Extreme Enterprise Architecture Planning

Extrapolating agile characteristics to the development of enterprise architectures

Hugo Ramos and André Vasconcelos

Instituto Superior Técnico, University of Lisbon, Av. Rovisco Pais, Lisbon, Portugal
{hugo.ramos, andre.vasconcelos}@ist.utl.pt


Abstract: When developing enterprise architectures, in the same way as software products, companies have to deal with client’s demand for fast results, while facing, at the same time, a big uncertainty on the requirements surrounding the project. This thesis investigates the similarities between the difficulties faced in both industries of Enterprise Architecture (EA) and Software Development. That is the starting point, for a proposal of an extension to Enterprise Architecture Planning (EAP) methodology based on the idea that common difficulties can lead to common solutions, to which we will call Extreme Enterprise Architecture Planning (XEAP). The new approach, that we propose and demonstrate on this document, introduce agile characteristics such as several iterations, sequence deliveries and solution partitioning, into the process of enterprise architectures development. These characteristics are the answer to the need to obtain faster results and grow the response capacity to changing requirements. As a way to demonstrate and analyze our proposal, we apply it to a real project that has Câmara Municipal de Cascais (Portuguese city council in Lisbon district) as client. Later on, we compare the results obtained by our approach, with the results obtained by GFI Portugal (Consultancy in the area of Strategic Information Systems Planning), on the same project.

1 INTRODUCTION

Nowadays all enterprises, ones more than others, face big difficulties that come from their surrounding environments. With relentless competition in almost all sectors, comes the increase of new offers, substitute and competitor products and consequently the growing necessity and demand for faster results while constantly facing uncertain requirements.

Software development industry is a particular example of an area where those problems were more obvious. Therefore the companies started felling the necessity to use new and innovative ways of developing their products, once the traditional and standard ones were not able to answer the market expectations that were increasing in a really fast way. Those necessities were fulfilled by the introduction of the called “agile software development methodologies”, such as Extreme Programming (XP) and Scrum. Both Scrum and XP, are based on the motivation to deliver fast results to the client in an incremental and partitioned way while having an inside-costumer involved on the process in order to have a constant feedback, allowing to easily overcome the requirements uncertainty that are typical in such projects.

When observing closely, we can identify some similarities between the needs of both software development and enterprise architectures development industries, expectedly concerning the client’s needs for faster results while having a big requirements uncertainty originated by the surrounding environment. With the similarity between needs and the success achieved by the agile approaches on the fulfilment of those needs in the software industry, we intend to extrapolate some of the main agile characteristics of those approaches into a well-known traditional enterprise architecture development methodology, in order to overcome the demands described above.

Our proposal will be based on EAP methodology of Steven Spewak (Spewak, S. and Hill, S., 1992), to which we intend to make some “agile” changes, transforming it into an iterative process, while introducing characteristics such as solution
partitioning and on-side costumer, with the clear objectives of reaching a methodology capable of deliver faster results to the clients, while dealing with a big requirements uncertainty (when compared with the standard methodologies).

### 2 PROBLEM

The uncertain environment which the enterprises face nowadays, are closely connected with the clients own changing requirements and visions of the business. Not surprisingly, maintaining control over the requirements process is nearly impossible as each customer group pushes for its own interests and the changing technologies lure customers into escalating demands (Brooks, JoAnn M. and all, 2008).

The surrounding environments and the competitiveness of the markets originate enormous difficulties when trying to clearly define requirements, making the development of enterprise architectures an even more difficult process. Not rarely, this problem leads projects into a two way path, where either the project continues its normal pace, keeps all the original plans ignoring the changing environment and requirements ending in a completely failed project unable to achieve the expected results, either it tries to answer in an appropriate way to the changes and uncertainty of the requirements and ends up completely failing the predicted and agreed time schedule and/or budget.

