the Tetradian weblogs

Creating a career in Enterprise Architecture

Theory, roles and practice

Tom Graves

Creating a career in enterprise-architecture

Theory, roles and practice

Tom Graves

This book is for sale at http://leanpub.com/tp-career

This version was published on 2022-05-19

This is a Leanpub book. Leanpub empowers authors and publishers with the Lean Publishing process. Lean Publishing is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

Contents

This anthology from the **Tetradian weblog** provides overviews and commentary on how to develop a career in enterprise-architecture, business-architecture and related fields, and on the various roles that apply in those fields. It is intended to be a useful introduction for newcomers to enterprise-architecture.

This sample contains around one-tenth of the content from the full anthology. The complete book includes about 30 posts and 30 images from the weblog. Those posts are split into nineteen groups:

- *Basics: Architecture as Career* - offers practical guidance for anyone interested in developing a career in enterprise-architecture or any of the related disciplines.
- *Basics: Architecture Roles* - describes the different roles across the overall architecture domains, what they each do, and how they together across the whole to support effective change.
- *Basics: Enterprise and Organisation* - clarifies the crucial distinctions between 'enterprise' and 'organisation', and what these mean in practice.
- *Basics: Architecture and Design* - shows how and why to differentiate berween architecture and design, and the different roles they play in the overall process of change.
- *Basics: Guiding Themes* - presents a set of simple yet essential checklists to guide architecture-development practice.
- *Basics: IT-Architecture* - explores the role of IT and IT-architecture in enterprise-architecture, and the risks presented by IT-centrism - an over-focus on IT above everything else.
- *Basics: Business-Architecture* - explores the role of business and business-architecture in enterprise-architecture, and the risks presented by business-centrism - an over-focus on business above everything else.
- *Basics: Big-Picture Architecture* - combines enterprise-architecture and the futures-disciplines to extend EA outward to a fully-global and longer-term scope and scale.

- *Basics: Examples* - presents practical examples of themes that that we may need to address in enterprise-architecture.
- *Basics: Concepts* - outlines some of the key concepts that underlie enterprise-architecture and the related disciplines.
- *Basics: Method* - presents methods for various specific tasks in architectures, their purpose, and how and why they add value to the overall process of change.
- *Basics: Content* - outlines some of the key content that we use in architectures, and the role that each item plays within the overall process.
- *Basics: Effectiveness* - introduces 'enterprise-effectiveness' as an alternate frame to 'enterprise-architecture', as EA has become so tightly associated with IT that it's routinely dismissed by many potential stakeholders as 'just an IT-thing' - missing out on the value that EA would provide.
- *Basics: Services* - explores some of the ways that services can be applied in architecture, design and implementation.
- *Basics: Values and Quality* - assesses how value, quality and values underpin the design and operations of every type of architecture.
- *Basics: Story* - demonstrates how to use story and narrative to guide and communicate what needs to be done in an architecture.
- *Basics: Uncertainty and Complexity* - reviews the impacts of complexity, uncertainty and uniqueness on architecture and system design.
- *Basics: Discipline* - indicates key issues and challenges of discipline, governance and formal-rigour, and what happens to an architecture or the like if any of these fail.
- *Basics: Related-Themes* - wanders through a variety of themes related to architectures and suchlike.

Where do we start with EA? – a practical question

You’re an experienced enterprise-architect, having spent most your working life in one industry. You now have a new job, in a new company, in an industry that’s entirely new to you. And the company at present has no architecture at all: you’re ‘it’. Where on earth do you start?

That’s the situation my friend Alan finds himself in right now. An interesting challenge – and some very real, very *practical* questions to face. Right here. Right now. *Today*.

(I guess this post can also start to answer in part the question from the previous post, ‘How do we make EA make sense?[1]’.)

So: in essence, we start from scratch.

