Building a case for an Agile Enterprise Architecture Process – Part 2

by Charles Edwards. 4th February 2007.
Version 1.00

Abstract

This series of papers considers the case for enhancing the good work the Open Group contributors have already produced in TOGAF 8.1 [1], by defining an lean and mean Enterprise Architecture (EA) Practice Process that can be picked up and used with minimal tailoring to get an EA Practice started quickly.

Enterprise Architects starting a new practice say "There is so much to do and so much complexity. Where do we start? How do we move forward in a structured and well defined way, but still add value to the organisation quickly?"

The goal of this second paper is to suggest using existing standards to drive out solutions to the problems EA Practices face (as posed in the previous Part 1 paper). To suggest a set of standards upon which an Agile Enterprise Architecture Process could be more specifically defined. Processes to help Enterprise Architects get started quickly and easily. Enhancements to the TOGAF process are to be based primarily on the newer and beneficial thinking Agile and Adaptive [10] principles, but on other influences as well.

Context of Enterprise Architecture within the organization

As the name implies, Enterprise Architecture looks after the Architecture for the whole Enterprise. It looks at the big picture, not just a single project view. Enterprise Architecture includes Business Architecture, Information Services Architecture and Technology Infrastructure Architecture.

In an enterprise, any extremely successful Project (development or COTS integration) could add or change some sub-set of all the business systems, software and technology architecture in the enterprise. See diagram Figure 1 - Enterprise Architecture for a simple representation of this concept.
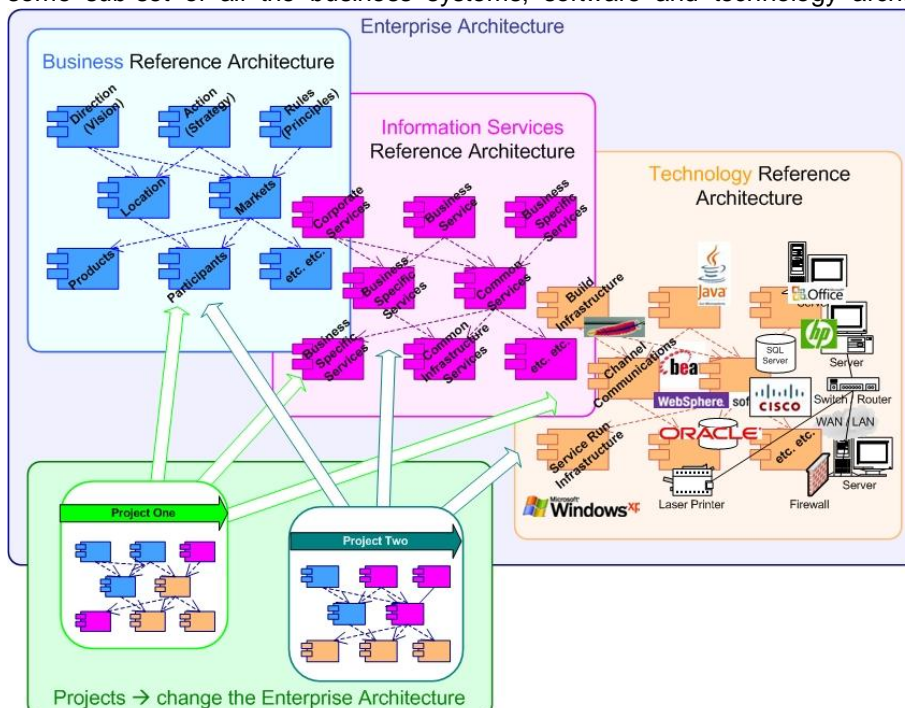
Even though any one Project may deliver the best and most reliable Product set for the particular purpose, it may harm the overall Enterprise Architecture.

For example a new project may conflict with some existing business processes, add a new vendor database product into the mix or add another application server instead of using the existing one. This could turn out to be very costly to support and manage in the longer term.