Alongside with this uncertain environment it is the organizations increasing needs and expectations for shorter cycles with production of return, as well as faster results (Spewak, S. and Tiemann, M., 2006). The constant changing environment and relentless competition that enterprises face today brings them a high necessity for fast results in all the areas evolving the business in order to adapt and create new opportunities (Land, Martin O., and all, 2009).

Some years ago, the problems identified above (environment uncertainty and demand for faster results), were deeply evident on the software development industry, while this started being one of the most competitive and fast-growing industries. At this time started being globally recognized the urgent need for efficient methods and practices capable of facing the recognized demands. As an attempt to answer those demands, the notion of agile approaches started rising, where instead of developing software as a big complex and flat process ending in a big delivery, it would be done in an iterative way with several small deliveries in order to embrace and manage the possible changes that may happen along the process, while dividing a big problem into smaller ones (Sommerville, I, 2010).

Analysing in a more particular way, the projects of enterprise architecture development are not different from the generality and in this case there are some problems that with the growing of the companies had become more and more difficult to deal which reclaim for a methodology capable of dealing with those problems in the same way agile approaches did on the software development projects.

Quite often in EA projects, the clients find themselves obligated to choose from their business functions, the ones that must be actually considered on the architecture. Other functions that may also be important and critical end up being left behind, due to the limited amount of time allocated to the completion of the project (Townsend, J., and all, 2008). Those cases show us that we can achieve a level of independency between systems, capable of being explored in a way that delivering the results of different systems separately and in several iterations becomes a requirement and success factor instead of an obligation due to the tight schedules or complexity of the project.

As a way to summarize our problem we present the questions that we try to answer with our work:

- Is it possible to successfully develop an EA using agile characteristics as a way to overcome the uncertainty of requirements and the demand for fast results?
- Is a standard and traditional enterprise architecture methodology capable of “accepting” the introduction of agile characteristics?
- Are the “intermediate” outcomes that result from each XEAP iteration, valuable to the client?
- Is it possible to have such independence between processes that an enterprise architecture solution can be delivered in several “pieces”?

### 3 RELATED WORK

#### 3.1 Agile Software Development

Agile Software Development appeared some years ago has an answer to the fast changing, uncertain and unpredictable environments that surround the projects of the software development industry. These environments include client uncertain requirements, new target markets, substitute and competitor
products/services and even economic changes. Although all the difficulties, this competitive and restless industry started demanding for methodologies capable of delivering fast results, once this started emerging as the main requirement of the clients, leaving behind important requirements like software quality (Sommerville, I., 2010).

The most famous and used agile methodologies are Scrum (Schwaber, K., 1995) and Extreme Programming (Beck, K., 1999), which introduced some new concepts and characteristics to the software development process. From those characteristics we can highlight the introduction of iterations with releases and deliverables to the client at each one of them, the on-side customer with a constant involvement with the project and the short-term goals over the long-term ones.

These agile approaches are especially effective with projects where the systems requirements change quickly during the development process, once they are made with the intention to rapidly deliver results in the form of working software to the clients so they can give their feedback and change any system requirements if necessary (Sommerville, I., 2010).

3.2 Enterprise Architecture

Enterprise architecture can be described as a governance and decision making instrument with the capacity to fulfil the gap between enterprise’s vision, strategy, and change projects. Enterprise architecture tries to deal with this gap, by achieving a common, shared and unanimous comprehension about what are the company structure, business model and the necessary systems to support that model (Land, Martin O., and all, 2009).

Enterprise architecture provides a way of centralize and stabilize the information about the environment, making it consistent and normalized throughout the company and among the stakeholders of the business (The Open Group, 2009).

Some of the most used enterprise architecture development methodologies are Enterprise Architecture Planning (Spewak, S. and Hill, S., 1992) and TOGAF ADM (The Open Group, 2009). EAP is an older methodology especially focused on the information systems and that does not go further than the planning of the “TO-BE” state of the client. Contrarily, ADM is a wider methodology, not only capable of planning all the enterprise architecture, but also with concerning on the actual implementation process and its governance as well as on the change management. Being an overall simpler and shorter methodology, EAP constitutes a more suitable process for our purposes of adding some new agile characteristics, and therefore is the basis for our proposal and the general steps involved.

