

Tom Graves / Tetradian

Random ramblings over the metaphoric edge

[Home](#) [About](#) [Tom Graves](#) [Tetradian](#) [Publishing](#) [\(admin\)](#)

[Home](#) › [Business](#) › Towards a whole-enterprise architecture standard – 5: Practices and toolsets

Towards a whole-enterprise architecture standard – 5: Practices and toolsets

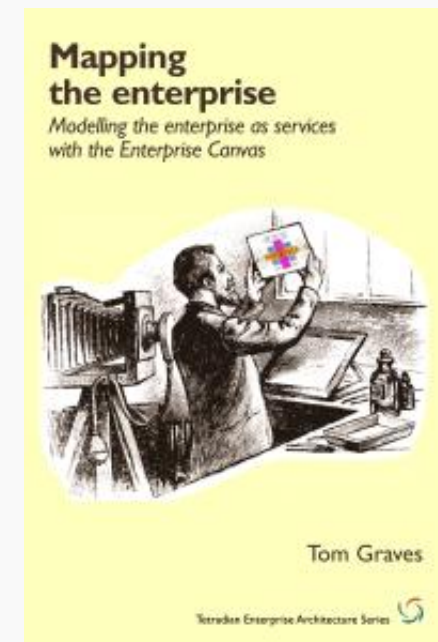
Posted on 9 May 2016 by Tom G — 1 Comment ↓

For a **viable enterprise architecture** [EA], now and into the future, *we need frameworks, methods and tools that can support the EA discipline's needs.*

This is Part 5 of a six-part series on proposals towards an enterprise-architecture framework and standard for whole-of-enterprise architecture:

1. [Introduction](#)
2. [Core concepts](#)
3. [Core method](#)
4. [Content-frameworks and add-in methods](#)
5. **Practices and toolsets**
6. Training and education

Books by Tom Graves



Mapping the Enterprise

This part explores how we put all of the ideas so far into real-world practice, using various reference-frameworks, checklists and add-in methods – as described in the previous part of this series – as ‘plug-ins’ to the core-method described in Part 3, in alignment with the core concepts described in Part 2. We’ll also briefly explore the roles that automated ‘EA-toolsets’ and repositories could play within the work, and what we need from those toolsets to do our work well.

Rationale

What do we *do* when we’re doing whole-enterprise architecture? How do we choose what to do, when, in what order? And how do we record what happens, the outcomes, the results?

Perhaps the core to all of this is the **‘Start Anywhere’ principle**, and the **focus on overall effectiveness of the enterprise**. Yes, the potential scope of whole-enterprise-architecture might at first seem impossibly huge: anything, anywhere, in any aspect or domain of the entire enterprise, and even beyond. Yet the crucial twist is that the enterprise is seen as an ecosystem, or ecosystem-of-ecosystems: whichever way we look at it, it’s always *one* integrated whole, deeply interdependent, deeply interwoven. In which case, *it doesn’t matter where we start*: if everything’s connected to everything else, then we connect with everywhere eventually.

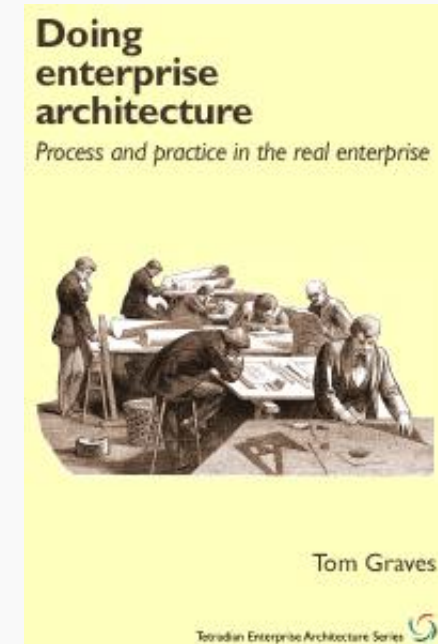
The **core-concepts** from Part 2 of this series allow us to describe the enterprise in fully-fractal terms.