Figure 1 - Enterprise Architecture within the organization

Problems being addressed

The first paper targeted some of the items below as issues to be addressed by implementing an Agile Enterprise Architecture Process:

1. **An Adaptive versus a Predictive approach** – The potential problem here is that Enterprise Architects use a large plan up front, rigid process (Predictive), instead of a time-boxed, continuous, risk-driven improvement process (Adaptive) to manage the EA practice and its output.
2. **Duality of process** – **Separate Cycles of EA and Projects** – Overcoming the potential problem of slowing down the Project process (i.e. making it non-Agile) just to consider all the views EA require.
3. **TOGAF ADM Phases, Cycles and Iterations** – Definitions are confusing and need to be clearer.
4. **Traceability between EA and Software development** – Closely related to 2 above is the Problem of multiple Projects potentially duplicating work and effort, by using Project time to build up an understanding about what already exists, exactly what will be impacted, external dependencies, etc. If traceability exists from the EA to the Projects, then this waste of time could be minimised.
5. **Separation of Data and Function; learn from Object Orientation** (OO) – The problem is structuring EA as we did pre 90's before Object Orientation. E.g. categorizing Information into distinct layers such as Data and Application. There are valid Data Views across Business, Information and Technology. We need to drag Enterprise Architecture ~~kicking and screaming :^)~~ into the new century!
6. **Maturity of the Organisation impacting EA -** A potential problem to be resolved is to identify the level of maturity of the organization's Enterprise Architecture. Papers have been written on the stages of an organisation's Architectural maturity [23]. An EA team can only take the organisation to the next stage of maturity in single steps and no faster. The EA practice process should recognise the maturity stage of the organization and react accordingly.
7. **Maturity of the EA Practice** – An EA Practice needs to overcome the problem of not being effective because they are reactive or "fire-fighting" instead of being pro-active and organised. To be Adaptive and not Predictive, under control and in an organised manner.

New problems added in this paper for consideration are:

8. **EA Practices do not have well defined simple processes to follow**– Unlike *software development teams* in organisations that have ever increasingly well defined process instances (in web form) such as RUP, OpenUP, XP, Scrum; *Enterprise Architecture teams* tend to work from experience, use high level frameworks or use closed proprietary processes rather than using or tailoring process instances specific to the organisation in web form to guide them. Two open standards exist: TOGAFs ADM [1] gives high level guidance, but is not specific enough (intentionally). EUP [2] gives an excellent starting point but not easy to tailor.
9. **EA Practices should follow modern standards** – In general because other than TOGAF & EUP no formal open standards for EA exist, practitioners use their experience and knowledge from other EA methodologies to define their own in-house processes. This leads to; a) mixtures of terminology, b) inventing of new names for concepts, c) methodologies related to individuals preferences and experience - when they leave the new suitor changes it all changes to the next individual style. This is a wasteful process, using up time and money and offers organisations little real value.

Standards Influences on an Agile EA Process Solution

The suggestion is to harvest from standards that already exist in closely related spaces and pull them together in a synergistic way that makes sense in the Enterprise Architecture Process context.

Many influences on software development have validity in the concept of an Agile EA. Some of these are:

- **Meta-Model input** - Business Motivation Model (BMM), IEEE 1471 and SPEM.
- **Visual Modelling Rigor & Structure**- Agile Modeling principles, UML.
- **EA Framework Content & Structure** – TOGAF & Zachman and other frameworks**.**
- **Software development process structure** – Unified Process, Scrum & Agile.
- **General Content** - Risk & COBIT Compliance.

Others not included in this paper, which one could look at for adding value are BPMN, OASIS SOA, CMMI, etc.

