MWSUG 2016 - Paper SY03

Enterprise Architecture for Analytics Using TOGAF

David Corliss, Ford Motor Company, Dearborn, MI

ABSTRACT

Enterprise Architecture (EA) is a set of practices for development and implementation of the overall design of a system. EA embraces hardware, software and analytics in a single, enterprise-wide environment. As the leading end-to-end analytic solution for large enterprises, SAS® systems can benefit greatly from the use of EA best practices. As a framework for enterprise architecture, TOGAF supports creation of a set of inter-connected building blocks. When applied to analytic systems, TOGAF practices promote greater efficiency, improved cost and ease of use, streamlining the delivery all desired analytic requirements. This presentation describes the principles of Enterprise Architecture and goes step-by-step through the TOGAF framework as applied to analytic architecture. Practical examples are given of both the process and benefits of the TOGAF framework in a variety of industries and settings.

INTRODUCTION

A central principle of Enterprise Architecture is the bringing together of business needs with the technical resources required to address them. This includes architecture practices relating to business requirements, hardware and software, data and information systems, and analytics. The establishment of an analytics architecture practice facilitates the development of robust and well-supported analytics operations that are seamlessly integrated with other system components such as data, software and hardware – often a point of failure in large analytic systems. Enterprise architecture for analytics helps assure that the completed project fully meets the analytic requirements of the business or organization requesting the work.

Enterprise architecture is not a new kind of project planning or system for resource management. These are used as part of Enterprise Architecture but the focus in EA is different. EA works by closely aligning business needs with required technical resources and processes.

ARCHITECTURE FRAMEWORKS

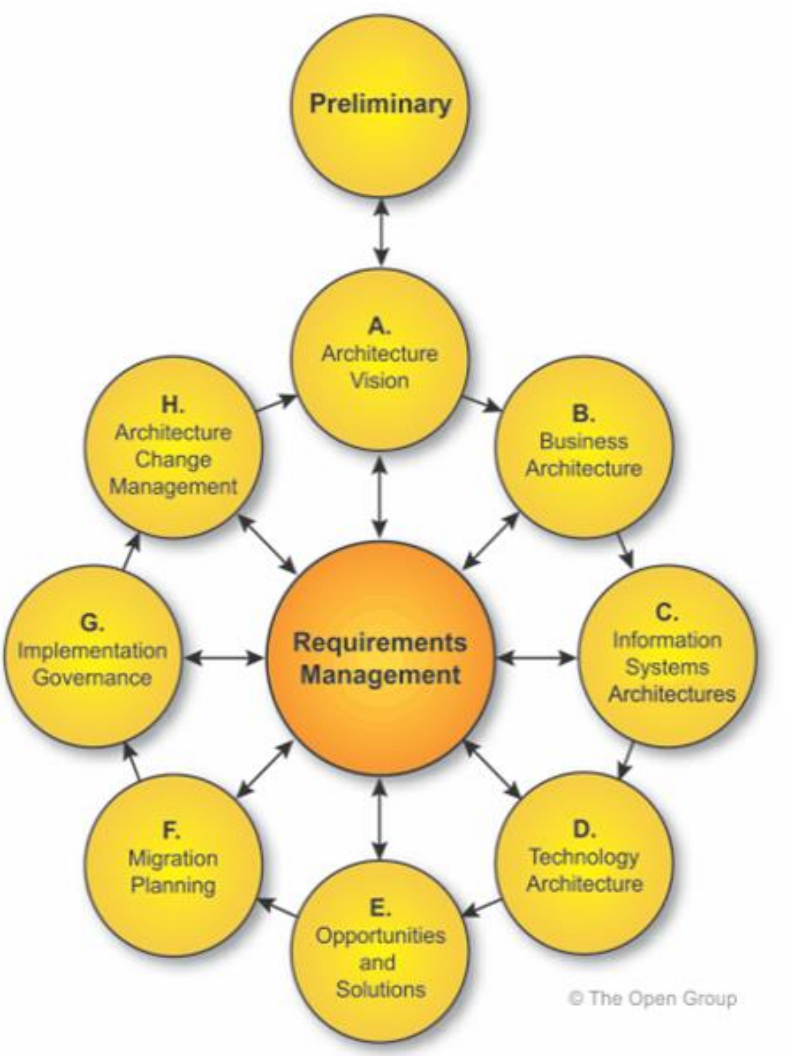
There are a number of architectural frameworks, facilitating the application of Enterprise Architecture to develop new systems. Each applies a set of principles to the description of the system to be developed, facilitating closer alignment of what the system is supposed to do and the resources used to create, test and operate it. Most architectural frameworks used today comprise an *ontology*, which is to say they define the objects to be created without specifying the details of how they are created.