3.3 Reference Models

Reference models are prototypes of some application domain. Those models intend to reduce significantly the trouble inherent to the creation of application-specific systems, where we can select the more important parts of the model and adapt them to a specific problem. When applicable, this possibility gives us a huge advantage in terms of both cost and time saving, on the development of the projects (Ramesh, B. and Jarke, M., 1999).

On our work in particular, we will use the reference models with the clear objective of presenting results to the client as soon as possible, through the delivery of a first high-level architecture based on one specific model, considered suitable for our project, once this models can be used as a starting point to construct project-specific models (Becker, J., and all, 2007).

4 PROPOSAL

Our proposal consists on extrapolating some characteristics of the agile approaches used in software development, as Extreme Programming and Scrum, to the domain of the enterprise architecture. We will base our process on the
methodology where we will introduce agile characteristics.

We will give special attention to the inclusion of several iterations in the process, as a way to transform a slow, big and complex process into several sequenced simpler iterations while exploring the possible independence between components of the solution, which in this case are the information systems that support different business functions.

4.1 Differences/new characteristics

Iterations
EAP makes the definition of the business model and the correspondent architectures in a flat process with a single iteration. XEAP proposes to produce enterprise information systems architectures through an iterative process composed by several cycles. This way, the client is able to keep on track with the project with a good understanding of what is being done and in possession of results since a really early stage of the project.

Solution partitioning
As second change proposed by us, is partitioning of complex enterprise architecture projects into a group of independent systems. This partitioning would allow us to incrementally deliver the solution and therefore reduce the time that clients wait to have final results.

Continuous client feedback
Having an iterative process as base for the methodology brings a bigger involvement of the client with the project and allows a more accurate feedback.

4.2 Step-by-step process

First iteration
In the first iteration we will have the first contact with the client and the business. One of the main goals of our proposal is to deliver results as fast as possible, and therefore we intend to present a reference model adapted to the general information that we have of the client. With this first architecture we will give the client a high-level view of what their architecture should look like, and which systems are the more suitable to support their business. With this, we immediately show the client the path that we intend to follow and where to guide their architecture.

Business macro-processes model
In this first step we present a reference model containing macro-processes considered suitable for the type of business we are dealing with, relating them with the main information entities that we are able to identify through a simple analysis of general information.

Data architecture
After identifying the more important information entities on the previous step, we must do a data architecture capable of describing in a first high-level the relation between those entities and their characteristics showing the structure that is achieved when relating all of them.

Applications architecture
By relating the reference macro-processes and information entities identified in the first step, we are now able to understand which applications should be supporting the model described. As a result of this step, we are able to provide to the client, since the very first meeting, a description of the applications and systems he must have, representing an ideal “TO-BE” state and a difficult, but clear objective for the future.

Second iteration
On the second iteration we start performing a complete cycle of EAP process, purposely missing the last step of implementation/migration plan, which we will do only once in the end of the last iteration.

On this cycle we will focus on the business processes of the client but will not go further into detailed sub-processes. Those will be addressed on the next iteration.

Values & Principles
This step defines the basis for the EA and for all its future decisions. This phase is performed only on this second iteration, once it defines values and principles that must be followed during the rest of the process.

Business processes model
This step marks the beginning of the organization “AS-IS” state definition. Firstly we will need to identify and relate the processes of the business and the information entities used by those processes in order to achieve an accurate model of their reality.

Current systems & technology
This step completes the definition of the present state of the company it is important to describe the systems and technologies that are supporting the business functions defined previously. This phase is not only important to define what we have in the present, but also to help us understanding what we can or cannot have in the future, once will give us the possibility to do a later evaluation of the impact
that the architected systems and technologies will have on the current ones.