The **metamethod** from Part 3 enables us to work in the same way, on any aspect of the architecture, at any level of detail, anywhere in the enterprise.

And the **metaframework** from Part 4 provides us with a means to use any context-specific framework or method, and still link them all together in a meaningful way.

So choose somewhere to start. Explore the architectures anywhere, address any

Ebooks by Tom Graves



EBook: Doing EA

Recent Posts

- [Towards a whole-enterprise architecture standard – 6: Training](#)
- [Towards a whole-enterprise architecture standard – 5: Practices and toolsets](#)
- [Towards a whole-enterprise architecture standard – 4:](#)

question. Doing so, and doing it well, with a solid understanding of enterprise-as-enterprise, will always enhance effectiveness *everywhere* in the enterprise.

Simple as that, really.

All we gotta do is start.

At which point, inevitably, someone says, “Yes, but where *do* we start?”...

To which the practical answer is “Look around”. We ***start with the business-question that’s in front of us right now***. Consider that mantra about effectiveness: “things work better when they work together, on purpose”. What is there around us right now that could usefully become better at that – or that could be more efficient, more reliable, more elegant, more ‘on-purpose’, more integrated with the rest of the enterprise? – some *practical* business-concern that, if we address it, will contribute in some useful way towards greater effectiveness overall for the organisation and the whole-enterprise. Preferably something small-but-useful at first, and, if we *are* just starting out, preferably also something that won’t matter much if we don’t get it right. But *that’s* where we start.

We ***start small-but-useful – and build outward from there***.

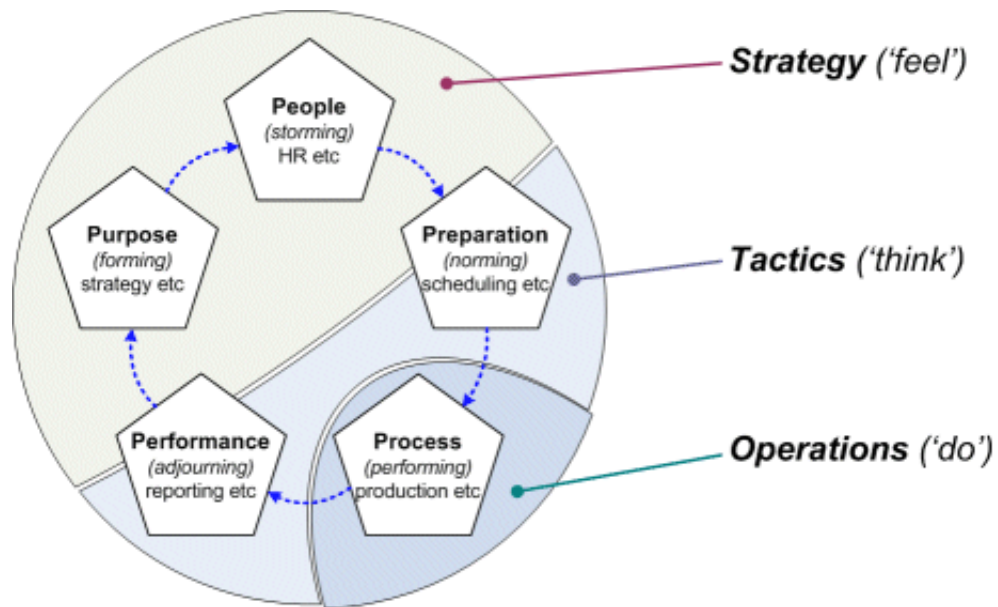
The trick here is that whatever we do, at whatever scope or scale, it’s ***always the same overall method***. Different in the detail every time, yes, sure, but always the same underlying metamethod:

