What is Enterprise Technical Debt?

Abstract

If you've ever tried to write a Business Case for justifying an Enterprise Architecture Practice, you'll realize it is difficult. So recently when I saw the best distillation of the whole concept into something that is simple to understand and makes the most sense to me, I thought it deserved to be aired.

It is the concept called the “Debt Metaphor” summed up in EA terms as “Enterprise Technical Debt”. It's brilliant because it translates the Architectural issues directly into money terms that the Business people understand.

Introduction

If you've ever tried to write a Business Case for justifying an Enterprise Architecture Practice, you'll agree that to translate what for technologists and others who "just get it" think of as “…just so obvious and why would we even want to discuss the why’s – in these days of financial crisis, let's just get on with it and not waste any more time and money describing it while the enterprise flounders…”

Enterprise Architecture when implemented correctly helps maximize the capability within the organisation to minimize the cost and time to converge its Enterprise Information Systems; to be as well organized and efficient as possible. These concepts save time and money, making the enterprise more Agile, etc. Bla bla bla....Spend thousands... save millions...“Yeah right”, they say, “we've heard it all before!”

These EA concepts and benefits are extremely difficult to describe in an easy to understand, meaningful manner, to business people. Particularly to do it so well that executives just want to rush out and spend on EA immediately once they have heard about the concepts. This is partly because over many years, so many promises have been made and broken, so many attempts have tried and failed, all in the name of "Enterprise Architecture", that the term itself now has become stigmatized. Architects who understand the concepts tend not to have the patience either, trying to convert or translate many of these architectural principles into money and benefits-realization terms. I have certainly tried to avoid this documentation and explanation process wherever possible, but in order to justify the existence and get any funding you really need to roll up your sleeves and be prepared to do the difficult sell.

The Debt Metaphor

So recently when I saw the best distillation of the whole concept into something that is simple to understand and makes the most sense to me, and I would suggest, to any executive of any size enterprise, I thought it deserved to be aired.. It is the concept called the Debt Metaphor summed up in EA terms as "Enterprise Technical Debt".

It's brilliant because it translates the Architectural issues directly into money terms that the Business people understand. I first saw this in Pragmatic EA [1] and it makes perfect sense to me. Also from Ward Cunningham describing it in Software terms (not EA terms but very similar) [2]

The concept of Debt Metaphor was first described by Ward Cunningham and there is a You Tube video by the man himself describing what he means below.
I am however taking this concept and broadening the scope of it to suit the broader Enterprise Architecture rather than the Software Solution Architecture of one software program in an enterprise:

http://www.youtube.com/watch?v=pqeJFYwnkJE

Martin Fowler [3] also explains it in simple terms. This paragraph below is taken directly from a blog by him and has been modified by me to suit Enterprise Architecture:

“"You have a piece of functionality that you need to add to your enterprise. You see two ways to do it, one is quick to do but is messy - you are sure that it will make further changes harder in the future. The other results in a cleaner design, but will take longer to put in place."

Technical Debt is a wonderful metaphor developed by Ward Cunningham, (modified by PragmaticEA)[2] to “Enterprise technical Debt” to help us think about this problem. In this metaphor, doing things the quick and dirty way sets us up with a Enterprise technical debt, which is similar to a financial debt. Like a financial debt, the Enterprise technical debt incurs interest payments, which come in the form of the extra effort that we have to do in future development projects because of the quick and dirty design choice. We can choose to continue paying the interest, or we can pay down the principal by refactoring the quick and dirty design into the better design. Although it costs to pay down the principal, we gain by reduced interest payments in the future.

The metaphor also explains why it may be sensible to do the quick and dirty approach. Just as a business incurs some debt to take advantage of a market opportunity Architects may incur “Enterprise” technical debt to hit an important deadline. The all too common problem is that IS/IT organizations let their debt get out of control and spend most of their future development effort paying crippling interest payments.”

Summary

To me - if we can achieve explaining this to the executives of any Enterprise, we would be well on our way to getting their buy-in to funding a business as usual Enterprise Architecture practice.

We could prove it works by measuring the money metrics in each case of how Enterprise Technical Debt (albeit rather subjective) are saved and lost and track the ongoing accumulation of interest on that Enterprise Technical Debt.

It should be captured as a money metric monthly and played back in total to the Enterprise monthly.

References


Contact details