

Agile Enterprise Architecture – Part 1

by [Charles Edwards](#). 14th December 2006.

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. 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.

The goal of this first paper is to look at the Concern Viewpoints that an Agile EA would potentially solve. Future papers will look into the potential solutions options. The paper assumes a little knowledge of TOGAF.

Context of Enterprise Architecture

A lot of thought has gone into software development and software architecture disciplines over the last few decades, with the Agile community at the forefront and making excellent progress more recently. The same cannot be said for the discipline of Enterprise Architecture (EA), which had serious focus in the 80's & 90's, but besides input from Scott Ambler with Enterprise Unified Process (EUP) [2], was distracted by the rise of the internet and dot com era. The good news is that EA seems to be on the rise again encouraged by the Open Group Architecture Framework (TOGAF) [1] who have been working away for some time.

One could assume that because EA has been around for years it is probably perceived by many as old fashioned and outdated, possibly even offering diminished value. Quite the opposite is true through. In today's world many organisations are struggling with ever more complexity, cost, changes and compliance pressure. Organisations have inherited silo'ed applications, multiple heterogeneous environments (such as different operating systems, databases, app servers, languages, etc.), and are trapped between running the business as usual while simultaneously having to re-invent their Systems Architecture. Not a trivial challenge.

Enterprise Architecture Definition

Enterprise is defined by the TOGAF book [3] as:

“A collection of organizations that have a common set of goals and/or a bottom line. In that sense, an enterprise can be a government agency, a whole corporation, a division of a corporation, a single department, or a chain of geographically distant organisations linked together by common ownership.” It is made more complex by defining supply chains of partners, customers and suppliers.

Architecture is defined by IEEE 1471-2000 [3] as:

“The fundamental organisation of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution.”

Enterprise Architecture (referred to as EA) is in the simplest of terms by putting the above two together:

“The fundamental organisation of all enterprise¹ systems, the components and their relationships to each other and the environment and all principles governing the enterprise systems design and evolution.”

To resolve the complexity of inter-enterprises relationships, TOGAF has defined a broader vision: “Boundaryless Information Flow™ will enable access to integrated information, within and among enterprises, based on open standards and global interoperability.” [3]

¹ Whichever elements your specific enterprise may comprise of will define your enterprise in this context.

Context of Enterprise Architecture within the organization

As the name implies, Enterprise Architecture looks after the Architecture for the whole Enterprise. This 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 below for a representation of this concept.