Which means that several threads need to start straight away, somewhat in parallel:

- the *politics* and pragmatics of architecture
- setting the stage – the ‘*big-picture*’
- finding *allies* – people who know ‘the trade’
- establishing *standards*
- finding the *story*

The first point is that *everything* about any form of enterprise-architecture is intensely ‘political’, in several different senses – which means we need to face the **politics** of this straight away, right from the start. One of the best guides to this is Kevin Smith’s

[1] http://weblog.tetradian.com/how-do-we-make-ea-make-sense/

PEAF[2] (Pragmatic Enterprise Architecture Framework). In essence, it's a step-by-step guide on how to get enterprise-architecture going within an organisation: see the website[3] and book[4] for more details. [Disclaimer: I edited the book.] The details are all there on the website anyway, so you don't even have to buy anything (though no doubt Kevin would like it if you did!).

Probably *the* single most important concern is to get 'buy-in' at senior level – certainly from the respective CxO for the main focus area (e.g. the CIO, for enterprise IT-architecture), but preferably from the CEO and entire executive. To be blunt, if you don't have that 'buy-in', you'll be going nowhere: you *need* to get the executive on-side.

As PEAF emphasises, the key to getting the executive on-side – and everyone else on-side, too – is *communication*. One valuable aspect of this is to get them *personally* engaged in describing the **big-picture** of the overall ecosystem in which the organisation operates, and where the organisation fits within that ecosystem. In effect, what we would do here is identify the high-level 'why' for which the organisation is a 'how' – in other words, the 'why' that provides the anchor for all of the organisation's strategy.

There's a lot of detail on how to do this in the chapter 'Step 1: Know your business' in my book *Doing Enterprise Architecture*[5] – that chapter is part of the 'sample-ebook' version that you can download for free from here[6]. (See the post 'Tools in action[7]' for some photos of those techniques in live use in an executive-level workshop.)

What I also usually do is plough through the publicly-available sources such as the organisation's website, publications, advertisements, intranet and annual-report. There's usually enough information there to build some preliminary models with which to get

[2] http://www.devx.com/enterprise/Article/47353
[3] http://pragmaticea.com/
[4] http://store.peaf.com/introductiontopeaf
[5] http://tetradianbooks.com/doing-ea/
[6] http://tetradianbooks.com/doing-ea-ebook/
[7] http://weblog.tetradian.com/tools-in-action/

started: or at least, enough for people to tell us that the models are wrong – which is a nicely sneaky way of getting them engaged in telling us their ideas about what it *should* be! You'll also notice in that 'Tools in action' post that we included people's own photos on some of the larger wall-mounted models: this again is a good way to get people engaged, and to get conversations going between them about what *they* can do to improve their own work-context.

Whilst we're doing this, we need to be looking for any **allies** – people who are *already* committed to other themes that connect with enterprise-architecture, and would be likely to see the value of synergy between those areas of interest. This is *really* important if we've only just started with the organisation, because – in my experience, anyway – enterprise-architectures depend greatly on person-to-person conversations and connections: knowing who to talk with, and how to talk with them, will depend in turn on backgrounds and credibility and personal-networks within that organisation that typically take at least five or more years to develop.

Those are people we *need* as allies: and finding them is one of our first and most urgent priorities as soon as we start work at new place. One tactic I've used for this is to sit in the cafe or whatever with a few interesting-looking diagrams spread out over the table: anyone who's interested enough to stop by and chat is likely to be that kind of 'connector', or will at least know someone who is. Again, despite all those models and the rest, what *really* drives the architecture – what makes it happen, in real-world practice – is person-to-person conversations.

Another concern that those allies can help us with straight away is in identifying the **standards** that apply in the context. Some standards would apply to just about every industry, but we'd be likely to know those already. Other standards will be generic for the industry as a whole, but they're usually not hard to find: for example, for this case, in retail, a quick web-search on "retail reference architecture”[8] turned up a swathe of IT-oriented standards from Oracle, Microsoft, Cisco and IBM, together with articles on how to put them to practical use. This is also where TOGAF[9] and the other enterprise-architecture ‘usual suspects’ will start to be of real value – though to be honest they can often be more of a hindrance than a help *before* this stage.

