An Enterprise Architecture and Data quality framework

Jerome Capirossi - NATEA-Consulting – jerome@capirossi.org
http://capirossi.org,
Pascal Rabier – La Mutuelle Generale – prabier@lamutuellegeneral.fr

Abstract: Insurance industry undergoes major regulatory changes regarding risk management like Solvency II which require managing data quality. This paper reports an experience feedback about the development of an enterprise architecture and data quality framework suitable for Insurance Industry and COTS[1] environments. The framework, inspired by TOGAF 9.1, is tailored to provide systemic views of enterprise organization, systems and data and to develop joint governance for enterprise architecture and data quality. The paper describes development approach and framework components including metamodel, repository, data quality, tools and governance. it may stand as a proposal for a TOGAF data quality extension.

1. Introduction

Insurance industry sector is undergoing big transformations [1] due to changes in risk regulation approach required by:

- New risks associated with longer life and weather
- Concerns about Asset managers long term solvability
- Financial markets stretching out Insurance Business

One of the first releases has been Solvency II directive which led to transformations not as smooth as one may think since Insurance Industry is an old already well structured industry with a fair technical debt. In addition, Insurance risk management relies on data crunching often processed by several stakeholders: brokers, contract managers… That's why data quality is recognized as critical success factor for transparency, good reports and good decision taking.

A mutual insurance company specialized in health and savings has required to extend its capabilities to planning and executing transformations with efficiency. Their today's approach being mainly function by function, does not allow any systemic analyze nor business involvement in projects who only provides requirements and waits for solution. Data quality is also poor since most of systems are COTS bringing their own view on data, and being integrated mainly by data syn-

[1] Commercial off the shelf software

chronization which appears complex. No existing tools allow to manage data quality especially data controls execution.

2. Approach

The company decided to develop a data quality governance approach. The main idea was the following: in a high Information Technology consuming context, data quality governance without good architecture governance does not make any sense. So it has been decided to develop an architecture and data quality governance joint framework.

Following objectives were set to be enabled in target organization:

- deciding swiftly and efficiently what transformations to do taking in account data quality risks
- addressing efficiently system design questions included data quality risks
- standardizing data quality risk approach in accordance with operational risk approach
- mitigating Information Systems complexity and technical debt
- better controlling and managing subcontractors involvements

A project was launched to customize and roll-out a comprehensive enterprise architecture framework with an extension for data quality which would include strategic and operational levels for governance and development purposes. This framework inspired by TOGAF 9.1[2] would contain content metamodel, repository, some methodological tools and governance organization.

Because it links enterprise architecture and data quality objectives, this framework would be a candidate to apply as a data quality extension of TOGAF 9.1 content metamodel.

3. Metamodel

First step has been to agree on a metamodel allowing 2 modeling levels: macro level on enterprise extent, detailed level for project design. Macro level was intended to shape the boundaries where detailed level has to stay confined. It should be powerful but should remain understandable by stakeholders if we wish they use it. Then, it will be a good basis for governing the whole information system design and data quality.

Metamodel design was conducted in compliance with TOGAF 9.1 and took “Business Service” for pivotal concept.

We define some additional concepts by grouping basic ones. All “Data Entity” related and dedicated to a same topic, “Customer” for instance, were grouped into an “Enterprise Data Entity” concept. All “Business Service” related to a same “Enterprise Data Entity” were grouped into a “Business Service Block”. All “Business Service Block” related to a same business function were grouped into a “Business Service Quarter”. If some Business Services were not directly related to

an “Enterprise Data Entity” but associated to a business function, they were grouped with a “Business Service Quarter” governed by this business function.

Finally, we grouped “Business Service Quarter” into “Business Service Area” according to they are falling into following categories : “Operations” “Operations Support” “Business Intelligence”.

