White Paper

Enterprise Architecture: Beyond just IT

WP0202 | July 2015

Karl Schulmeisters

Karl Schulmeisters is Technology Advisor for the Carver Global Health Group where he provides expert leadership in Business Process Architecture, Enterprise Architecture and Cloud Computing. Karl's current emphasis is on the impact and integration of disruptive technologies into traditional enterprise IT organizations: Cloud, Mobility, Consumerized IT, Machine Learning/Big Data and Social Media.

Karl Schulmeisters is an internationally recognized speaker – his most recent speaking engagement was at the Congress on the Future of Engineering Software in St. Petersburg Russia. He welcomes your comments at *karl.schulmeisters@cg-hg.com*

Enterprise Architecture – To IT and BEYOND!

As with my previous white paper: Enterprise Architecture and Agile Development, Contradiction or Synergy, this paper is focused outside of the IT implementation details of Enterprise Architecture. It comes out of my experiences using EA to facilitate rationalization of the technologies involved in the product portfolio of a major technology vendor.

In an earlier paper I asked the question Enterprise Architecture: Is it just Innovation Management? Is it just Application Portfolio Management? I then proceeded to discuss how both disciplines can benefit from using an Enterprise Architecture approach. This paper expands on that theme to discuss how Enterprise Architecture can play a very useful role in the customer facing product development for technology driven corporations.

Review: What Enterprise Architecture is

First a quick review. What is Enterprise Architecture? Many start by invoking Zachman Frameworks[i] and TOGAF®[ii] and similar implementation approaches of Enterprise Architecture. But fundamentally what Enterprise Architecture is about is managing technology complexity in harmony with business goals.

Initially it was developed for aligning the IT organization's services, technologies and capabilities with the business goals and needs of the corporation. It was a way of dealing with the technological complexity of Information Technology in the context of business needs.

Today though, technology driven corporations face ever increasing complexity in their product lines as well. And since Enterprise Architecture is about managing technology complexity in harmony with business goals, it seems logical to apply it to aligning product technologies with business goals as well.

Traditionally the focus on managing technological complexity has applied to internal Application Portfolio Management and Infrastructure Technology management leveraging tools like ITIL.

In the age of ever accelerating technical development the need to integrate disruptive technologies not only into business practices but into customer offerings as well has never been more important.

The question is though what technologies to focus on. For example,

- Should IoT[iii] (Internet of Things) be part of your customer offering? Or is it too early in the product maturity curve?
- Should you be building for iOS? Android? Or even maybe Windows Phone? How do you chose?
- Cloud adoption is accelerating. Should you be using Cloud as a means of delivering solutions to customers? If so is avoiding vendor lock-in an issue? Should you go with one of the “Big Three” vendors or a smaller more agile player? Should your Cloud engagement be integrated into your existing infrastructure or be standalone?
- Do your customers expect you to deliver solutions on their employee BYO devices? Or should you be offering a standalone web solution? Or should you integrate into your client’s corporate portal

These are but four of the major categories of decision making that CTOs and Business Development leadership face in technology companies. Each of these decisions can be aided through the application of Enterprise Architecture principles.

Is Enterprise Architecture Owned by the CIO or the CTO?

Notice that I mentioned CTOs here. Historically the Enterprise Architecture discipline has originated in the CIO's organization. So what is the difference? Well let's first look at the difference between the two roles. Tech Republic™ [iv] defines the difference as follows:

Chief Information Officer

- Serves as the company's top technology infrastructure manager
- Runs the organization's internal IT operations
- Works to streamline business processes with technology
- Focuses on internal customers (users and business units) [emphasis added]
- Collaborates and manages vendors that supply infrastructure solutions
- Aligns the company's IT infrastructure with business priorities
- Develops strategies to increase the company's bottom line (profitability)
- Has to be a skilled and organized manager to be successful

Chief Technology Officer

- Serves as the company's top technology architect
- Runs the organization's engineering group
- Uses technology to enhance the company's product offerings
- Focuses on external customers (buyers) [emphasis added]
- Collaborates and manages vendors that supply solutions to enhance the company's product(s)
- Aligns the company's product architecture with business priorities
- Develops strategies to increase the company's top line (revenue)
- Has to be a creative and innovative technologist to be successful

And although the two roles can and sometimes do blur in the real world the key difference is that the CIO is focused on internal customers and the CTO is focused on the external customers. So with Enterprise Architecture originating in the CIO organization it is little surprise that it has been more internally focused and less frequently used to manage technical complexity in customer offerings.

But the key issues – aligning technology with business goals, be they operational business processes or the business driven capabilities of the products you are selling – are very similar. In both cases you have an existing business structure as well as a roadmap for the future, and in both cases you have a technology structure and roadmaps for those technologies in the future.

And as with the internal IT Application Portfolio Management and Innovation Management, there is a need to align the complexity of the Information Technologies being used to implement customer facing products and the Business Goals – in this case product features – being supported. Two key differences, however, do come into play when using Enterprise Architecture for managing this external facing complexity:

1. *Non IT Considerations* – products may often incorporate other non-IT technologies and technology considerations (such as a high tech material being used), and there may be more regulatory constraints imposed.

2. *Brand Management* – Internally it is possible to make a product change simply considering functionality and retraining costs. When those changes affect the look and feel and other perceptions of the product, Brand management considerations also start to play a major role in the business roadmaps.

So to answer the question posed at the start of this section – Does the CIO or CTO own Enterprise Architecture in a company – the answer is that both do. But for their respective domains – because there are considerations the CTO or the Product Portfolio management team has to react to that do not apply to an internal IT organization.

