes, these the owners or tasks must:

1) Recognize that effective Quality Systems require a commitment from 100% of all organizational personnel and at all levels! This is not something that the workers nor suppliers must do without management, quality is a mindset that must permeate the organization and decisions at every level of your organization must reflect this quality focus;
2) Get a commitment for the full cost of the program, support the request for support with a profound analysis of the changes to your system and the resultant, detailed, improvements to your cost structure;
3) Articulate and become intensely focused on the quality objective, experience shows that quality is not for the faint-hearted as it requires the maximal energy to reach and maintain[7]… energy that, fortunately, is more than offset by the enhanced results you'll obtain;
4) Expect to TRAIN, TRAIN, TRAIN… ENCOURAGE, ENCOURGE, ENCOURAGE and CONTRONT[8], CONFRONT, CONFRONT your employees and yourself as you work through the implementation of a new system of thinking;
5) If, on your own, you can NOT create a clear, concise statement of the new organization with a quality focus, both in its structure and how it will operate in the final end-state and, importantly, find the ardent desire for that envisioned state, then, based on the TQS system articulated by Dr. Feigenbaum, you should enlist the aid of a quality professional that can help facilitate the desired change;
6) Find within yourself that 'Quality-Whisperer[v]' that will know when to do what, when, find that and you will be able to continue the improvement of your quality outcomes.

CONCLUSIONS

Achieving Quality outcomes CAN be achieved in the SAS programming world but, like all quality programs implementations, you must know what must be fixed and how to implement efficacious changes to your system and organization to achieve those goals. The devil is in the details but, thousands or millions of organizations have implemented effective quality systems in every industry imaginable so, if you are interested I believe you now have enough knowledge to start the journey…

IF... you have the ardent desire for quality, my vote is to… GO FOR IT! But, hey, I'm not you therefore,

You choose, it is only your (work) life.

---

[6] In the 70s and 80s, the premium Japanese manufacturing companies considered procedural documents older than 6 months to be outdated and ineffective; they likely still do!

[7] It is said that if this were easy everyone would have already have implemented outstanding quality processes, the experience of this quality professional indicates that it is only easy for the people that have or rise to, and stay at, the energy level required by your new master, quality minded-ness.

[8] Confront, as in, to face unflinchingly.

Paper 175-2010

Quality Outcomes and SAS:
Structuring, Managing and Working in SAS®-Based Programming Groups

Joe Perry, Perry & Associates Consulting, Inc. Oceanside, CA

APPENDIX I: OPPORTUNITIES FOR IMPROVING SAS-BASED QUALITY OUTCOMES

General Myths

1) The levels of supervision ‘professional’ programmers should have on their work should be less than the level of supervision placed upon other professions like medical doctors;
2) Professionals know how to perform their work because they are, just that, professionals;
3) SAS *programming* is different from *programming* in other languages like C++;

Poor Operational Controls

4) Over reliance on procedural controls to obtain compliance outcomes;
5) Heavy dependence on assumed knowledge to control work activity;
6) Poor procedural controls on supervisory and mentoring processes;
7) Over-specification of requirements or meta-data;
8) The above leads to… indeterminate specifications for products or services;
9) Un-enforced departmental policies, procedures and/or work practices;
10) Departmental personnel lack deference to the needed departmental procedural quality controls;
11) Discontinuities in procedural documents;
12) Lack of programming project plans;
13) Inadequate focus on, and resourcing of, supervisory personnel to support leadership and mentoring;

The Use of Passive Control Methods, Especially in the Choice of the Tone of Communication

14) Weak procedural requirements lacking operational verification mechanisms;
15) Passive voice used in the definition of procedural and product controls;
16) Circular definitions of critical activities;

Ill-Defined Quality Structures and Systems

17) Inadequate internal resourcing to perform work, both supervisory and programming related, this is related to the conflating of project and schedule management[vi];
18) Disengaged *company* Quality Control departments:
    a. QC departments typically have no senior-level SAS qualified resources;
    b. QC department’s undue deference to *organizational* quality systems (many QC departments see their role as ensuring procedural documents exist and training is accomplished per the schedule in a training matrix);
    c. No objective statistical programming quality related measures have been defined;
19) Definition of ‘auditable’ QC documentation is poorly understood;
20) Associated with the above, un-auditable quality control documentation is the norm and hinders process improvement efforts;
21) No check- and control- point quality measures are defined throughout the SAS related work processes;
22) There is a lack of separation of departmental programming and quality control functions;