We agreed on a 1st model rule which was: only one “Business Service Block” is granted to update a given “Enterprise Data Entity”. This meant that if you need to have an up-to-date information from a given “Enterprise Data Entity”, you have to use a “Business Service” of its “Business Service Block”. Conversely, if you catch an event which requires updating an “Enterprise Data Entity”, you need to use a “Business Service” of its “Business Service Block”.

We created a dedicated “Business Service Area”, called “Master Data”, which contains all “Business Service Quarter” associated with “Enterprise Data Entity” which could be qualified as Master Data. They are “Customers”, “Products”, “Organization”, “Contracts”, “Persons”, “Partners”, and “Nomenclatures”. Such business services are called from most of systems.

Applications being mainly COTS, most of projects have to deal with integration. For that purpose, we define a pattern which states that each COTS would be viewed as a collection of application components which are grouped by “Business Service Block” called “Logical Application block”. Since it supports whole “Business Service Block”, it supports all business services and, basically, all data flows provided by the block. Then, main requirement for COTS integration is that interfaces realized by application components should provide data related to the “Enterprise Data Entity” associated with the “Business Service Block”.

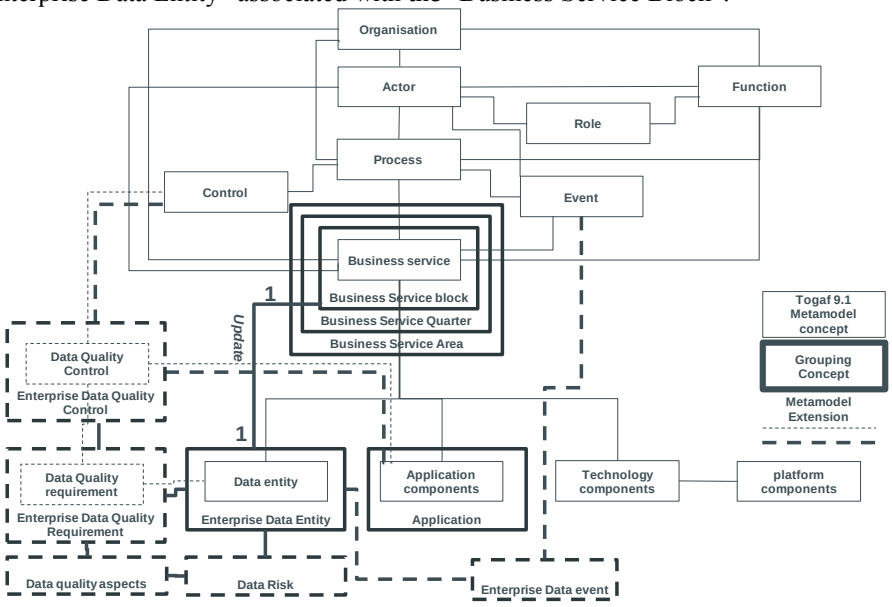

Fig 1 – Metamodel

As each data entity within an “Enterprise Data Entity” would be standardized, COTS would provide a standardized representation of enterprise data which would be aligned with enterprise point of view and not with COTS points of view. In addition, it has the advantage of keeping enterprise away from COTS changes often driven externally.

Same rule would apply to application components since they should get entity data from providers in a standardized form.

Once basic metamodel has been designed, we designed an extension dedicated to data quality management which is detailed in the following paragraph.

4. Data quality metamodel extension

This extension follows same multi-level principle. The lowest basic concept is “Data Quality requirement” which applies to a “Data entity”. It is associated to one or several “Data quality control” which allow to check if requirement is met by Data of the “Data Entity”.