What we’re also looking for are all the other standards and guidelines and workarounds and the like that are specific to *this* organisation, some of which – perhaps many – may not exist anywhere in any written form. And that again is where our allies can be *really* helpful, because otherwise we’d have little chance to know what these are. (And yet everyone else would expect us to know them, because, after all, we are ‘the architects’, aren’t we? – we’re the ones who are supposed to *know* all this stuff…?)

And we’ll also need to be on the lookout for standards that *should* be there, and aren’t. Which can be a little bit tricky, from a political perspective – not least because it tends to highlight issues that people ‘should’ have known about already, and didn’t… Once again, our allies will be invaluable here, in finding suitably-stealthy ways to introduce these ideas, and to smooth out any ruffled-feathers that may arise.

One trap to watch for is to beware of bringing too many assumptions from our previous organisation and industry: many of those assumptions *will not work* in this new context. The skills and experience of ‘how[10] to[11] do[12] architecture[13]’ are probably the only part of the work that will remain unchanged: we need to able and willing to challenge ourselves on just about everything other than that.

Almost all of that above is about enterprise-architecture as *struc-*

[8] http://www.google.co.uk/search?q=retail+reference+architecture

[9] http://www.opengroup.org/togaf/

[10] http://tetradianbooks.com/everydayea/

[11] http://tetradianbooks.com/ecanvas/

[12] http://tetradianbooks.com/silos/

[13] http://tetradianbooks.com/real-ea/

ture. The other side is about about architecture as **story**, the second of Matthew Frederick's 'two points of view[14]' on architecture:

> ARCHITECTURE IS AN EXERCISE IN *NARRATIVE*. Architecture is a vehicle for the telling of stories, a canvas for relaying societal myths, a stage for the theater of everyday life.

Although hardly acknowledged at present in mainstream 'enterprise'-architecture, this is enormously important: story is *emotive*; story embeds *meaning*; story *engages*. Stories *matter*: in a very real sense, everything about the architecture is or represents or describes a story. Even the enterprise itself is a story[15]. Which means that it's well worth while to go 'looking for the story', much like a journalist or filmmaker or any other storyteller would.

What I usually do for this is 'go for a walk', either metaphorically via the website and intranet and so on, or – preferable – literally go for a walk, in person or with one of my new 'allies', around some of the organisation's sites and spaces, looking for all of those interweaving stories that hold everything together. Some of these stories are straightforward enough: every transit through a business-process is a story; every customer-experience or 'value-journey' holds a story; every transaction is part of a story that extends far beyond the transaction itself.

Yet there are also the many stories that employees and others tell themselves, and tell each other, about what works, about what doesn't, about what is or isn't valued in practice within the organisation, about workarounds or special-cases that no-one's gotten round to documenting but without which the store or warehouse or whatever won't work. Those stories are often *really* important from a structure-perspective, too.

[14] http://weblog.tetradian.com/two-povs-on-ea/

[15] http://weblog.tetradian.com/the-enterprise-is-the-story/

And there's the story – or stories – that the organisation tells about itself, about how it positions itself in the market, about what it values most and would most like to share with others; and the stories that others in turn tell about the organisation – including whether they believe that the organisation holds to its purported values. Those last stories are some of the most essential real-world feedback for strategy – which in turn feeds back into changes in structure, in the what and how and where and when of the conventional enterprise-architecture.

Best stop there for now, I guess. But, yes, starting a new architecture is always a real challenge – yet always an interesting and worthwhile one!

I hope this has been useful, anyway: and good luck!

Source (Tetradian weblog)

- *Date*: 2011/10/24
- *URL*: where-do-we-start-with-ea-a-practical-question[16]
- *Comments*: 2
- *Categories*: Business, Enterprise architecture
- *Tags*: Business, business architecture, effectiveness, enterprise, Enterprise architecture, methodology, narrative, values

[16] http://weblog.tetradian.com/where-do-we-start-with-ea-a-practical-question

How much should an enterprise-architect know and do?