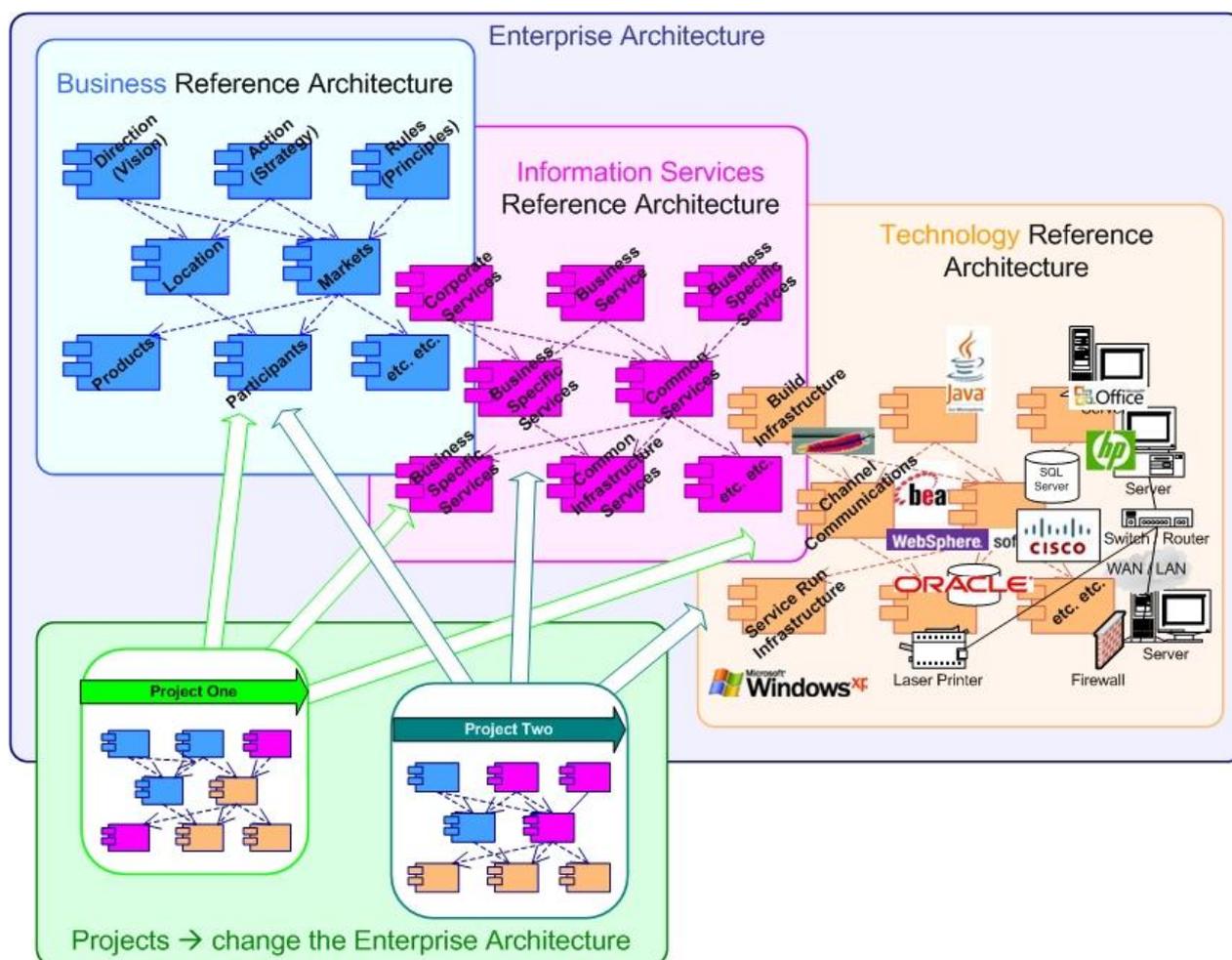


Figure 1 - Enterprise Architecture within the organization

Even though any 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 costly and difficult to support and manage in the longer term.

Problem views - What Concerns does Agile EA solve?

There are many different threads of concern influencing the need for an Agile EA. Concerns are looked at from various different viewpoints, but each view considers EA as a whole, and tries to resolve the particular problem view from that viewpoint. The problem views are first listed here in summary and then the paper tackles them one by one in more detail to build up the case for an Agile EA:

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. We explore this option by questioning whether the Agile & Adaptive Principles logic works for EA.

2. **Duality of process – Separate Cycles** – Overcoming the potential problem of slowing down the Project process (i.e. making it non-Agile) just to consider all the views EA require. The solution proposal is to split the lifecycles into two separate adaptive process cycles. One for Projects and a separate one for EA, both adaptive in manner, both using their own Phases and Iterations.
3. **TOGAF ADM Phases, Cycles and Iterations** – TOGAF is confusing and it needs to be clearer.
4. **Traceability between EA and Software development** – Closely related to the above is the Problem of multiple Projects potentially duplicating work and effort, by taking 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 OO. Categorizing some of it into distinct layers such as Information and Application. A solution would be to simply call it Information Services and have different views for Information, Applications, Data, Services, etc.
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 [9]. An EA team can only take the organisation to the next stage of maturity in single steps and no faster. A solution is to tailor and adopt an EA practice process that recognises the maturity stage of the organization and reacts 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.

1. An Adaptive versus a Predictive approach

Enterprise Architecture versus Agile Software Development

You might wonder; If Agile is about better and more efficient ways of developing software, why would we want to make Enterprise Architecture more Agile? After all EA is not directly involved with actually developing software (dependent up on the EA practice); EA is only really concerned about the big picture. Concerned about what already exists in the enterprise and what should exist once any one particular project is complete. Also influencing and aligning the enterprise architecture direction, project by project.