One common enterprise architecture framework is the Zachman system, developed maintained by John Zachman who first created the concepts at IBM, beginning in 1984. This ontology is diagramed as a grid or matrix, with fundamental questions (who, what etc.) about a system in the columns and types of objects (physical, logical, conceptual etc.) in the rows. For example, at the intersection of “What” and “Logical” in the Zachman framework is the Data Model. The Zachman Framework simply defines the objects required. It is not a methodology and there is no order or priority given to the objects defined in the matrix.

In contrast to Zachman, the TOGAF Enterprise architecture framework defines a methodology in addition to the specific objects to be created. TOGAF is an acronym of The Open Group Architectural Framework. The Open Group is a consortium of more than 500 organizations often involved in creating, using, or supporting UNIX platforms. TOGAF, now in release 9.1, began development in 1995 and continues to be maintained by The Open Group. TOGAF is cyclical and iterative, with each component of “Phase” set in a specific order to facilitate system development. In this way, the Zachman architectural framework is an ontological *taxonomy* – classifying but not ordering the system components it defines – while TOGAF is an ontological *methodology*, imposing the outline of a process to develop to objects it defines.

TOGAF is best known for its Architectural Development Method (ADM), the cyclical and iterative process it applies to the development of complex systems. TOGAF components are usually described in a modified circle, manifesting the step-by-step and repetitive nature of the methodology.

TOGAF Architectural Development Method (ADM)



An important way in which TOGAF differs from the Zachman is this cyclical process. The architecture implemented at the end of the process is assessed against requirements developed in earlier stages. This interaction between different phases, the ability to iterate phases as needed, and a constant referral back to the basic business needs of the customer or business team requesting the work, which are established at the beginning of the process, constitute important strengths of the TOGAF methodology.

TOGAF FOR ANALYTICS - STEP BY STEP

The TOGAF framework can seem very abstract without substantive examples. This has been complicated by a paucity of thoroughly worked-out examples in TOGAF literature – a point that has been the subject of some criticism of the TOGAF framework. The prime motivator of this paper is a desired to produce a detailed application of TOGAF for analytic use cases.

As an example, let us consider a hypothetical instance that, while fictional, is designed to represent the development of a complex, end-to-end solution for an analytic use case embracing a wide variety of requirements common in modern analytic platforms. The use case is this:

- Create a database, optimized for analytics, from all United States Decennial Census records now available only in print, not digital format
- Digitize the hard-copy content of the text using the best current technology
- Develop a relational database in a SAS server environment containing the census records in a single data table
- Create a second table with geocoding, matching each census record to a location
- Create a big data analytic platform using a SAS – GreenPlum – Hadoop stack
- Data storage will be in the Hadoop layer
- Analytics will be performed using SAS but the computations will occur in database, i.e., the Hadoop layer
- Analytics will include querying, visualization using SAS Visual Analytics, model development and implementation. Modeling capabilities must include advanced time series models available from SAS ETS.

This high-level statement of basic needs and guiding principles for the project, called the Architecture Vision, is produced during Phase A of the TOGAF Architectural Development Method.