**Data architecture**
With the information entities identified before, we are able to formulate a data architecture that shows how those entities must be connected and structured in order to have the most efficiency possible when manipulating the data that supports the business.

**Applications architecture**
Through the understanding of how the business processes use each information entity, we are able to formulate and present a first group of candidate applications, which together can effectively support the organization activity. The result of this step is achieved using a CRUD matrix.

**Technology architecture**
After defining the more suitable applications for the business we must define the technology that will support those applications. Having into account that we are already on the second iteration, we will have to understand how the existing technology can or cannot handle those applications and what are the necessities, if any, of the organization in terms of technology infrastructure.

**Third iteration**
The third iteration is in general similar to the second one. The only difference between them two is on the level of detail of the business model and the processes used. On the third iteration we will decompose the previous processes into sub-processes and find some more information entities that they may use, going even further on the level of detail of our architectures.

**Business sub-processes model**
On this step of the third iteration we will start redefining the “AS-IS” state of the organization, now with some detail about their processes and information entities. We can now decompose the processes identified on the previous iteration into sub-processes and therefore identity new information entities.

**Current systems & technology**
This step marks the finish of the last “AS-IS” definition, and therefore the one that needs to be more close to the reality as possible. On this particular step we identify the systems that support the business sub-processes and information entities identified. Having into account that we are on the last iteration, we expect to have identified the complete set of systems that the organization currently use.

**Data architecture**
This step corresponds to the final data architecture, representing the necessary and more suitable structure of the complete set of information supporting the business.

**Application architecture**
This step corresponds to the final applications architecture. The relation between the most detailed sub-processes and all the information entities of the business will allow us to identify the final set of applications capable of supporting the complete organization activity.

**Technology architecture**
On this step we will finish our architecture definition with the presentation of the technology capable of supporting the applications identified on the previous step.

**Implementation/Migration plan**
Finishing our methodology and the project is the Implementation/Migration plan, where we make a planning of the systems that need to be implemented and installed, where we include effort, resources and benefits estimates, and an impact analysis on the current systems.

5 CASE STUDY

In order to demonstrate our proposed solution, we are applying our artefact to a real-world problem form a city council called Câmara Municipal de Cascais (CMC), where we are able to get all the information necessary to apply our approach and develop a complete architecture based on their business processes, systems and infrastructure supporting their activity.

5.1 Applying the methodology

**First Iteration**

**Business macro-processes model**
In order to achieve a preliminary business model we relate the macro-processes, based on the PCF reference model (APQC, 2012.) chosen for this specific project, with the information entities that are common to almost all businesses in general and are suitable to this one in particular.

**Data architecture**
On this step we relate all the data entities identified before. Through this relation we are able to provide a general view of the data structure that is more suitable for the business with special attention to the information that is shared by more than one entity.
Application architecture
After relating the macro-processes and information entities, identified before as being core to the business, through a CRUD matrix we achieve a clear understanding of the candidate applications capable of supporting the client activity in the most effective and sustainable way possible. During this process we must keep in mind that we are working with a reference model, and therefore, we are presenting an ideal situation to the client of what should his business architecture, and not yet representing is actual situation, which will be addressed later on the process following iterations.

Second iteration
Values & principles
The project will have the maximum duration of 3 months and must have into attention the fact that the operational part of the business is considered the most fragile at this point and where the client feels that our work can bring more benefits for their future, once the management and support processes are considered to work quite well at the moment. Although that we proposed to the client and to ourselves that we could address both sides of the business (operational and support) on the agreed time, without compromising the final product. We will search for a continuous contribution of the client in order to keep our work as informed has possible all the time.