Using a UML package diagram to show the information listed above, this is the current collection of packages of closely related information which could help assist defining an Agile Enterprise Architecture Process.
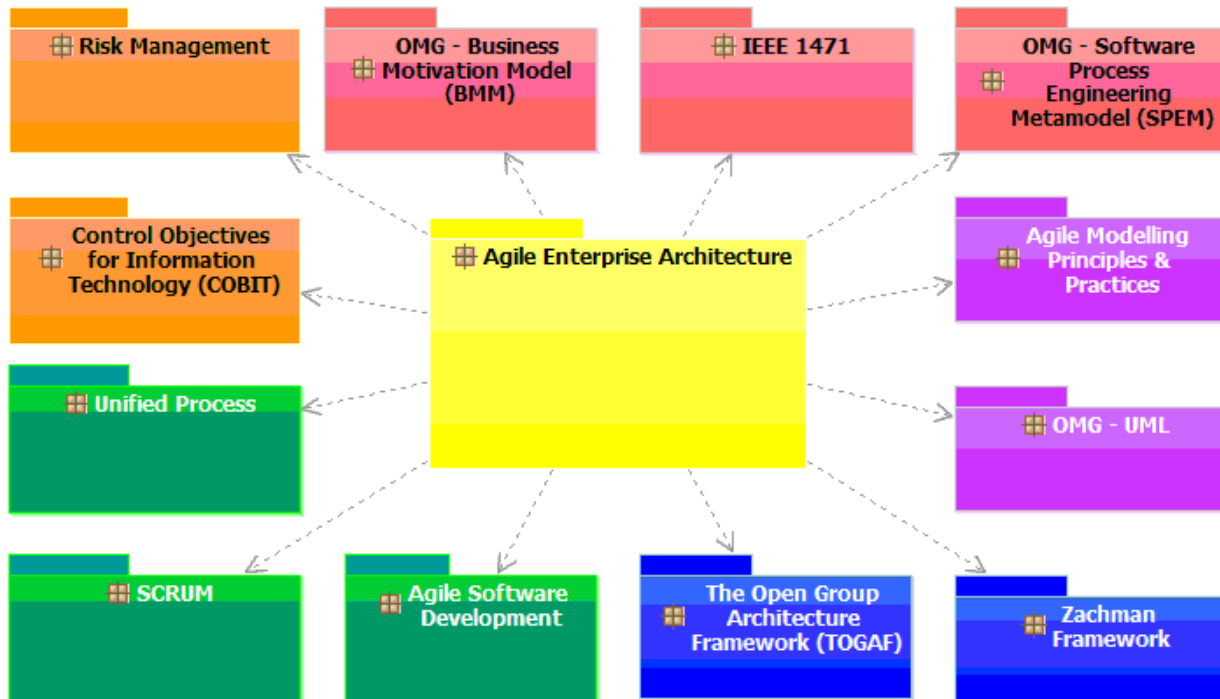

Figure 2 - Influences on an Agile Enterprise Architecture

Let us explore each of these concepts in a little more detail moving clockwise from the first red one at the top.

Meta-Model input

- **Business Motivation Model (BMM) - [10]**
  - o EA has always pushed the concept of driving both business and IT strategy from first principles. The BMM is a meta-model of how the concepts of Strategy, Vision, Goals, Objectives, Principles, Policy, Directives, Critical Success factors, etc. are defined and related. It is good to make use of well thought out and defined standards. Much wasted time in research and development of these concepts could be saved on EA Practices.

- **IEEE 1471 - [4], [5]**
  - o This is an Architectural meta-model that describes the main things Architecture is concerned about and how they relate and interoperate. The main concept this introduces to Agile EA is the Concept of Viewpoints and Views. Another concept that is central to defining EA and different views on similar items.

- **OMG – Software Process Engineering Meta-model (SPEM) – [6], [7]**
  - o A meta-model describing how to document process for software developers. In simple terms it deals with Packages of Roles, of Activities, of Work-Products. It takes both a static structure view and dynamic behaviour view. This is a sound basis upon which to describe a Process for an Agile Enterprise Architecture. SPEM is evolving quickly at the moment and will be a sound basis for not only modelling software development, but many more generic processes no doubt. It also gives a sound basis for naming conventions of concepts and their inter-relationships.