Data quality requirement should met at best following rules :

- Unambiguously distinct from other requirements,
- It applies to a “Data Entity” or a specific part of “Data Entity”
- It provides value to the business in the mastery of risk of non-quality
- It applies to all instances of a business object or a subset. In this case it will specify the filter criteria (eg contracts whose event "contract")
- It must be tested and verified, it is subject to one or more controls
- It can be weighted with respect to the factor whether it provides fully or partially

Macro level contains “Enterprise Data Quality Requirement” which target all “Data Entity” of an “Enterprise Data Entity”. Each “Enterprise Data Quality Requirement” is linked with one or more “Enterprise Data Quality Control” which check that requirement is met at level of “Enterprise Data Entity”. In some case, “Enterprise Data Quality Control” may cascade some “Data Quality Control” defined at “Data Quality” level.

In metamodel, data entities are indirectly associated with business events which trigger business services. We added “Enterprise Data Event” concept which group business events directly linked with parts of a given “Enterprise Data Entity” into a kind of macro event. All “Enterprise Data Event” of a given “Enterprise Data Entity” constitutes its lifecycle.

Using the guidelines of International Association for Information and Data Quality[2], we defined data quality aspects [6] suitable for Mutual Insurance business. So, a given “Data Quality Requirement” addresses only one data quality as-

[2] http://iaidq.org

pect. A given Data quality aspect may be addressed by more than one “Data Quality Requirement”. Here after the list of aspects we agreed on:

Table 4.1 – Aspects of Data Quality

Aspect	**Description**
Accuracy	Data must be properly valued by business transactions.
Consistency	Any data must be consistent with other ones throughout all its life cycle
Uniqueness	No duplicates, either multiple instances of the same data identifier or different identifiers associated with the data.
Integrity	Any data should remain consistent with management rules: regarding mandatory and optional attributes, and other business rules.
Availability	Data must be available whenever user needs to access to it.
Traceability	Data changes are recorded in an audit trail enterprise facility
Completeness	Data must figure all instances of Entities
Compliance	Data representation is consistent with norms and standards of industry or in force in the enterprise
Freshness	Data must reflect the current state is no lag or delay
Intelligibility	Data characteristics and description of different states of life cycle must be obvious and easily understood by stakeholders.

For a given “Enterprise Data Entity”, a subset of aspects is selected in accordance with risks attached to data. Indeed, for being mitigated each risk requires some quality aspects to be matched by data. We identify the following risks and their associated aspects:

Table 4.2 – Links between quality risks and aspects

Data risk	**Aspect**
Management : bad operations and process supervision	Uniqueness consistency integrity traceability intelligibility
Contractual : bad bisness contract data	Accuracy uniqueness integrity availability traceability freshness

Data risk	**Aspect**
Operational : bad transactional data	Uniqueness consistency compliance availability freshness intelligibility
Intelligence : bad reporting and decision data	Uniqueness consistency integrity availability traceability completeness

	completeness
Financial : bad accounting and finance data	Accuracy consistency integrity availability traceability freshness completeness

	intelligibility
Regulatory : bad regulatory reporting data	Accuracy consistency compliance integrity availability traceability freshness completeness

A given "Enterprise Data Quality Control" is executed either by the "Application Component" which produce the dataflow, or other application components or by a business control associated with a process activity as we will see in the following paragraph.

5. Complementary architecture patterns

Since business service blocks are tightly linked with enterprise data entities, this metamodel enabled an architecture style which may be called data driven architecture.

In this respect, COTS are viewed as collections of business service blocks which match their functional modules. Pattern states that any COTS dataflow realize logically a remote business service call. An application component called interface executes business service operations and provides data corresponding to the business event. Then any dataflow results from a join of an "Enterprise Data Event" with a "Enterprise Data Entity".