PRELIMINARY PHASE: FRAMEWORK AND PRINCIPLES

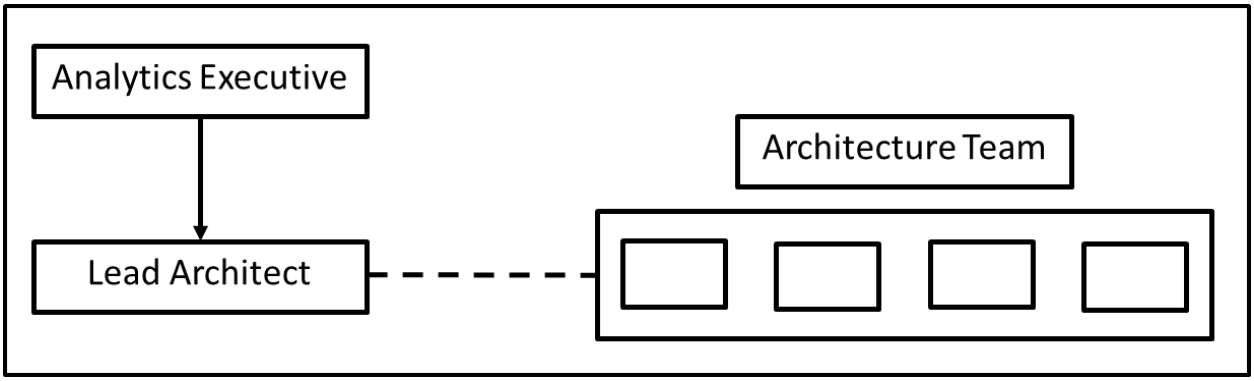
Before work can begin on the use case – or even to define it in very concrete terms – TOGAF methodology begins with establishing an architecture team. Team members unfamiliar with enterprise architecture principles in general or TOGAF in particular should receive training. This group determines how they will go about using the TOGAF framework. TOGAF methodology is an "open framework". In practice this means that it is flexible, supporting different teams with different needs, use cases and contexts. Rather than give an exact, inflexible set of steps in a "one-size-fits-all" manner, TOGAF is designed to be adapted by on organization's enterprise architects to fit their own needs best. Thus, the first step is to decide how TOGAF will be applied to architecture development at the organization. In this Preliminary stage, the architects will want to establish a document repository to contain standards, architectural principles and corporate processes under which architecture will be conducted.

Deliverables for the Preliminary Phase can include:

- A statement describing the organization's use and scope of Enterprise Architecture, including use of TOGAF methodology
- TOGAF documentation and training materials
- Organization of an Enterprise Architecture team and a description of the roles of its members
- Documentation standards for EA projects, including storage and access of project documentation

In our example, our Census Analytics Architecture Team or CAAT elects to have a single Lead Architect charged with applying TOGAF methodology to the development of the census data and analytics platform. This individual is supported by a committee acting on a consulting basis. It is agreed that when a project begins that requires enterprise architecture support, the Lead architect is responsible for hands-on architectural tasks while the CAAT will meet weekly. The Lead Architect creates a document repository for architectural principles, standards and accepted practices. All of this is done before the group is available to receive project requests. In this example, one the documents created by the Lead Architect is an organizational chart like this one:

A Preliminary Phase Deliverable: An Organization Chart for Enterprise Architecture Projects



It is essential to understand that *TOGAF does not require this organizational structure or these steps*! This is one possible solution under TOGAF principles. This particular organizational strategy likely would work well only in a fairly small organization, with only one Lead Architect. The organization could just as well have decided to have a team of architects, each assigned a different role – for example: one for data, one for hardware and infrastructure, and one for analytics methods and practices. The essential thing to keep in mind is that TOGAF principles restrict its use to informing what tasks are to be done without specifying how tasks must be performed. In the Preliminary stage, the team needs to agree on how decision will be made, establish standards and secure resources for the project. TOGAF states these as tasks to be performed without specifying how to do them. The organization's architects need to determine for themselves what is best for their own context.

PHASE A: ARCHITECTURAL VISION

The next phase of architectural development, Architecture Vision, is labeled Phase A with the previous phase regarded as preliminary. It begins with receiving a request to work on project. It is strongly recommended that this request be made in writing and that all activities, tasks and assignments included in a project plan document that is maintained for the duration of the project. A project-level repository should be created to contain documents specific to the particular project.