Visual Modelling Rigor & Structure

Enterprise Architecture practice has always been based upon the concept of visual modelling to convey complex concepts. To model wherever and whatever possible within the Enterprise.

- **Agile Modelling Principles and Practices - [11], [12]**
  - o There are a lot of excellent Agile Modeling Principles and Agile Modelling Practices that have their basis in Software development, that work equally well for Agile Enterprise Architecture. This particular area though is open to lots of heated discussion because in general EA traditionally see modelling the whole enterprise as the objective while typically Agile principles follow the concept of model only what is necessary. This is the subject of a paper all on it's own and will not be covered here, other than to say that they are actually not incompatible, but have a duality and both work together for their particular purposes.

- **Unified Modelling Language (UML) – [9], [13]**
  - o UML was derived out of a core need to model software and how to describe software development. The principles derived in the UML meta-model mean that that UML2 can easily be extended to define EA concepts. This can be done by define a standard UML Enterprise Architecture Profile. As an example one already exists for DODAF as a plug-in to the IBM RSA modelling tool.

EA Framework Content & Structure

- **Zachman – [20]**
  - o The Zachman Framework has been around for decades and is a good solid framework for partitioning out the Enterprise Architecture, but because is has not been open enough seems to have failed becoming the de-facto open standard for EA. It has however been extended and enhanced more recently and still appears wherever anyone talks about EA. In any event the concept of the six interrogatives of who, why, where, what, when and how are generic enough to bring into any definition of EA.

- **TOGAF Architecture Development Method (ADM) – [21]**
  - o An excellent collection of useful EA content (at version 8.1.1) and is in the process of being extended to bring in a lot more relevant information in version 9.0. It will form the core content of the Agile EA.

Software Development Process Structure & Content

- **Scrum – [18]**
  - o Scrum is one of those universal concepts that is well documented and kept extremely lean and mean. It has therefore become easy to learn, easy to practice and in so doing has spread quickly. It becomes the core of the iterative management process that will be used in the Agile EA. The Agile EA should keep as lean and mean as possible to be successful.

- **Unified Processes & Agile – [8], [19]**
  - o There are many flavours of the Unified Process. They are best described in two excellent articles, one called the *Agile Unified Process* [8] and the other called *New Methodology* [19]. All these flavours of Software development have relevance in the concept of the Agile EA, and many of their concepts will have input into the Agile EA.

General Content

- **Control Objectives for Information and related Technology (COBIT) – [14]**
  - o The Control Objectives for Information and related Technology (COBIT) is a set of best practices (framework) for information management created by the Information Systems Audit and Control Association (ISACA), and the IT Governance Institute (ITGI) in 1992. It provides managers, auditors, and IT users with a set of generally accepted measures, indicators, processes and best practices to assist them in maximizing the benefits derived through the

use of information technology and developing appropriate IT governance and control in a company. [14] While COBIT is not all encompassing – this is an excellent and rigorous start to identify which bits are relevant to EA and ensure they are built into the Agile EA process to ensure compliance.

- **Risk Management – [16], [22]**
    - o "Risk" is defined as any uncertainty that could affect achievement of objectives, including those that could hurt (*threats*) and those that could help (*opportunities*). This is an opportunity to use the Risk management process as an input and driver to the EA iterative content on a regular basis.

Solutions - defining an Agile EA by improving on TOGAF

The TOGAF and ADM component, while rich in information and full of very useful content, tends not to be of great help when you start an EA Practice and want to know what to do next. It lacks in Process structure at the detailed level (Activities, Tasks, Roles, Work Products) and the ability to easily access guidance. While TOGAF and the ADM is full of excellent content, it tends to appear heavy-weight and does not have a simpler starter alternative.

Although TOGAF representatives say not, the perception from the crop circle diagram in the ADM is a Serial process and does not easily describe a set of activities that could be happening in parallel within an EA Practice, such as understanding the Business, Information services and Technology all at the same time. Also Change Management and Implementation Governance happen continuously and not only once a project starts or once opportunities and solutions are defined. The naming convention of Phases is totally at odds with the SPEM definition of a Phase. There are quite a few of these anomalies that require further exploration, refinement and re-factoring to make this more of an off-the-shelf easy process to implement.