A great question this morning from Australian enterprise-architect Mike Aikins[17] (@AussiMike[18]):

> Noted some of your comments recently on how you feel more of a facilitator than the creator of an EA when working with a client. Question is how relevant/important is it for the consultant EA to have deep industry/sector knowledge in order to fulfil that role?

There are actually two questions there:

- To what extent should EAs themselves aim to design an architecture for the enterprise?
- Either way, how much industry/enterprise knowledge does the EA need?

The two questions are closely related, but it's useful to answer them separately and then link them together afterwards.

How much should an EA aim to 'architect the enterprise'?

Architecture is both descriptive and prescriptive. Although architecture itself must always face toward the 'big picture' view, there's also always a *large* component of real, practical, concrete design –

[17] http://mikeaikins.squarespace.com
[18] http://twitter.com/AussiMike

because architecture only becomes useful when it *does* touch the real world. (We've all seen plenty of 'concept architectures' that would be no use at all to any real person – and that applies to building-architecture as much as it does to EA!) So an architect is also always a designer – a *creator* of what people then experience as 'architecture' in the real world.

But there's an interesting trade-off here. The clients *must* always be not merely involved, but deeply *engaged* in the design. If that doesn't happen, they won't feel that they own it ('own' as personal responsibility, that is, rather than mere possession). And if they don't feel that commitment towards it – that it is *their* choice, *their* creation, rather than something imposed on them – the structure will fail, if only because they'll find themselves fighting against it in all manner of small subtle ways, consciously or not. Or, to put it the positive way round, the architecture will work best when the client feels that it expresses who they *are*, how they *relate*, what they *know*, what they *do*, in their own concrete, practical terms. To make that happen, the architect needs to elicit all of those things from the clients – and hence does need to be a firm yet genuinely humble facilitator.

At the same time, each architect does need to express their own choices in the architecture: every building by Gehry or Gaudi, Frank Lloyd Wright or Charles Rennie Mackintosh, is instantly recognisable as such. So the opinions and politics and worldview of each architect do also matter: which means that, especially as an external consultant, we do need to ensure that our views do align reasonably well with those of the respective clients, to ensure that the inevitable gaps can be bridged enough to make the architecture work. (This is one of the key reasons why IT-centric 'enterprise-architecture' so often fails: business-folks rarely appreciate IT-types trying to redesign the business world to fit into their own rather restricted image. :-)) This is more about empathy than sympathy: we need to be able to listen, to respect the clients' knowledge and desires, to yield when appropriate; yet also able to respect our *own* knowledge, and to know when to stand our own ground. What we know and how we express our vision *does* matter – and that's precisely why the client employs us, after all.

Which brings us to the second question: how much do we need to know about the business itself?

How much industry/enterprise knowledge does the EA need?

Obviously, there's another interesting trade-off here, because what we call 'architecture' is actually a complex mix of big-picture aspiration and real-world design. To put it at its simplest:

- design depends on domain-specific *specialist* knowledge – lots of it
- architecture depends on link-between-domains *generalist* knowledge – lots of it

So we need both types of knowledge – and lots of both, which is why it takes a long time to become competent as an architect. But domain-specific knowledge is *relatively* easy to acquire: almost all education and almost all organisational structures push towards specialisation of some form. So to balance that, ***the architect must be a consummate generalist***. You need to be able to learn the rudiments of a domain or a business very fast indeed – sometimes mere minutes may be all that you'll have, in which to get something both valid and usable enough to work with. Even more, you need to be able not only to grasp the 'world' of each specialist, and thence to converse intelligently and usefully in their own specific terms, but also to link all of the 'worlds' together in new, more effective ways. We need very strong people-skills, to be able to engage the attention and commitment of people in domain and at every level, from the cleaners and call-centre workers right the way up to the boardroom (who sometimes seem to have little awareness of what their cleaners and call-centre workers actually *do*…). The specialists often won't know how their worlds connect with others, if at all, so they won't be able to help you much in that: it's up to you to understand the whole *as* a whole, and make it work well for everyone.