Business processes model
After doing a survey of the enterprise business processes and information entities we start describing the “AS-IS” state, by relating them. On the process of gathering all the processes, we tried to make correspondence between them and the macro-processes defined on the first iteration, as a way to understand how far the reality of the organization is from the reference model. We must keep in mind that although the reference model constitutes an example of good structure for the business, it is not the only possibility, and therefore, it is not mandatory for a successful business to have that appearance.
Current systems & technology
After some interviews and research we identified the main systems that are supporting the business. Figure 6 presents the list of those systems with a brief description for each one.

<table>
<thead>
<tr>
<th>System name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GDCC</td>
<td>Document management solution. Supports the dematerialization of decision and administrative processes. Currently implemented on a Web environment, allows real-time access to all the information stored. All the income correspondence of CMCC is registered here in GDCC.</td>
</tr>
<tr>
<td>ARC 2000</td>
<td>This is the ERP solution of CMCC, divided by several modules and information management systems. The independence that exist between the several modules and systems, allows the global management of all the information that specific needs of the users. ARC performs the complete information treatment throughout the organization. It is divided in several modules for all the different areas where it is being used.</td>
</tr>
<tr>
<td>CRM</td>
<td>This is the system to manage the relation with the citizens. Eases the creation and maintenance of a clear vision of the clients, from the very first contact till the end of any request treatment process.</td>
</tr>
<tr>
<td>Legacy SQL/Visual Basic</td>
<td>Application developed internally, based on SQL and Visual Basic. It is responsible for the management of economic activities licensing processes. Allows the management, renewal and printing of licenses.</td>
</tr>
<tr>
<td>SIGWeb</td>
<td>Web version of the geographic information system. Provides all the geographic information to the users, both within CMCC and external ones. Developed and customized accordingly to CMCC needs.</td>
</tr>
<tr>
<td>AutoCAD</td>
<td>Solution for the areas of projects development, allowing doing it in an aggregated way.</td>
</tr>
<tr>
<td>GEDOM</td>
<td>Customized platform based on Microsoft Project Enterprise Server system. This allows the users to manage all the projects of each department.</td>
</tr>
<tr>
<td>Occurrences Manager</td>
<td>Solution connected to SIG. Allows geolocalising places where the employees of CMCC have to go, as a response to some occurrence.</td>
</tr>
<tr>
<td>MAPrefc</td>
<td>With this tool it is possible to do data mapping, based on geographic analysis. It has been designed to visually obtain the relations between data, georeferences and other geographic information.</td>
</tr>
<tr>
<td>Legx</td>
<td>Solution available through web. Allows searching the legislation and provides all the published legislative acts.</td>
</tr>
<tr>
<td>BzTalk</td>
<td>Business Intelligence solution. Incorporates several business applications: EDI, Business Activity Monitoring (BAM) and RFID resources.</td>
</tr>
<tr>
<td>Project Server</td>
<td>Used for the planning of construction projects. Provides advanced tools for small teams or individuals, assigned to manage projects. Provides tools for the management of work, time, resources and budget.</td>
</tr>
<tr>
<td>Adobe Creative Suite</td>
<td>Set of tools for designers with the need to express ideas in a precise and fluid way. Provides the possibility to create images and attractive graphics.</td>
</tr>
<tr>
<td>Queues Manager</td>
<td>Allows the centralization of requests waiting queues. Provides an easy way to produce statistic information concerning the requests answering process.</td>
</tr>
<tr>
<td>Iagitrium</td>
<td>Allows the global management of CMCC cultural patrimony.</td>
</tr>
<tr>
<td>DataEase</td>
<td>It is used to register purchasing procedures, placing minutes and manage warehouse.</td>
</tr>
<tr>
<td>XtraH</td>
<td>Application managing all the resources operating outside on the field. Provides real-time information about workers and machines, helping the managers to make better decisions based on reliable data.</td>
</tr>
</tbody>
</table>

Figure 5: Relations between business processes and current systems.

Applications architecture
On this second iteration we were able to identify 10

Figure 6: Second iteration relations between information entities.