In this phase, the enterprise architect establishes the scope of the architecture project. Stakeholders are consulted to develop a high level understand their business needs. This is used to develop an Architectural Vision statement, documenting the overall principles to be applied in developing the project the architecture will support. The Architectural Vision is presented to the organization's management, who must sign-off on the project and agree to allocate appropriate resources before the project can go forward.

Deliverables for Phase A can include:

- A Request for Work, to be filled out by the person or group requesting the project
- An Architectural Statement of Work - a high-level statement of the architectural vision, including the overall scope and guiding principles for the project and providing a high-level description of the architectural work to be performed
- Customer Stakeholders: a short list of leaders representing the needs of the teams that will use the SAS analytic environment and providing business direction
- Project team leaders: a list of people responsible for different parts of the project who have agreed to contribute time and manage required resources. Representation could include project management, purchasing and contracts, hardware support, analytic software engineering, data architecture to assure information storage and data flow meets the business requirements, and analytics architecture to assure the analytic functionality of the completed platform adequately support the use cases of the Customer Stakeholders. If appropriate – especially in regulatory environments and / or when dealing with very secure personal data such as medical or financial information, a lead legal representative may be advisable.
- Architecture Object Repository

- While the Lead Architect will not produce a detailed time line – this will be done later by Project Management if the project receives go-ahead from management – an overall understanding of project scope will include a general idea of how long it will take to complete.
- Documented sign-off at the executive level for the project to go forward – this is essential.

In our example project, converting hard-copy historical Census records into a SAS analytic database, our Lead Architect starts by meeting with executives to understand their high level goals and objectives. Management designates both customer stakeholders and the team leads for different aspects of the project, one for the digitization effort, one for data management, one for analytics etc. The Lead Architect meets with the team to define the roles and responsibilities of each team, understand what each contributes and what they will need. She then drafts a Project Charter, including a high level description of the project scope, the guiding principles, and the project team. The Lead Architect reviews the materials with the Architecture Team to edit and confirm the content. The Project Charter is presented to executive management. Approval from management to proceed, including allocation of funding and other resources, is required before moving forward.

PHASE B: BUSINESS ARCHITECTURE

At its core, Enterprise Architecture is the art and science of bringing together customer requirements with technical solutions, assuring what is created by technical teams delivers a platform that meets all the needs of the customer - internal or external to the enterprise – requesting the project. In TOGAF methodology, Phase B focuses on the business requirements of the system. This phase will document stakeholder needs and any concerns they have about the business objectives of any currently existing system the new one intended to improve or replace.

The Lead Architect uses this to document the business purpose and functions of the current platform, if any. In TOGAF terminology, this is called the *Baseline* Architecture. Next, material from interviewing stakeholders is used to create a description of what the new system should do to support the business. These business requirements for the new system are called the business *Target* Architecture. Comparison of the two provides a *Gap Analysis*, detailing the differences in the present day architecture and the requirements for the new system.

Deliverables for Phase B can include:

- Business Requirements. These often form a business section in an overall requirements document, including the results of the next two steps.
- A description of the Baseline Architecture
- A Gap Analysis, describing what the new system should be intended to do for the business that the old system was never intended to perform

In our example project, the Lead Architect interviews stakeholders to get a picture of both the existing system, which only access census records collected in a digital format, as well as the new business requirement to digitize older, hand-written census records and put all the records on the same SAS analytic platform. The gap analysis focuses on digitalization of the hand-written census records and also the deployment of new SAS tools released since the legacy system was designed and built. The business requirements identified in this process become one section of an overall requirements document providing a blueprint for the construction of the new system.

One weakness of standard TOGAF methodology is that there is no explicit facility or phase for analytics. This paper recommends analytics requirements be considered in each phase, integrated with tasks appropriate for each. It is noted that canonical TOGAF methodology, beginning with Version 9, now does this for security requirements. The same should be done in each phase for analytics. In our example project, it was decided in this this phase that the business requirements would be met by implementing new SAS analytic procedures released since the development of the legacy system. This practice will be repeated at each step, performing the tasks pertinent to the phase needed to support analytic functionality.