So to reply to Mike's original question, the short (and unhelpful!) answer is "it all depends", because there's yet another huge trade-off here. The reality is that there's a limit to how much any one person can know, which leads to two very different types of EA roles:

- the *internal consultant*, with in-depth knowledge of the organisation
- the *external consultant*, with in-depth knowledge of the world *beyond* the organisation, including the EA discipline itself

The internal consultants' value lies in what they know (sometimes even more in *who* they know) of their own specific business context; paradoxically, the external consultants' value often lies in what they *don't* know, and in the sometimes 'stupid'-seeming questions they ask so as to discover what they need to know. External consultants can challenge an organisation's assumptions and 'givens' with far more licence and freedom than most 'insiders' would have; 'insiders' know the organisation's deep culture in ways that would never be available to any 'outsider'. Somehow we need to balance the two – the worst balance being where a closed group of outside specialists create 'the architecture', and then walk away, leaving the organisation with no architecture capability of their own and no way to *use* the work that's been done. (That seems to be a common tactic amongst the 'big-name' consultancies: it delivers minimal real usable value to the client but creates long-term 'consultant dependency' – which may be a nice way to milk the client for fat consultancy-fees, of course, but seems little better than fraud, in my opinion… :-|).

Most of my own work is in the 'external consultant' role, creating context and capability. I've done a certain amount of 'inside consultant' work in my time, but mainly enough to gain deep respect for the fact that it takes *years* – decades, even – to build up the knowledge and connections enough to do real whole-of-organisation architecture from the inside. So for most of my clients, my real value is not that I know their business in detail, but that I can learn *enough* detail fast, and connect that to the whole of the extended-enterprise within which their own enterprise (organisation, business-unit, domain, whatever) will operate and exist. (See my presentation '*What is an enterprise?*[19]' for more on this.) Here I perhaps need to emphasise two key points:

- the relevant enterprise is always larger than the nominal organisation in scope
- an organisation is bounded by *rules*, whereas an enterprise is bounded by *shared commitment*

Which means that whatever type of 'enterprise architecture' we do, we need to know a lot more than just our own scope. IT-infrastructure architects need to understand the applications and data that will run in their infrastructure; data-architects need to understand the business-use of that data as information and knowledge for decision-support; business-architects need to understand the broader enterprise, both horizontally (partners, supply-chain etc) and vertically (market, clients, prospects, non-clients, anti-clients, social context etc). The in-depth knowledge of our own domain is (relatively) easy to obtain; it's going outside our own scope that's a lot harder, simply because so much of it is literally 'alien'.

As a consultant EA, I need to be able to translate the strangenesses of those 'alien worlds' into something that makes practical sense for my clients. I have to make those 'alien worlds' seem safe for them, too. And I need to know all of it well enough not to make

[19] http://www.slideshare.net/tetradian/what-is-an-enterprise

any serious mistakes! An internal-EA's knowledge is usually design-focussed, literally into the depth of the detail; an external-EA's knowledge is *necessarily* far more generalist – the opposite of 'depth', in a sense - looking outward, making connections, drawing analogies and innovations from every other available discipline and domain.

So how much knowledge – and what knowledge – do we really need?

A good specialist can describe and deliver 'best-practice' for the industry. As an architect and a generalist, I need to understand what 'best-practice' looks like at the present – hence, yes, I do need in-depth knowledge of the industry, or at least know how and where and from whom I can acquire it fast. But I *also* need to be able to describe and deliver far *more* than existing 'best-practice' – in fact something that will not only deliver 'even-better-practice' now, but will continue to elicit new improvements to overall effectiveness onward into the future. To do that, I sometimes need to *deliberately 'forget'* all of what I know about current 'best-practice' in the organisation and industry – because the broader enterprise often has different ideas, and better ideas at that!