Data architecture
Through an analysis of the information entities identified before, and after a clear understanding of their characteristics, purposes and use cases, we can achieve a data architecture capable of showing a suitable structure capable of fulfilling the business demands in an effective way, free of incompatibilities between entities.

Figure 9: Second iteration CRUD matrix
main applications that seem to be the most appropriate, in order to accurately support the business processes and information entities described on the previous steps.

**Technology architecture**

Having the list of current systems, we will now identify the needs of CMC technology structure, considering the applications identified on the applications architecture phase. Here we will make some recommendations on which systems should be replaced, removed, upgraded, kept or added to the CMC structure.

**Third iteration**

On this third iteration, we decompose each business process identified on the second iteration, into subprocesses. Those subprocesses are the base for our last delivery.

**Business sub-processes model**

**Current systems & technology**

On this third iteration, we identify some new systems that we did not include on this same step from last iteration.

**Data architecture**

On this third iteration, we decompose each business process identified on the second iteration, into subprocesses. Those subprocesses are the base for our last delivery.
Figure 11 shows the third iteration data architecture. This diagram puts together all the information entities found as being core to the business of CMC and shows how they must interact with each other.

**Applications architecture**

On our last iteration, we manage to find 15 core applications that seem to represent the most accurate and effective structure capable of supporting the existent interaction between business sub-processes and information entities.

**Technology architecture**

Figure 13 presents the last iteration technology architecture. This shows which systems should support each application identified on the previous step. On the systems presented, we include systems that were already on CMC IT structure, and also new systems that we consider to be essential to acquire.

**Implementation/Migration plan**

As last step of our EA development process, we make several recommendations to our client, in order to help him with the implementation and migration from his old condition, to the new architectures that we just presented. On those recommendations, we include the order that we consider to be the best for the identified applications implementation - The implementation order should be the following: App 2; App 1; App 3; App 6; App 7; App 5; App 8; App 4; App 9; App 11; App 12; App 13; App 10; App 14; App 15.

5 EVALUATION

GFI’s methodology is very similar to traditional EAP, and that seems to constitute the perfect condition to analyze and test the differences between the original methodology, and the extension that we try to develop with this work. That comparison is made through figure 14.

5.1 Analysing preliminary results

From figure 14, we can highlight some points that help us understanding the outcomes of our work, and what they mean:

- On the very first week of project, we were able to deliver a data and applications architectures, providing, in the earliest stage of the process, a high-level view of the final stage result. The very limited knowledge about the organization was surpassed by the usage of reference models capable of representing in a superficial, but elucidative way, the client’s business model and organization structure.
• The need for fewer interviews on the process of building our second iteration, allowed us to have a second architecture, with an intermediary level of processes detail, ready to present to the client before the traditional approach had finish its complete set of interviews.

• We were able to maintain a separation between Operation and Management and Support processes, during all the development of our work. This intended to show that there is a level of independence between processes, that, when explored, can be used to introduce the solution partitioning agile characteristic into this type of project.

6.1 Contributions

The main academic contributions of our work are:

• How and which agile software development practices can be extrapolated to enterprise architecture development;

• How Enterprise Architecture Planning methodology can be extended and accept the introduction of “agile” characteristics;

The enterprise contributions of our work are:

• How the introduction of “agile” characteristics in EAP can help with the delivery of faster results and quicker response to environment changes when compared with the standard approaches;

• Understand if characteristics like process iterations, small releases and continuous client are the responsible for achieving faster results and bigger response capacity to changing requirements;

REFERENCES


Figure 14: Comparison between XEAP and GFT’s methodology

6 CONCLUSION

Our work addresses the difficulties faced by the today’s companies when performing projects of enterprise architecture. The client’s high exigency for fast results and the constant change of requirements were identified as main demands that needed to be fulfilled and were was a flaw on the existing enterprise architecture development methods.

The demands that motivated the emergence of agile approaches in software development industries and their similarities with the ones described above, guided our solution into an extrapolation of the main and more adequate characteristics of those approaches into a well-known EA development methodology.