PHASE C: INFORMATION SYSTEM ARCHITECTURE

This phase addresses logical and physical data components of the system, determining what resources need to be included, how they should be organized and managed. The focus of Information Architecture phase is on the data elements themselves; software used to handle the data belongs to the next phase (Technical Architecture).

Deliverables for Phase C can include:

- A list of data tables and how they are connected, without going into the fine detail of a full Data Model
- A description of how each data component is needed to deliver specific business objectives stated in Phase B
- A description of the Baseline Architecture, that is, the logical and physical data components of the legacy system to be improved or replaced
- A Gap Analysis, describing what the new data elements and connections are needed to deliver the new system's business requirements
- A data flow diagram, describing how the data flows between logical elements of the system

In our example project, interviews of stakeholders by the Lead Architect identify the need to add digitized records while making minimal changes to how the records are managed after they are added to those in the legacy system. It is decided to retain the baseline architecture when adding the newly digitized records. It is determined that the deployment of new SAS analytic procedures released since the original system was created does not impact the data architecture. If the team had chosen to adopt SAS Visual Analytics and Visual Statistics, a LASR server would be required, resulting in changes to the data architecture. For our example project, new data and analytic procedures are required but not a new visualization tool.

The Gap Analysis identifies a single need: to acquire new records without changing how the records are stored, connected or managed. Not every review results in a change to the legacy system. While the same structure is retained, the architecture process has provided the opportunity to review the logical data elements and determine if any improvements are needed to deliver the business requirements. The question of additional capacity required to store and manage the records belongs to the next phase: Technical Architecture. The information system requirements identified in this process are added to the business requirements in the overall requirements document for the new system.

PHASE D: TECHNICAL ARCHITECTURE

This phase addresses logical software and hardware components of the system, determining what resources need to be included, how they should be organized and managed. As before, the focus remains on developing a blueprint to deliver the needs of the customer or business team requesting the new system. In the case of Technical Architecture, this phase first determines the software platform needed to deliver the required functionality. This, combined with the information system requirements from Phase C, are used to determine the hardware needed to support the system.

Deliverables for Phase D can include:

- A determination of which software tools will be used, without going into details of engineering or deployment
- Selecting the hardware platform
- Server arrangement and network structure
- Selection of middleware and network software required to deliver the functional business requirements of the system
- A technical network diagram, describing the servers, connected systems, and data flow between the physical objects

In our example project, interviews of stakeholders identify a preference to make minimal changes in the technical architecture required to support the new records. The hand-written records are to be processed externally by a vendor specializing in this task, who provides digital records in the same format as the existing records. The SAS software is upgraded to the latest version in order to acquire new procedures not available in the legacy system. The hardware configuration from the legacy system is retained, while additional storage capacity is added to accommodate the large number of additional records. The technical requirements identified in this process are added to the business and information system requirements to complete the overall requirements document for the new system.

PHASE E: OPPORTUNITIES AND SOLUTIONS

This phase address the means to deliver the requirements determined in earlier phases – business requirements in Phase B, Information System Requirements in Phase C, and software and hardware requirements in Phase D. Leveraging a single, comprehensive requirements document with sections for each TOGAF phase, a roadmap for the delivery of these requirements is scoped and designed. This is a high level description, not a detailed project plan, in keeping with the Enterprise Architecture principle as an ontology: defining what is to be built, not how to do it.