Content

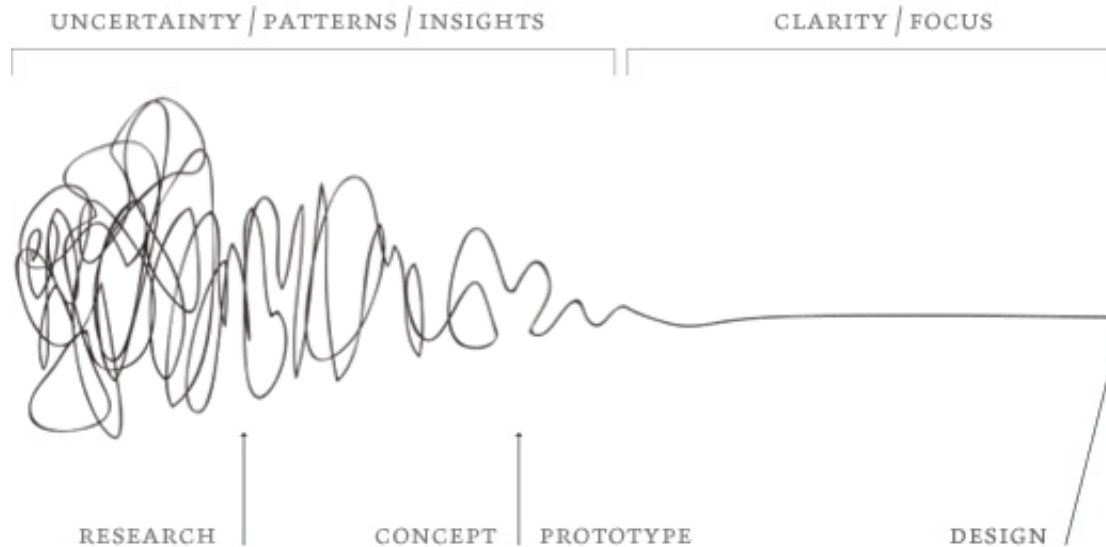
- [Towards a whole-enterprise architecture standard – 3: Method](#)
- [Towards a whole-enterprise architecture standard – 2: Core](#)

Recent Comments

- [Doug McDavid](#) on [Towards a whole-enterprise architecture standard – 5: Practices and toolsets](#)
- [4 Layers Of Enterprise Architecture | Architecture Fan](#) on [On layers in enterprise-architecture](#)
- [Tom G](#) on [Towards a whole-enterprise architecture standard – 3: Method](#)
- [Tom G](#) on [Towards a whole-enterprise architecture standard – 3: Method](#)
- [Jerome Abudah](#) on [Towards a whole-enterprise architecture standard – 3: Method](#)



And beneath that, what we're actually working on each time is the Squiggle:



...via a crossmap that looks something like this:

Categories

Select Category

Tags

anarchist **Business**
business-IT divide
business
architecture
chaos **complexity**
conference culture cynefin
decision-making disruption
economics
effectiveness
enterprise
enterprise 2.0
Enterprise
architecture
enterprise canvas **Futures**
IT-centrism **Knowledge**
metamodel methodology
mythquake narrative
narrative knowledge Open Group
paradigm power quality
RBPEA responsibility
SCAN Scribbles / writing



There's more detail on this in the post '[Selling EA – 3: What and how](#)'. Or, if you're more familiar with the TOGAF ADM (Architecture Development Method), we could crossmap that to the method here as follows:

- **Purpose:** identify business-question, scope and scale (via metaframework 'tags'), and success-criteria for this iteration
[TOGAF: ADM Phase A is approximately equivalent, but you'll need to explicitly include the business-question, scope, scale or success-criteria, and also beware of TOGAF's tendency to impose IT-centrism by default]
- **People:** identify and engage with all relevant stakeholders to address the business-question
[TOGAF: there's no distinct ADM phase for this, but it's loosely covered in TOGAF ch.24, '[Stakeholder Management](#)']
- **Preparation:** assess and plan for architecture action, in context of the business-question, scope, scale, success-criteria and stakeholders
[TOGAF: do the same tactics as described in ADM Phases B, C or D - current assessment, future needs, gap-analysis - but adapted for this iteration's business-question rather the ADM's hardwired scope and scale, and with extra attention on stakeholders and success-criteria]
- **Process:** enact the plan – usually with recursive review and re-plan (though *during* action itself there may be no time for real-time review)
[TOGAF: essentially the same as for ADM Phases E/F/G - though note that

sense-making

service-oriented enterprise skills

social media Society story

strategy taxonomy togaf

twitter values worldview

Archives

Select Month

Phases E/F incorporate design elements that would be [more typical of domain-architecture](#) or, more, solution-architecture than at whole-enterprise-architecture level]