The simple answer is; it's the Principles that Agile and Adaptive software development stand for, that make sense to follow as an Enterprise Architecture set of processes and practice. Let's explore two categories:

The Agile Manifesto Principles

Let's look at the four principles from the Agile manifesto [3] and comment on them from an EA point of view as to whether the principles work within an EA context:

- **Individuals and interactions over processes and tools**
 - **Yes** – People collaborate in much more of a real-time basis, compared with tools and process, which tend to imply more of a delayed response time. This does not imply do not use process and tools, but collaborate and interact using the people skills primarily first, then refine the models using the tools and processes.
- **Working 'software' over comprehensive documentation**
 - **Yes** – especially if you replace the word 'software' with 'systems' or 'architecture'. Any good architect would have a working Architecture over comprehensive documentation any day. This implies; be risk driven, attack the highest risk to the enterprise first. In many companies they do not even have the concept of an Enterprise Architecture, so they have already defaulted to practicing this principle by fire-fighting; but an enterprise can only take it to the next level of efficiency by reducing complexity with more comprehensive Models.
- **Customer collaboration over contract negotiation**
 - **Yes** – If we define all the people that EA interact with as customers. In most cases, it takes time to build up trust and prove the EA practice's worth. This can only be done by showing

real added value and tangible benefits in monthly increments. No business will sit back and wait for a year while some team of people ferret away building up something they do not understand, costs a lot and appears to offer no value, even if you have put it down in some form of contract.

- **Responding to change over following a plan**
 - **Yes** – To be successful you need to be Adaptive. Have a plan for the short term (measured in weeks) and stick to it as much as possible, but respond to change within a Risk managed environment. This does not mean that there is no big picture plan; it is just not in great detail.

Agile-Adaptive Project Management Declaration of Independence (DOI)

Let's look at the six principles from the Agile Project Management [5] and comment on them from an EA point of view to see whether these principles work within an EA context:

- We **increase return on investment** by making continuous flow of value our focus.
 - **Yes** – Very definitely, this is crucial, especially since many executives battle to see the value an EA Practice brings to the organisation. An EA Practice should take on and deliver value as early as possible to ensure its continued existence.
- We **deliver reliable results** by engaging customers in frequent interactions and shared ownership.
 - **Yes**. - Once the business see the value and share ownership, the results become more reliable as they become engaged and open up.
- We **expect uncertainty** and manage for it through iterations, anticipation, and adaptation.
 - **Yes** – Uncertainty is the only certainty in business today, with the possible exception that uncertainty should not stem from poor iteration planning, a culture of fire-fighting and bad management practice, but rather from controlled rapid business change.
- We **unleash creativity and innovation** by recognizing that individuals are the ultimate source of value, and creating an environment where they can make a difference.
 - **Yes** – Books such as Peopleware [6] and The Mythical Man-Month [7] as well as other books have been saying this for years, and since EA teams have people this makes perfect sense.
- We **boost performance** through group accountability for results and shared responsibility for team effectiveness.
 - **Yes** – An EA Practice shares much responsibility in general, are typically small teams that are far reaching and have the same potential to have performance boosted and be effective.
- We **improve effectiveness and reliability** through situationally specific strategies, processes and practices.
 - **Yes** – The EA team in this sense benefit the organisation by taking this concept to the Enterprise level, because while Project teams each improve effectiveness on a particular Project, the EA team manage this concept across all Projects, Business Services, Systems, Information and Technologies to improve effectiveness and reliability for the whole enterprise.

2. Duality of process - Separate Cycles

Just like the Yin and Yang, there is a duality of Process, between *Software Projects Teams* (doing development or integration) and the *Enterprise Architecture Team*. Each team has its own cycle of Phases and Iterations within which it functions.

Software development should proceed in an Agile manner within the Software lifecycle (development or integration phases), minimising the Risk to the end product and to get a set of working software functionality delivered incrementally.

Likewise, the Enterprise Architecture Practice should proceed within its own lifecycle of phases. EA should engage with each Project at regular points to ensure the Project is following the Enterprise Strategies (or take a Tactical Dispensation). EA should ensure it does NOT slow a project's ability to progress by demanding huge time-consuming and lengthy deliverables.



There needs to be a balance between the ability to deliver and the longer-term benefits of aligning the overall enterprise architecture.

Enterprise Architecture by nature takes a **long-term view** of the Enterprise. It is more precise and accurate, building up a “town-plan” but only at a **low detailed level** of granularity and over a longer time span.

Projects on the other hand take a more **short-term view** because their sole concern should be to deliver working functionality quickly. They need to be precise but in a much **higher detailed level** of granularity, in a smaller more focused area.