In designing the architecture roadmap, it may become clear that the delivery on some system components is best done in stages. In TOGAF terminology, these are called Transitional Architectures. These are often needed to deliver system components that are very large, difficult, take a long time, or present a significant impact for users. Examples of transitional architectures include:

- Upgrading SAS to a new version – a simple step that can be performed quickly – while developing a new hardware platform over time and then migrating users
- Implementing a SAS LASR server first in a sand box environment with access only to a select team of users to test and refine the system, taking time to train users on Visual Analytics and Visual Statistics which the LASR server to operate, and then rolling it out to the large SAS community
- Shifting from code-driven tools requiring programmers to point-and-click tools such as SAS Enterprise Miner by training and converting one team at a time, rather than trying to launch every team in the company or organization on a completely new way of creating an analytic products on the same day

Deliverables for Phase E can include:

- Architecture roadmap
- Transitional architectures, if needed - breaking up project development and delivery into stages
- Business Transformation Readiness Assessment

In our example project, the full architecture team meets to review all parts of the gap analysis and draft a delivery roadmap. It is decided that SAS Visual Analytics and Visual Statistics – and the LASR server they require – are beyond the scope of the current customer requirements but might be developed in a future project. The basic project deliverables are for a SAS version upgrade and additional capacity to be added to existing servers. The actual digitalization of the hard-written census records is outsourced to a company specializing in this process that will deliver the digitized records in a SAS dataset with the same data specifications and the legacy platform. With these considerations, it is decided that transitional architectures are not needed for this project. In terms of the ability of the business to adapt to planned system changes, the minimal impact to the end users - who will be able to access the new, digitized records on the same platform using the same tools – the business is considered ready to transition to the new system.

PHASE F: MIGRATION PLANNING

As construction on the system nears completion, TOGAF Phase F begins to plan the migration of migration of data, users and tasks to the new system. As a first step in this phase, the Architecture Roadmap should be reviewed and updated as needed in response to any issues arising during system

development and construction. The implementation and migration plan is detailed, describing how users and Baseline to the Target architecture.

Deliverables for Phase F can include:

- An updated Architecture Roadmap, if needed
- Implementation and Migration Plan, including a general timeline
- Sign-off from executive management to go forward with implementation
- Communication to all stakeholders about the plan, including important dates

In our example project, the full architecture team meets to review progress on the construction of the new system. Finding no open issues that could change the system architecture, it is determined no update to the Architecture Roadmap is needed. Some users have begun to ask for other SAS tools, including SAS Visual Analytics, to be included in the new platform and are told these will be considered for a new project once user feedback is received on actual usage of the new platform following deployment. The Lead Architect stresses the importance of carrying through approved plans, designating addition al requests for future projects once user feedback on the new system is received. The Lead Architect interviews stakeholders on the readiness and timing of different teams to migrate to the new system. In this application, additional capacity will be added to the SAS server over a weekend during a normal maintenance window. Once the new capacity is confirmed as operational, the SAS version software upgrade is implemented during a weekend maintenance window. Feedback is collected to determine if there are any issues following the software version upgrade and any outstanding issues are forwarded to the local SAS support team. Once there are no outstanding issues with the hardware expansion and software version upgrade, the new digitized census records are loaded to the existing SAS tables in the upgraded platform.

PHASE G: IMPLEMENTATION GOVERNANCE

Phase G focuses on oversight of the implantation process, providing assurance that the architecture implemented conforms to the intended Target architecture. Metrics assessing conformance need to be based on the original performance requirements developed in the Business, Information Systems, and Technical Architecture Phases. These can take the form of an audit assessing conformance of the final system with the requirements for the Target architecture. The Lead Architect is responsible to develop and perform this Compliance Assessment, reporting the results to the full Architecture Team for review. The team then develops a Compliance Summary documenting that what was originally requested was actually delivered and the business needs of the customer or organization requesting the new system have been fully met. This summary is presented to executive management, who must sign-off on it to complete the project.

Following implementation of a project, it often occurs that some users will realize they want additional items or functionality not mentioned when the requirements were first developed and finalized. These additional, out-of-scope requests should be treated as Change Request to the new system, which are addressed in the next Phase H: Change Management.