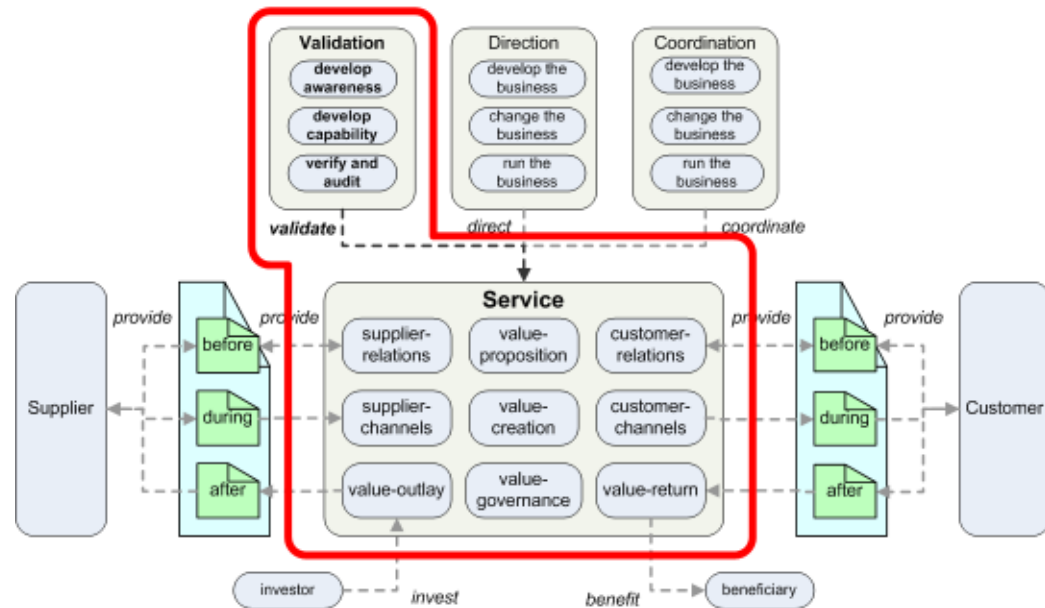
- **Performance:** assess business-outcomes, benefits-realisation and lessons-learned (linked to the initial success-criteria from *Purpose*), and provide setup for any future architecture-cycle

[TOGAF: explore the technology-landscape as for ADM Phase H, but also add much more attention on business-outcomes, benefits-realisation and lessons-learned]

And also:

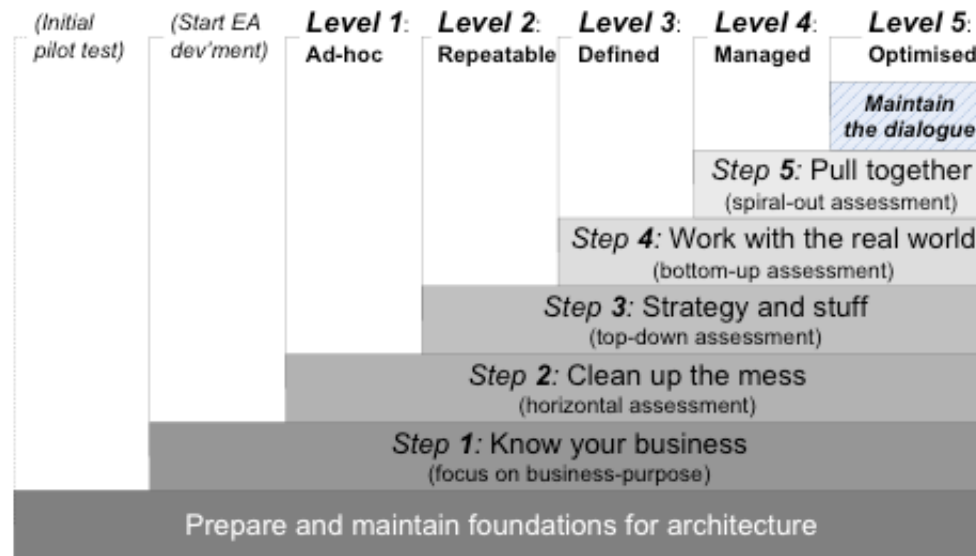
- **Recursion:** explicit support for full fractal recursion in any direction and to any depth
[TOGAF: expect to use the 'loop-back' recursion supported in the ADM, but more fractally, and also often well beyond the constraints of the ADM's hardwired scope]
- **Repository:** explicit requirement for a shared storage-area for documents, discussions, whiteboards, requirements, diagrams, models, and anything else needed to guide and govern architecture-development – though may be implemented in any appropriate form
[TOGAF: although the ADM diagram constrains the repository to [requirements only](#), not that the ADM Phase descriptions describe more detail about what needs to go into the repository, and see also TOGAF's description of the role of [toolsets](#) for [architectural artefacts](#)]