To constrain the amount of needed 'depth-knowledge' to a level that's achievable, we can usually set the scope-boundaries to those of the broader enterprise – again, always at least a couple of steps larger than whatever our own 'enterprise' may be. If we're doing business-architecture for a brewery, for example, we obviously need to understand our own business-drivers and internal business-context. We need to understand the drivers and context of our immediate market: clients such as shops and bars; other brewers and other direct competitors; 'up-side' supply chain such as grains, containers, energy, water; 'down-side' supply-chain such truckers, warehouses, distributors – in other words, all the usual interweavings of the transaction-economy. But we *also* need to understand what's happening *beyond* our immediate market, especially where it interweaves with the attention-economy and trust/reputationeconomy: hence the importance of non-clients, anti-clients, other intersecting service-providers such as police, schools, medical services, and the community in general. What are some of the entirely different forms of social-entertainment that could sideline beer entirely? Or non-social entertainment, or even non-entertainment, such as the potential impact of fundamentalist religion? If we remain solely introspective, looking only at our own immediate world ('the competition' and so on), we can't complain if our 'enterprise' is suddenly overwhelmed by a tsunami of change that could have been entirely expected – and architected for – if only we'd had the sense to look out to sea…

So to come back to the original question again, the short answer is yes, we do need "deep industry/sector knowledge", to fulfil the design side of the architect's role. But in practice, we probably need less of that in-depth knowledge than you might expect, because there are plenty of specialists who can give us everything we're likely to need. Instead, what we probably need much *more* of is '*in-breadth*' knowledge – because that's what the *architecture* side of our work needs most.

Hope this make sense, anyway – and thanks again for the question!

Source (Tetradian weblog)

- *Date*: 2010/01/13
- *URL*: what-should-ea-know-and-do[20]
- *Comments*: 8
- *Categories*: Business, Enterprise architecture
- *Tags*: Business, business architecture, enterprise, Enterprise architecture, skills

[20] http://weblog.tetradian.com/what-should-ea-know-and-do

Pinball-wizard

Chatting with a colleague the other day about where I'd gotten to so far with my explorations on new toolsets on enterprise-architecture[21]. I'd been talking on what I saw about what it needed to do overall; then a bit about the range of devices it needed to be accessible on; then a bit more about HTML5 frameworks for device-independent design; and then on into some of the finer points about the trade-offs between MySQL[22] versus SQLite[23] versus MongoDB[24] versus Neo4J[25] and suchlike as suitable data-storage. At that point he pretty much exploded: "That's not enterprise-architecture!", he roared. "Enterprise-architecture is big-picture, big design - you shouldn't be looking at any of that of low-level stuff!"

Yet when we talk with software-developers and suchlike, we get hit with the opposite kind of complaints: "If you're agile, who needs architects? Architecture's just Big Design Up Front, isn't it? - if everything's emergent, that kind of upfront thinking is just the wrong thing to do!"

As architects, looks like we're wrong pretty much whatever we do, doesn't it?

It's not quite as bad as that, though. It's not a simple 'either/or', with both sides wrong, and us as pig-in-the-middle: it's more about subtle shades of grey and suchlike. I think it was Gene Hughson[26] who once commented here that "Computers are binary, but people aren't!" - and neither is the real-world, for that matter. And certainly Gene's recent post 'Who Needs Architects? - When

[21] http://weblog.tetradian.com/new-toolsets-for-ea/
[22] http://www.mysql.com/
[23] http://www.sqlite.org/
[24] http://www.mongodb.com/
[25] http://neo4j.com/
[26] http://twitter.com/genehughson

Tactics Do Not Add Up To Strategy[27]' gives me some hope that architects *do* still deliver real value - as does Simon Brown[28] in his writings on software-architecture and the need for 'Just enough up front design[29]'.

But what *is* 'just enough'? How do we know when enough *is* enough?

Well, it depends, doesn't it? Which takes us right back into that loop[30] again… Sigh…

Maybe there's a need for a better metaphor here?