A good solution would be to define an Agile EA process with Phases, Iterations, Work Products, Roles and Activities, etc. to build upon all the good content work already in TOGAF, and bring it into a more SPEM like structure and make it a lot more Agile.

An Agile and cut down simple starter version of the TOGAF ADM would be an excellent starting point, from which more complex and detailed process definitions for EA could be extended.

Conclusion

Combining the Solution influences (and there are probably many more for consideration) to solve some of the Problem views it is possible to define an Adaptive and Agile (Iterative) approach to managing an Agile Enterprise Architecture Practice. It seems that the author is not alone in thinking that EA can co-exist with Agile. Cutter Consortium also thinks so. See [24]

As per the standards argument, the closer all the terminology and definition of detailed aspects such as Roles and Work products are related to the existing disciplines in IT, the easier the collaboration between EA and IT in general.

The suggestion is to use the Eclipse Process Framework (EPF) [17] Composer tool to do this; The Agile EA team would define a Web based set of process pages that would become a plug-in for any Organisation to use or to tailor to suit their own particular needs.

Our effort started in December and work currently resides on www.AgileEA.com. If you or anyone you know would be interested in collaborating on this definition of an Agile EA Process, then please get in touch with the author via our wiki http://agileea.wikidot.com/ and help establish an Agile EA Process plug-in that would be an open product freely downloadable from the EPF website.

References

[1] The Open Group; www.opengroup.org

[2] Scott Ambler, Enterprise Unified Process (EUP); http://www.enterpriseunifiedprocess.com/

[3] The Open Group; ISBN 193162456-9. "TOGAF 8.1. Enterprise Edition. The Open Group Architectural Framework."

[4] IEEE 1471 - http://en.wikipedia.org/wiki/IEEE_1471

[5] IEEE 1471 - http://www.enterprise-architecture.info/Images/Documents/IEEE%201471-2000.pdf

[6] SPEM - http://www.omg.org/technology/documents/modeling_spec_catalog.htm

[7] SPEM - http://www.omg.org/technology/documents/formal/spem.htm

[8] Alhir Sinan Si; The Agile Unified Process (AUP); http://home.comcast.net/~salhir; March 2005.

[9] UML - http://www.uml.org/

[10] Business Motivation Model http://www.omg.org/technology/documents/bms_spec_catalog.htm

[11] Ambler, Scott; Agile Modeling Principles; http://www.agilemodeling.com/principles.htm

[12] Ambler, Scott; Agile Modeling Practices; http://www.agilemodeling.com/practices.htm

[13] UML - http://en.wikipedia.org/wiki/Unified_Modeling_Language

[14] ISACA; COBIT www.isaca.org/cobit

[15] http://en.wikipedia.org/wiki/COBIT

[16] Hillson, David: www.risk-doctor.com

[17] The Eclipse Process Framework Composer; www.eclipse.org/epf

[18] Scrum; http://www.controlchaos.com/

[19] Fowler, Martin; The New Methodology http://www.martinfowler.com/articles/newMethodology.html (2005)

[20] Zachman Framework - http://www.zifa.com/framework.html & http://www.valuebasedmanagement.net/methods_zachman_enterprise_architecture.html

[21] TOGAF – ADM http://en.wikipedia.org/wiki/TOGAF#Architecture_Development_Method

[22] Risk Management Standard - http://www.theirm.org/publications/documents/Risk_Management_Standard_030820.pdf

[23] , Galen; The Four stages of Enterprise Architecture, Jan 2006 http://www.cio.com/archive/120106/fea_mit.html

[24] Jim Watson, Michael Rosen, and Kurt Guenther 2005 - Are Agile Methods and Enterprise Architecture Compatible? Yes, with Effort - http://www.cutter.com/project/fulltext/reports/2005/11/index.html