Deliverables for Phase G can include:

- Standards for Architecture Governance, describing how the conformance audit will be performed
- Conformance Assessment
- Executive Sign-off
- Initial Change Requests

In our example project, the lead architect borrows the auditing standards from a previous project and adapts them as needed. After review and input from the Architecture Team, she writes the Conformance Assessment based on the requirements for the Target architecture. The audit is performed by the Lead Architect when the new system is in beta testing. In additional to documenting complete conformance of

the new system with the Target architecture, a request is made to add SAS Visual Analytics and Visual Statistics. The Lead Architect writes this up as a Change Request, beginning the next phase in the ADM.

PHASE H: ARCHITECTURE CHANGE MANAGEMENT

This phase supports processes and governance for changes to the system after implementation is complete. For small changes, with limited impact on system architecture, operations or usage, a Change Request can be made. While TOGAF methodology should still be applied, the process is streamlined with very small tasks completed at each step. With small changes, architecture documents should be edited, not replaced. Change Management begins with a request from the business or organization owning the system. Change requests from third parties should be directed to system owners, who can decide if the change request will be submitted. An Impact Assessment may be performed to help decide if a request qualifies as a small request. Requests that require a significant revision of the system architecture, capabilities or usage should be regarded as a new project with full application of TOGAF methodology. A communications plan should be developed and implemented to keep stakeholders informed of the change at each major step.

Deliverables for Phase H can include:

- Capability assessment, describing the ability of the system and users to adapt to proposed changes
- Change Request Documents
- Impact Assessments
- Documentation of any revisions to system requirements
- Communications Plans

In our sample project, the Lead Architect receives two requests in the course of beta testing. One is for the addition of a single, new SAS procedure currently in beta release and not scheduled for general release for almost a year. An impact assessment is performed, where it is found SAS is willing, in this example case, to provide a code patch containing the new SAS procedure. No change to information systems, technical or analytics architecture is required; the only change to the business architecture is the addition of the functionality of the one new SAS procedure. The request is reviewed by the system owner, who approves this Change Request.

A second request is made to add the SAS Visual Analytics and Visual Statistics to the system. An impact assessment finds this requires very significant changes: The request represents a large expansion to the system capabilities stated in the business requirements and a considerable cost is involved. The requested tools require a SAS LASR server, which must be acquired, set up, and integrated with existing data systems. Analytic usage is expanded and significantly altered for those adopting the SAS VA / VS point-and-click environment. While the system owners agree to support the request, this is not a small change. It is decided to initiate a new project using TOGAF methodology applied to analytics to support the new request.

TOGAF FOR ANALYTICS - BEST PRACTICES AND GUIDING PRINCIPLES

- Maintain the basic Enterprise Architecture principle of bringing together technical resources with business requirements, always keeping clearly in focus the needs of the business or customer and how the new system adds value by meeting those needs
- Consider the application of Analytics Architecture at every phase of project development and implementation. Please note: this essential step is not mentioned in standard TPOGAF documentation and classes!
- Enterprise Architecture creates on ontology: defining what is to be built, not how to do it
- TOGAF methodology consists of guidelines and recommended practices – never exact requirements – and should be adapted fit the needs of the particular project and company and industry standards

- TOGAF methodology can be iterative, re-visiting earlier phases as needed to deliver business needs
- Document everything, maintaining documents in a single repository accessible to all architecture team members
- Executive acceptance and sign-off, including allocation of resources for the architecture project, is needed at the outset and confirmed at major milestones along the way

CONCLUSION

TOGAF is one of the leading Enterprise Architecture frameworks in use today. It supports the development of high quality systems through the close alignment of business needs with the technical resources used to develop complex systems. When applied to the development of SAS platforms, the TOGAF methodology facilitates faster, more efficient and cost-effective system development with fewer performance issues due to closer alignment of business needs with technical development. Since few detailed examples of TOGAF methodology have been found previously in publicly-available literature, especially for SAS applications, it is hoped this document will foster greater adoption of this methodology offering great advantages for SAS system development.

ACKNOWLEDGMENTS

TOGAF is a registered trademark of The Open Group in the United States and other countries.

CONTACT INFORMATION

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

David J Corliss
Ford Motor Company
dcorlis4@ford.com

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.