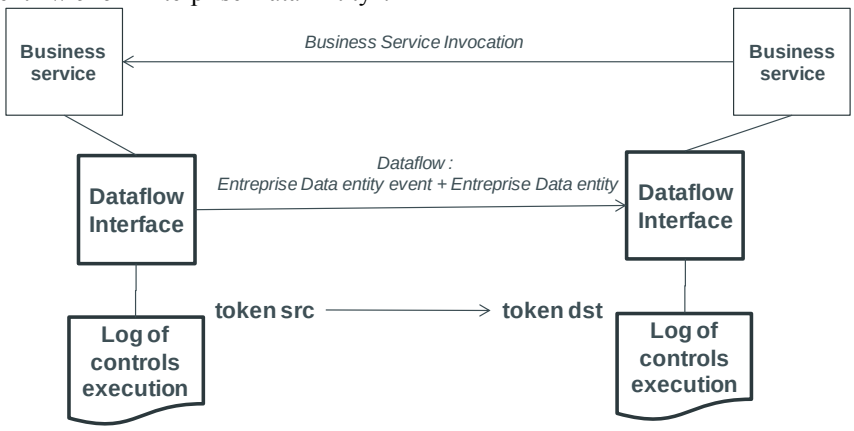

Fig 2 – Data flow reference Architecture

The application component acting as provider is responsible for executing "Enterprise Data Quality Control" and "Data Quality Control", it records execution

results in a log associated with an execution token. In this way, all logs could be consolidated to give a crossing view of data quality control results along a business process execution.

Token has been defined to identify any application component interface execution across information system. For enabling end-to-end traceability, token from source is recorded with destination token in consumer application component logging facility.

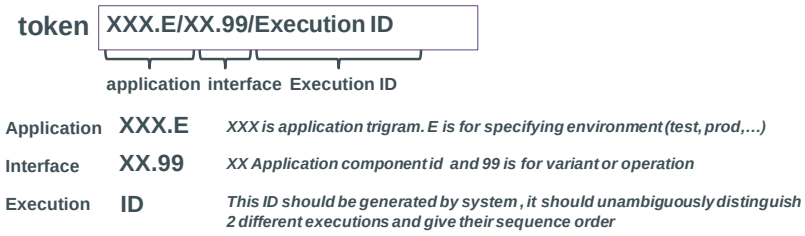

Fig 3 – Token definition

Business service blocks set in “Master Data” business service are realized either by application components of a given COTS which, for some of them, provide directly dataflows to other applications or by a special MDM component which aggregates data from several COTS to create all business services of the block.

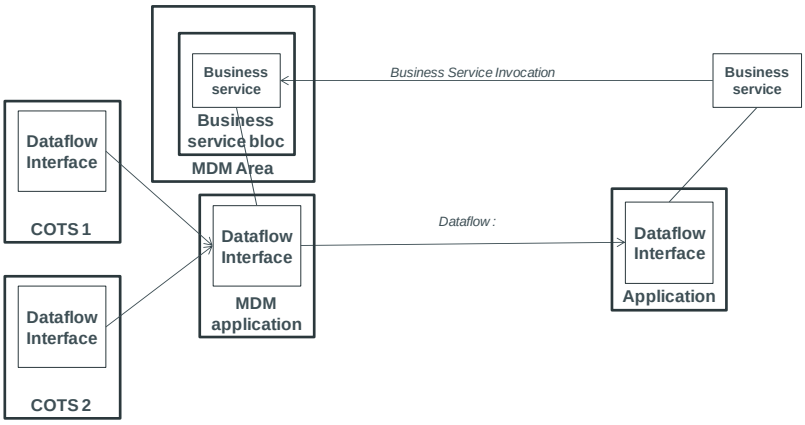

Fig 4 – MDM logical Pattern

In this case control execution and requirements are under the responsibility of the MDM Application components.

6. Tools

Once metamodel and some architecture patterns have been agreed, it has been time to describe architecture landscapes at different levels, macro levels for whole enterprise, detailed level for projects.

Then, enterprise architecture people defined macro level landscapes: process, organization, events, business services, business services blocks, enterprise data entity. These were intended to be used as boundaries and landmarks for projects specifying and developing solutions.

Macro landscapes were done with ARIS from Software AG, but other tools are suitable since they allow to spread architecture landscapes access across all projects stakeholders and architects.