Considerations in Applying Enterprise Architecture to Product Portfolios

Except in the cases of very new startups, new product decisions are mostly driven by market considerations: customer expectations, competitor offerings, price performance targets. Thus the change here is originating from the business side of the equation more often than not. This is what I dubbed Outside-In in an earlier whitepaper *Enterprise Architecture: Inside Out or Outside In?*[v], because technology is the follower to externally driven considerations.

In the white paper *Enterprise Architecture: Outside-In Business Goals to Infrastructure*[vi] I suggested that a good starting point for *Outside-In Enterprise Architecture is to use the ITIL Maturity Model*[vii]. In particular I suggested mapping the Business functions – in this case the solutions, and their product features into the applications - in this case technologies – that support them. I suggested that Orbus Software's iServer Enterprise Architecture Solution with ArchiMate was a good tool to use for this.

However as I mentioned before, there are other considerations that customer facing products need to consider. And most of these come in as either business requirements or constraints on the business roadmaps:

- *Regulatory issues* – for example products in the healthcare field will often have Patient Privacy data restrictions (HIPAA[viii]). These would be entered as product requirements
- *Competitive Features* – the list of capabilities required to compete with a competitors product offering would also be entered as a product requirement
- *Capability and Feature Roadmaps* – These would be entered as business roadmaps. Essentially future capabilities
- *Customer driven technology requirements* – This is where traditionally IT focussed Enterprise Architecture would disqualify a solution for incompatibility. But in the Product Portfolio management case, a conflict between customer driven technology requirements and the product technology would instead trigger a cost benefit analysis of the work involved in customization vs the profit and opportunity presented
- *Brand Management* – customer facing products often have strong brand associations. Maintaining such branding profiles would be both a product requirements tracking task as well as a business roadmap task
- *Non IT Technologies* – as more and more products integrate Information Technology as part of other technologies, there needs to be a bidirectional capability dependency tracking. For example, for a robotic floor cleaner there clearly is the on-board control logic. But the details of that control might vary depending on whether the robot is a floor mopping robot, one focussed on pet households and one that is optimized for rug cleaning.

As can be seen, none of these really fall outside the ability of Enterprise Architecture to support and integrate, but it takes some planning to map the different issues properly.

Where Does Inside-Out Fit

In today's competitive markets, innovation in the product line is far more frequent and important than the role it plays for internal Information Technology evolution. Where an internal application can continue to run an older version of the OS or middleware – customer facing products need to continually innovate for the latest platform and technology environments

This is where Enterprise Architecture has both the most potential to shine as well as to be the most harmful to product innovation. The strength

Enterprise Architecture brings is that by aligning technologies to their business requirements and roadmaps, Enterprise Architecture can reduce the amount of technologies and "innovations" that really are just "solutions in search of a problem". The formalism of mapping a technology into a product feature requirement – particularly one on an existing business/ sales roadmap - helps clarify the value add of a new technology.

The flip side of this is that it is often not only difficult but impossible to map a disruptive technology into an existing sales or business roadmap.

A concrete example might be IoT enablement. Much of what is currently being put forth as disruptive Internet-of-Things technology, is simply something that in the past would end up being sold by late night infomercials. Yet at the same time it is clear that IoT holds huge promise and future capabilities in areas like medical devices. An example of this latter is the Automated Emergency Defibrillation Drone[ix].

By applying the Enterprise Architecture principles of identifying the corresponding Business Goals and Customer needs early on a much better case can be made for validating or rejecting those business goals. This is particularly important in the identification of the "minimum viable product"

Enterprise Architecture Can Improve Product Portfolios but There is a risk

So as we have discussed in this paper, Enterprise Architecture and tools like iServer can improve customer facing Product Portfolios. Existing product technologies become better aligned with the products, product lines become more distinctively delineated, product Branding is made more visible in the early stages of the design cycle, and almost most importantly, it is less likely to have a product created that is a "solution in search of a problem".

Yet there are risks with this approach. As we first suggested in our paper on Enterprise Architecture and Agile[x] Enterprise Architecture by its very nature is designed to slow down the rate of change. For products that are disruptive, or for markets that are in the process of being disrupted, it can be dangerous to take a "go slow" approach.

References

[i] Zachman Framework - www.eacoe.org/enterprise-architecture-defined.php?gclid=CLj_7ImHp8YCFWQTwwodMKMJUQ

[ii] The Open Group Architecture Framework - http://pubs.opengroup.org/architecture/togaf9-doc/arch/

[iii] IoT - www.forbes.com/sites/jacobmorgan/2014/05/13/simple-explanation-internet-things-that-anyone-can-understand/

[iv] Tech Republic Tech Sanity Check, June 22, 2008, www.techrepublic.com/blog/tech-sanity-check/sanity-check-whats-the-difference-between-cio-and-cto/

[v] www.orbussoftware.com/resources/downloads/enterprise-architecture-inside-out-or-outside-in/

[vi] www.orbussoftware.com/resources/downloads/enterprise-architecture-outside-in-business-goals-to-infrastructure/

[vii] http://itil.selfsurvey.org/

[viii] HIPAA www.hhs.gov/ocr/privacy/index.html

[ix] http://techcrunch.com/2014/10/31/this-ambulance-drone-can-fly-into-trouble-with-first-aid/

[x] Enterprise Architecture and Agile Development, Contradiction or Synergy? www.orbussoftware.com/resources/downloads/enterprise-architecture-and-agile-development,-contradiction-or-synergy/

Orbus Software UK
London

Orbus Software US
New York

Orbus Software AUS
Sydney

Orbus Software RSA
Johannesburg

enquiries@orbussoftware.com | **www.orbussoftware.com**

Seattle Software Ltd. Victoria House, 50-58 Victoria Road, Farnborough, Hampshire, GU14 7PG. T/A Orbus Software. Registered in England and Wales 5196435