All of which kinda reminded me of that classic song by The Who:

(Photo by Donald Bell via Flickr)

[27] https://genehughson.wordpress.com/2015/01/03/who-needs-architects-when-tactics-do-not-add-up-to-strategy/

[28] http://twitter.com/simonbrown

[29] https://leanpub.com/software-architecture-for-developers/read#just-enough

[30] http://weblog.tetradian.com/an-ea-mantra/

Yep - enterprise-architect as **Pinball Wizard**[31]. :-)

Think of architecture as a game of pinball, where the aim of the game is not so much to rack up a numeric score, as to develop new understandings about the respective context. In classic pinball, it's a 'finite-game[32]': we *play to 'win'*, the score itself is the only thing that matters, and hence we always want to keep going for as long as we can. But in architectures, we know we're in a broader 'infinite-game[33]', in which we *play to learn*, and in which it's the *usefulness* of the understandings that matter most. Hence we'll play for long enough - *just enough time*, giving us a score[34] of *just enough understandings* - to give us what we need for the current task: at that point, we let the ball go, turn towards other tasks, and come back to play another game whenever we need it.

To develop understanding about some architectural concern, a domain-architect will bounce the focus around their domain-context quite a bit. No surprises there.

But the catch for enterprise-architects is that, by definition, enterprise-architecture must cover *every* domain in the respective enterprise - which means that we'll likely find ourselves bouncing around a *lot*. **Everything depends on everything else**; and **everywhere and nowhere is 'the centre', all at the same time**. Everything is just as important as everything else: the architecture can be made or broken not just by the high-level ideas and decisions in the abstract but by practical constraints around what's achievable right here, right now.

(Getting that balance of 'just enough everywhere' right is crucially important. To understand that point, we only need to look at what happens when the balance *isn't* right, and something crucial gets missed out of consideration, often at the smaller scale: Le Corbusier's infamous irreparably-leaking roofs, for example; Frank

[31] http://en.wikipedia.org/wiki/Pinball_Wizard

[32] http://en.wikipedia.org/wiki/Finite_and_Infinite_Games

[33] http://en.wikipedia.org/wiki/Finite_and_Infinite_Games

[34] http://weblog.tetradian.com/checking-the-score/

Lloyd Wright's maintenance-nightmare of the Marin Civic Centre; Norman Foster's boxy, echoing airport-terminals that somehow leave no place for *people*; Rafael Viñoly's curved 'Walkie-Talkie' office-building as unintended solar-furnace - the list goes on and on...)

That's what creates the *need* for that bounce, the pinballing-around, from high-level to fine detail, up, down, sideways, round. It's also where the challenge lies for the enterprise-architect - and doing it all in a viable, *usable* timescale, too. To paraphrase that song's lyrics[35] somewhat:

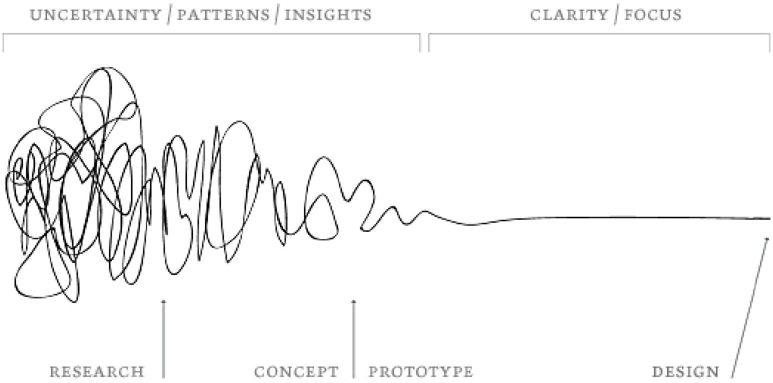
> a pinball wizard, it's got to be a thrill
> a pinball wizard, it's such a subtle skill

Rack up that score! Build the understanding! Cling! Flash! Go! Don't let that ball drop until you *want* it to drop!