For Data quality, we design a dedicated tool, "Enterprise Data Entity Dictionary" (EDED), which complements landscapes with special information regarding enterprise data entities. We pay attention to be compatible by ISO/IEC 11179 standard [3] related to metadata. Especially we include context information which allows to accept more than one data representation for a couple (event, enterprise data entity) according to the information system context. For example, you may need to have a comprehensive data representation for loading a datawarehouse and a short representation of the same data for a Business transaction.

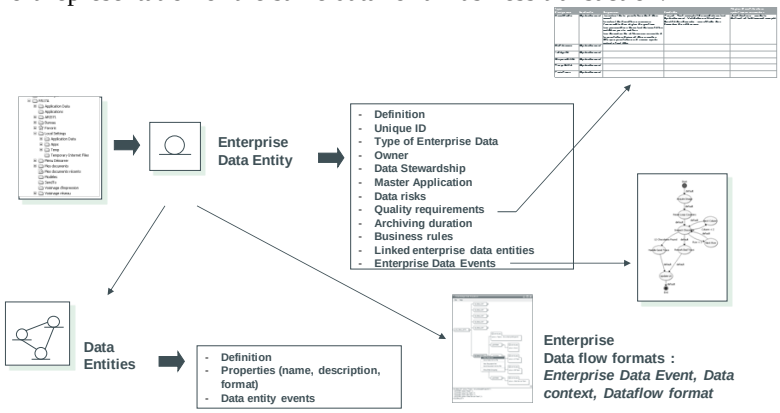

Fig 5 – "Enterprise Data Entity Dictionary" (EDED)

EDED only addresses macro level of architecture, detailed level is covered by regular data dictionaries which usually come with data systems and data modelling tools.

EDED is a repository all information needed by Architects and Data governance. It contains:

- Definition: a shared definition of the Enterprise Data Entity,
- ID: unique identifier by which data is recorded and plotted,
- Type of Enterprise Data Entity: Master data, operation data, intelligence data

- Linked Enterprise Data Entities: the relationships that bind to other Enterprise Data Entities,
- Owner: business owner accountable for quality
- Data stewardship: Business entities that manage operationally data under the guidance of the owner,
- Archiving Duration
- Data Risks
- Enterprise Data quality requirements: with aspects associated to data risks.
- Business Rules: constraints and business rules that must be complied global business object
- Enterprise Data quality events: by the form of finite state diagram

It contains also descriptions of all data entities which constitute "Enterprise Data Entity" under the form of an UML 2.0 class diagram and a table of public data flows formats ordered by event and context. A data flows format is deemed public when it is involved in a dataflow across different enterprise service blocks.

Technical data documentation and dataflows formats are developed by projects and stay under control of IT people. EDED provides only references to retrieve these documents, but does not store them since they are more related to systems than to Enterprise Data.

For Data governance purpose, we design an indicator [5] which could be calculated periodically and support a review by management. It is made of architecture maturity indicators which are viewed as levers for management and operations indicators