3. TOGAF ADM Phases, Cycles and Iterations

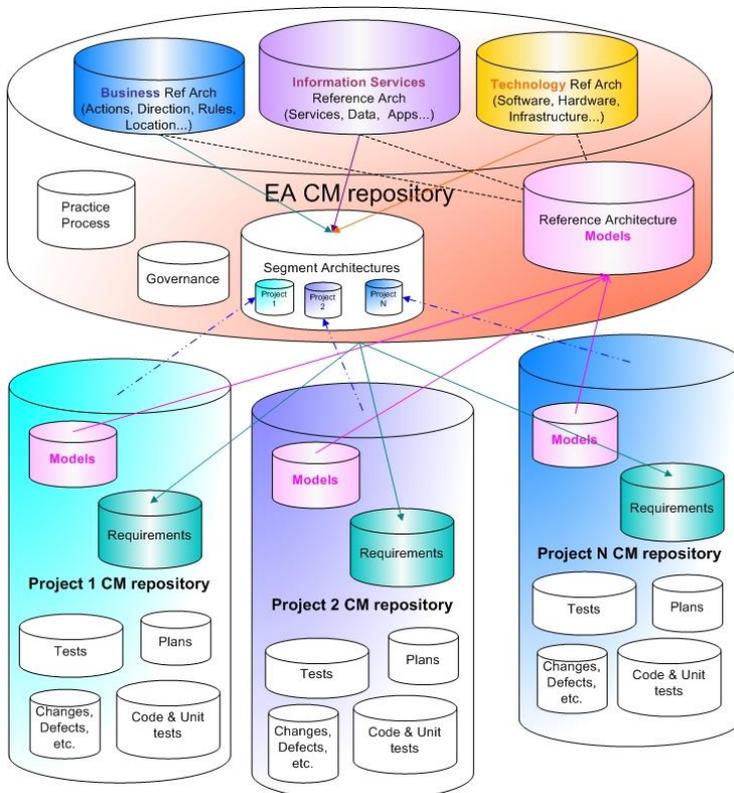
In general the TOGAF seems vague and confusing about the ADM phases and cycles.

- Should ADM be used as an *EA team cycle* going through all the phases, looking at EA as a whole and all the many projects?
- Should ADM represent a *cycle of Phases to be used on individual Projects*? Does this mean there can be many happening at once, because of the many projects happening at once?
- Can it be used at *different levels of abstraction*? The Enterprise level, the Business Stream level or the Project level? The answer is probably yes to all of these. None the less not clear.
- Is it left *deliberately vague* so that you can fit it to suit your own situation? If the latter then people can invent anything they like and it ceases to add value as a guiding mechanism.
- The words cycle, phase, iteration and stage seem to be used to have similar meaning.

It seems to confuse things because it assumes a one to one mapping between an EA Practice and a particular Project, which is never the case; there are multiple organization projects and application changes.

4. Traceability between EA and Software development

If the Enterprise Architecture has existing models and details about the enterprise architecture, then there are many benefits in tracing from this central repository into the project repositories:



Downward traceability: From an EA perspective; the ability to know which projects are working on what parts of the business, the information services and the technology. Project Management and EA can then understand impacts, potential duplication and this helps contains scope-creep.

Upward traceability: The Project’s ability to know exactly which parts of the organization any Project requirement will affect, be dependent upon and thereby impact. Traceability to existing context helps solve the problem of Projects wasting time investigating what is in place already. They will immediately have all the relevant information they need to understand the new requirements.

Traceability is the strongest way for connecting Projects to the EA. This would be if tools could support this seamlessly and effortlessly. A simpler way to still achieve these benefits might be to have the EA

Figure 2 - Traceability from EA to Projects

Models and information detail published on an intranet site for example, so that Projects could get an easy and “known to be the most up-to-date” information about the Enterprise Architecture.

5. Separation of Data and Function; learn from Object Orientation (OO)

This battle raged on in software development about Object Orientation in the 90’s and was solved back then, where it was understood that for the most part data is not separate from function. Classes contain both function and data. Somehow it seemed to escape the Enterprise Architects, or it appears they never updated their thinking on this.