There's another reason for the bounce, too. Sitting up in an ivory-tower somewhere, all just Big Ideas, without testing anything, without any contact with the real-world? - yeah, that'll likely lead to the more disastrous end of Big Design Up Front, all right. Or stuck in Analysis Paralysis. But if we also get *out there*, allow ourselves to bounce around the context-space in whatever way feels right - "he plays by sense of smell", as the song puts it - we can then experiment, test, compare and contrast, link big-idea to real-world practicality, use real-world innovations to inform big-ideas again, bounce, bounce, up, down, sideways, round. And that's when things start to gain a better chance of becoming *real*. Of *being* real, too. The point here is that the real process of developing new ideas is inherently messy - as illustrated so well by Damien Newman's 'the Squiggle[36]':

[35] http://www.azlyrics.com/lyrics/who/pinballwizard.html

[36] http://v2.centralstory.com/about/squiggle/

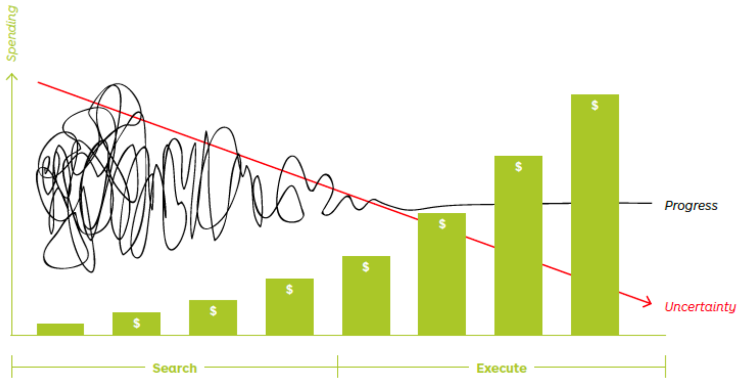


(Graphic by Damien Newman)

(This also illustrates again why so many of us get so frustrated by the existing ‘enterprise-architecture’ toolsets: for the most part, most are only of much use right over at the far end of the Squiggle - not in the beginning or the middle, where we most need their help!)

In a recent Tweet, Alex Osterwalder[37] took the Squiggle one step further, cross-mapping it to the risks and costs of experimentation, in effect keeping the cost (in every sense) of experiments inversely proportional to the level of uncertainty:

[37] http://twitter.com/AlexOsterwalder



(Graphic by Alex Osterwalder, adapted from an original graphic by Damien Newman)

One of the key points about playing 'pinball-wizard', as an enterprise-architect, is that for those early stages most of the experiments are little more than 'thought-experiments' - at most a scribbled hand-drawn diagram, a few quick lines of code, or whatever the equivalent might be - that each cost next to nothing in time, money or whatever, but deliver useful insights that can then be applied and cross-linked *elsewhere* in the overall context-space. So yes, there were, for example, good reasons why I looked at those different database technologies for possible toolsets: I needed to keep hold of the big-picture focus, yet I also needed to understand what was (is!) feasible in terms of real-world support for those questions around identity, content, connection, change, navigation and search that we'd need for any *implementable* toolset. I do spend a lot of time up in the 'big-picture' domains, it's true: but all of that runs a real risk of a fall into myopic meaningless unless I do continuously connect it back down to the real-world as well.

Yet there's one further twist we need to note. When we play 'pinball-wizard', yes, we're the player, choosing which buttons to press, which flippers to flap, in order to keep the ball in play for as long as we need. *Yet we're also the metaphoric ball that's being bounced* *around in that context-space* - which means that *we* are the ones who need to take all the hits from those bounces, too. Hence kinda '*ouch!*' at times, as most of us know all too well… - but that's the way the game goes, isn't it? Oh well.

Clunk! Plang! Bzzz! Rack up that score! Ding-d-ding-ding!

Pinball-wizard indeed… :-)

Source (Tetradian weblog)

- *Date*: 2015/01/08
- *URL*: pinball-wizard[38]
- *Comments*: 3
- *Categories*: Complexity / Structure, Enterprise architecture, Knowledge
- *Tags*: decision-making, Enterprise architecture, metatools, modelling, sense-making, toolsets

[38] http://weblog.tetradian.com/pinball-wizard