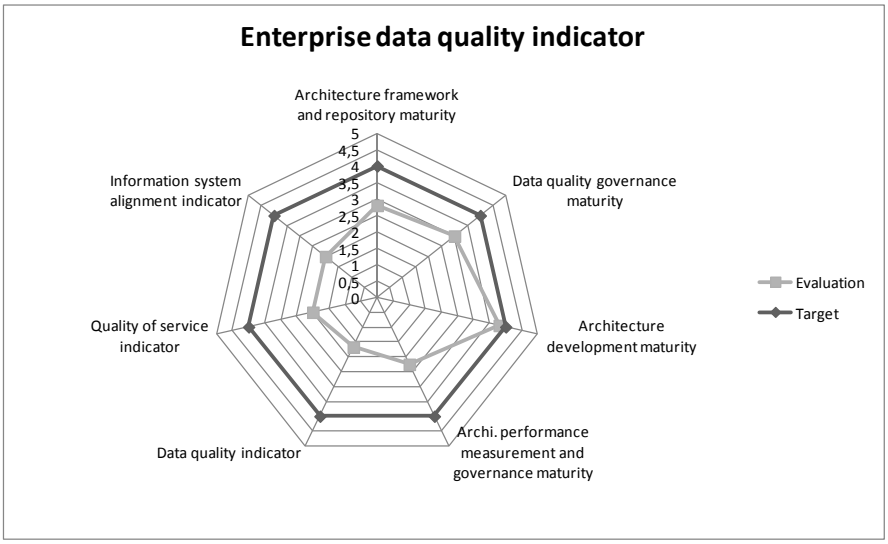

Fig 6 – Enterprise Architecture and Data Quality indicator

. Detail is figured below:

- Maturity of Architecture framework and repository: Maintain and spread processes maps, Business services maps, data and application architectures, patterns...
- Maturity of Data quality governance: completion of EDED, effective usage of EDED, performance of Data quality management processes
- Maturity of Architecture development: targets definition and roadmaps, projects scoping, architecture change management
- Maturity of performance measurement and architecture governance
- Consolidated Data quality indicator
- Consolidated quality of service indicator,
- Consolidated information systems alignment indicator to business inspired by COBIT 4.1 [4].

A more detailed version is used by Enterprise architects for managing activity. Then, each indicator is associated with a measurement process, some automated, some others requiring an expert assessment, others a survey.

Table 6.1 – detailed indicators of architecture and data quality performance

Grouped indicator	**Detailed indicators**
Maturity of Architecture framework and repository	Completion, compliance and communication of Business architecture maps : organization, processes, services Completion, compliance and communication of Data and Application architecture maps Completion, uptodate and communication of Architecture frame work : processes, pattern, references
Maturity of Data quality governance	Completion, compliance and communication of EDED Effective usage of EDED Performance of Data quality management processes
Maturity of Architecture development	Business and system target and roadmap Enterprise Architecture alignment with strategy Projects compliance with Enterprise Architecture Architecture Requirements Management Architecture change management process Architecture Competences Definition and sourcing
Maturity of performance measurement and architecture governance	Architecture Stakeholders management Architecture Board compliance and performance Project architecture compliance and performance Indicator calculation process compliance and performance
Data quality indicator	Control execution performance

	Control coverage : executed versus defined Incident due to lack of data quality
Quality of service indicator	% Compliance with Quality of service engagements for each Business domain
Information system alignment	Business alignment detailed indicators: • Compliance with rules regulating • User satisfaction • The IS is a factor of business productivity • The IS is a factor of business security • Frequent and serious incidents in production • Low confidence in IT to achieve projects or improve operations • Ability to integrate business constraints and / or exploit the opportunities of information systems Non functional alignment detailed indicators: • Information system agility • Information system testability • Information system maintainability • Information system easy operations • Information system security • Information system scalability • Information system documentation

7. Architecture and data governance

Architecture governance and Data governance rely on the same charter which states that among ultimate goals of Enterprise information systems there is processing data with high quality standards.
Architecture governance main body is the Architecture Board which is accountable for framework change management, architecture development, architecture performance measurement. Mostly, it has been inspired by TOGAF Architecture Board definition.