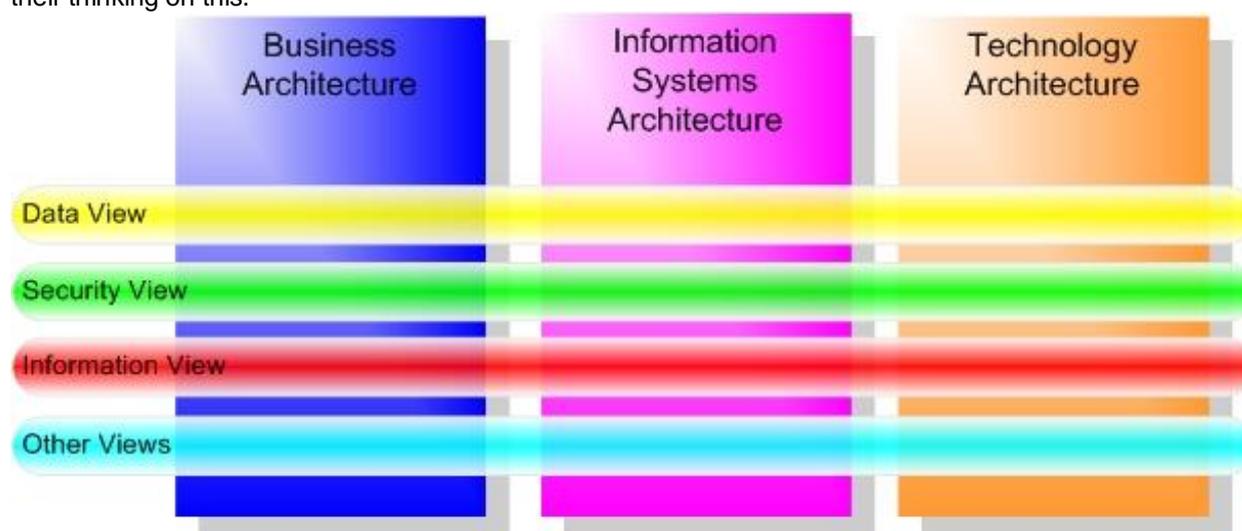


Figure 3 - Cross-cutting Aspects or Views of Enterprise Architecture

Data and Information are definitely valid views, but are cross-cutting aspects that cut across the whole of EA, from Business Architecture to Technology Architecture.

Just as Security is not a separate layer, it is just another view on the whole of EA from Business Architecture to Technology Architecture, and affects the whole. Information Systems is a good way of defining the layer/s that were called “Application” and “Data” or “Information”.

6. Maturity of the Organisation impacting EA

More can be read in detail in the article on Four Steps to Enterprise Architecture [9], but essentially Galen Gruman states that an organisation goes through four stages of maturity of Architecture:

1. **Business Silos** – Multiple applications with islands of information and functional duplication.
2. **Standardized Technology** – Solving the issue of many operating systems, DB’s, Servers, etc.
3. **Standardized Process** – End to end joined up business processes for the enterprise.
4. **Business Modularity** - Business processes and their supporting technologies are reusable modules.

The article also states that an organisation can only mature by moving from one stage to the next. Moving any quicker or jumping stages causes failure. On the assumption this is true the Agile Enterprise Architecture should take into account the level of maturity and adjust its process of implementing its EA accordingly.

7. Maturity of the EA Practice

How much time does the EA Practice spend getting it’s own house in order before helping get the enterprise house in order? This is one of the biggest dilemmas for an Enterprise Architect.

The business agreed to establish the EA Practice based upon the promise of the many valuable benefits to be derived through having such a Practice. Typically an EA Practice is made up of a set of expensive Architects. This means a high expectation from business not only to see value added quickly for the

businesses sake, but simply to justify the expensive EA Practice outlay. So from the business perspective EA should push effort to 99% for **Enterprise** and 1% Architecture.

The effectiveness of the EA team is constrained by EA Practice process maturity. If an EA Practice is in the early stages of incubation, then their working Methods, Processes, Work Products and Roles will be largely undefined. Each person will work as they see fit, with their own experience, assumptions, rules, goals and direction, instead of working as a team with a pre-defined common vision, processes, activities, roles and work products. By definition a disorganised EA team will at best be ineffective and most likely add to the problems of the Enterprise rather than make the enterprise more streamlined, efficient and aligned. So from an EA perspective they would vote for 1% Enterprise and 99% **Architecture**, at least initially.