Inconsistent Skills Development

23) Low priorities placed on skills and career development;
24) Few SAS personnel can write state-based descriptions of requirement and consequently resort to process-based descriptions of programmer requirements, this condition reduces assumptions of independence of the QC activities, including programming to a hope rather than a verifiable assumption[vii].
25) Lack of independence of programming activity confounds mimicry with skills attainment, i.e. there is a lot of copying of programs and, hence, mimicking of other programmer’s work rather than the acquiring the skills and ability to create the program from scratch;
26) Low awareness and use of software industry applicable Good Programming Practices during SAS programming.

Paper 175-2010

Quality Outcomes and SAS: Structuring, Managing and Working in SAS®-Based Programming Groups

Joe Perry, Perry & Associates Consulting, Inc. Oceanside, CA

The author can be contacted at:

Joe Perry
Perry & Associates Consulting
342 Spring Canyon Way
Oceanside, CA 92057
Phone: 760-822-7959
E-mail: PerryNAssociates@cox.net

---

[i] Starting on Page 211

[ii] Generally, process statements devised by *any one of us* have confounded both the end-state and process actions, actions that suffer from two ailments. First, to understand what the goal of a process statement, translating the Frame of Reference of the process used by the author must be made into the Frame of Reference of the recipient, a translation that is subject to subtle disconnections and misunderstandings; and second, the process description may, in fact, NOT achieve the same end-state as thought by the author.

This topic requires a more thorough discussion that is beyond the scope of this paper but this issue can be seen in the difference between SAS SQL statements vs. SAS Data and Procedural programming, though both programs, process statements framed with SQL primitives do not always translate well to the other SAS paradigms.

[iii] *Total Quality Control (1951) by* Armand V. Feigenbaum, CEO of General Systems Co. in Pittsfield, MA

[iv] This will be familiar to anyone working in a tightly regulated environment like drug development or the banking industries. The basic rule is that the operator can NOT repudiate that the work was performed as stated.

[v] See also the movie 'Horse Whisperer' with Robert Redford.

[vi] See http://en.wikipedia.org/wiki/Project_management for a fuller discussion on this topic but an extract follows.

Project Management is the discipline of planning, organizing and managing resources to bring about the successful completion of specific project's goals and objectives. Closely related to, and sometimes conflated with, schedule management, a distinguishing characteristic of Project Management, in the Operations Research sense of the term, is that **PM is an optimization process**. The **primary challenge of Project Management is to achieve all of the project's goals and objectives while honoring the preconceived project constraints**. Typical constraints are scope, time and budget but skills-sets and resource availability play a large factor in the feasibility of an overall project plan.

[vii] To illustrate the point that using process description rather state descriptions has problems we use illustrate with programming in SAS and SAS/SQL, say, you describe the process, as an SQL programmer would write it.

Taking an SQL query (especially the process for a complex query requiring multiple merges), the traditionally trained SAS programmer would probably NOT understand, and be able to recreate, the final state requirement for *their* code. If on the other hand, the description of the process was a parsimonious statement of the final end-state requirements, two salubrious effects will occur. First, either programmer is free to write code in whatever programming modality they are most comfortable with and/or secondly, IF there are differences in outcomes (nuanced or not) between the two programming tasks then, the detected difference(s) can be used to update the common control document (i.e. the specifications), all without violating our assumption of independence.

Additionally, writing end-state requirements in terms of process descriptions, though common in the programming world, tends to over-specify what is wanted. Since how to perform a task is, generally, beyond the scope of a specifications or requirements document there can only be a few outcomes from this method of specification. This is especially clear if one meant the specification for use as an RFP, sent to multiple vendors, the more process oriented the specifications contain the surer it is that the vendor has been pre-selected and the RFP is rigged. The more end-state oriented you get both creative solutions to your requests as well as true independence of decisions as the specifications do not subtly include statements that favor one software package over all others.