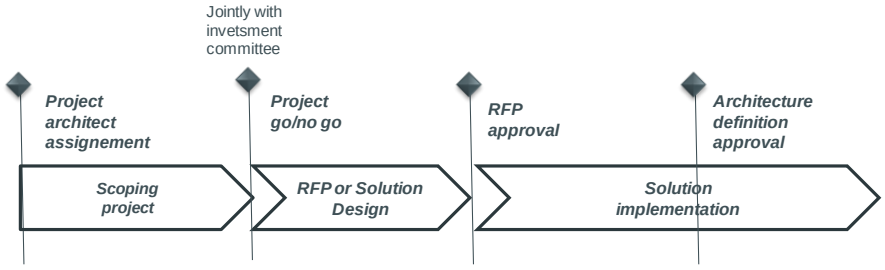

Fig 7 –Architecture Board milestones

It deals with data quality when overseeing architecture development and performance measurement. In architecture development, it approves project architecture definition which include data architecture and data quality requirements.
From a point of view of a given project, all architecture board milestones are undergone submitting the same "Architecture Definition Document" at different stages of development. For helping and supporting projects, we wrote comprehensive guidelines to fulfill the document in compliance with Enterprise architecture standards and objectives and consistent with the company macro architecture. These guidelines make easier integrators management when they are from outside the company.
In performance measurement, architecture board collaborates with data quality governance bodies which provide indicators dimensions directly related to operational data quality.
Data quality governance relies on datastewardship for operational data management and on Data manager for governance. Datastewards perform day to day business information operations, they are using systems and executing controls on data, those specified in EDED[3]. Data manager is a top manager of an enterprise function. He is at stake with data quality for achieving business process performance of business function. He review periodically Architecture and data quality performance indicator. In the case of problems, he requests either action plans to be executed by datatewards or sponsors data quality projects. Then, projects undergo Architecture governance milestone.

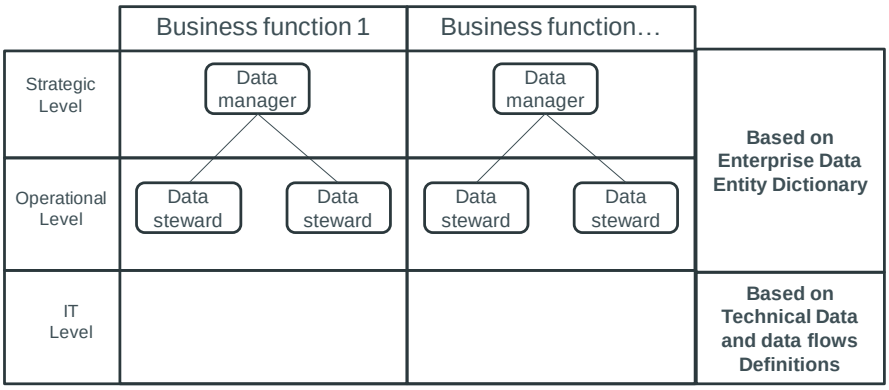

Fig 8 –Architecture Board milestones

The indicator is reviewed periodically by the enterprise board and may be linked with enterprise operational risk management.

[3] Enterprise Data Entity Dictionary

8. Conclusion

We rolled-out the framework, coaching projects for developing their Architecture Definition Document. The roll-out was progressive, starting with an experiment on 2 projects, which would be followed by a generalization.

Experimental phase was useful, not only to track implementation bugs, but also to let projects state that framework had simplified architecture definition work. As a result, they became keen to adopt it. Spreading such a message was a good advantage for framework global adoption which is still ongoing.

9. References

[1] KPMG (August 2012) Survey - Economic Capital Modeling in the Insurance Industry - http://www.kpmg.com/Ca/en/IssuesAndInsights/ArticlesPublications/Documents/economic-capital-modeling-july-2012.pdf

[2] Opengroup (2010) - TOGAF 9.1 - http://pubs.opengroup.org/architecture/togaf9-doc/arch/

[3] ISO/IEC 11179 - http://metadata-standards.org/

[4] ISACA (2007) - COBIT 4.1 - http://www.isaca.org/

[5] Steve Thorn (March 2012) – When was your las enterprise architecture maturity assessment – Open group Blog - http://blog.opengroup.org/2012/03/02/when-was-your-last-enterprise-architecture-maturity-assessment/

[6] Wang, Richard Y; Strong, Diane M - Beyond accuracy: What data quality means to data consumers - Journal of Management Information Systems; Spring 1996; 12, 4; ABI/INFORM Global