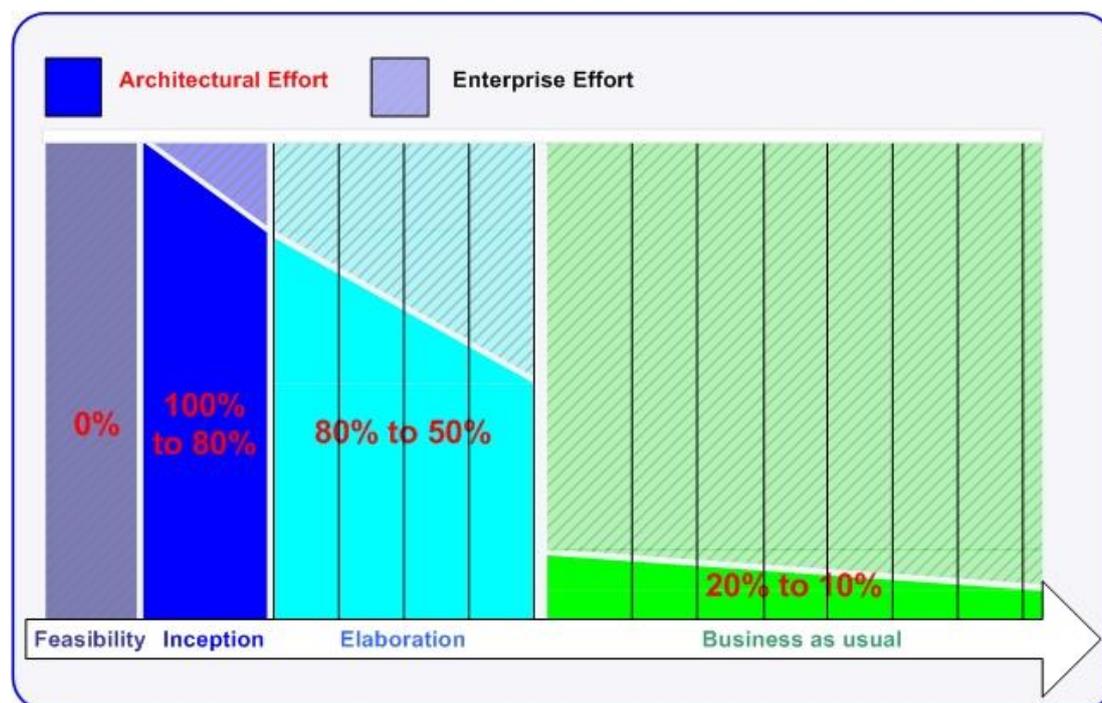


Figure 4 - EA Architectural vs. Enterprise Effort

The middle ground solution would be to take a curve of an ever decreasing percentage of Architectural effort over time, starting from 100% going down to around 20%-10%, based upon the particular Phase of EA. In the diagram below, Enterprise effort is shown in angled lines, the balance Architectural.

Most effort is spent on building the methods, process and work products in Inception, slightly less during Elaboration where the EA Practice starts engaging with the business up to 50% of the time. The real benefits begin being delivered into the Enterprise during the Business as Usual Phase on-going iterations.

Summary

The TOGAF contains an excellent set of information content for an Enterprise Architecture Practice. It offers a fantastic start to get an EA Practice running and mature, but it does have room for expansion and improvement. The Open Group admit this and are currently working on all sorts of improvements to bring out a version 9.0.

The next paper will explore potential solutions to some of the problems posed here.

If you have any feedback positive or negative, or any other suggestions or problems that should be overcome in Enterprise Architecture, please feel free to collaborate with the author.

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." (TOGAF can be downloaded: <http://www.opengroup.org/architecture/togaf8/downloads.htm>)
- [4] The Agile Manifesto; www.agilemanifesto.org
- [5] The Agile Project Management; www.agileprojectmgt.com; "Declaration of Interdependence for Agile-Adaptive Management".
- [6] DeMarco, T & Lister, T; Peopleware, Productive Projects and Teams. (1st edition 1987, 2nd 1999.)
- [7] Brooks, F; The Mythical Man-month. (1st Edition 1975, 14th 2000.)
- [9] Gruman, Galen; The Four stages of Enterprise Architecture, Jan 2006
http://www.cio.com/archive/120106/fea_mit.html
- [10] Fowler, Martin; The New Methodology <http://www.martinfowler.com/articles/newMethodology.html> (2005)