For each success-metric in the current-iteration's purpose, and for each principle or value in the organisation and the overall shared-enterprise, there needs to be some means to gather and verify content about that metric, or alignment to that principle or value. That's how we 'close the loop' to ensure overall effectiveness. There will often need to be distinct services that do this, such as those modelled as '[validation-services](#)' in service-oriented modelling with [Enterprise Canvas](#):



(In TOGAF, it's possible to describe some types of enterprise-values via the optional 'Motivation plug-in' in the TOGAF [Content Metamodel](#), but there seems to be no indication of how these would be used in architecture practice. There's a useful section on how to specify [architecture-principles](#), but again there's no real indication of how to use them – and the examples are essentially for design-level IT only.)

To guide choices about what to work on and where to start, we can also use the [Maturity Model](#) summarised in [Part 3](#) of this series:



This provides a recommended order in which to do architecture-development. It isn't essential to develop the architecture in this sequence – in fact real-world exigencies and urgencies will often force us to do otherwise. But we do need to be aware that whenever we *do* do anything out-of-sequence, it's likely to increase what's known as '**Enterprise Architectural-Debt**', which we definitely need to manage in a conscious way.

- Step 1 (*'Know your business'*): Identify the core 'story' that underpins the shared-enterprise and the organisation's relationship with it; also the capabilities and resources that the organisation can bring to bear on this story.
- Step 2 (*'Clean up the mess'*): Identify possibilities to increase efficiency and reduce Enterprise Architectural-Debt.
- Step 3 (*'Strategy and stuff'*): Identify strategic opportunities, and how to implement the requisite underlying support for execution of those strategies.
- Step 4 (*'Work with the real world'*): Identify and assess options arising from 'out there' or from the front-line, and take action to implement support and promulgation of those changes as appropriate.
- Step 5 (*'Pull together'*): Identify, assess and address business 'pain-points' at any level of the organisation and its operations, or anywhere in the broader shared-enterprise.

Whatever business-question we're asked to work on, we should:

- keep aware at all time of the **core-concepts** outlined in [Part 2](#);
- use the **metamethod** outlined in [Part 3](#);
- and use that method with **context-specific content-frames and plug-in methods**, guided by the **metaframework** as outlined in [Part 4](#).

Remember, though, that the method is recursive – we need to keep track of how far down the rabbit-hole we go!

To keep the length of this post down, I won't give a full worked-example here, though I'll probably do an add-on post to describe this. In the meantime, see the post '[Selling EA – 1: What do EA clients want?](#)' for some 30+ examples of real business-questions that we might be asked to address in whole-enterprise architecture. (Note that all of those examples are at fairly high-level, and all organisation-oriented: in practice we would just as often be asked to address that are 'upward' at the whole-enterprise level – such as [anticlient concerns](#), '[Pass The Grenade](#)' risks or [hidden dangers in business-model designs](#) - or 'downward' to detail-level concerns in technology and elsewhere.)

The other major concern here is about **toolsets**.

As described in Part 3, the architecture-metamethod depends to a significant extent on being able to store and retrieve information in some kind of shared-repository, to pass between iteration-phases, between iterations, and for future review. I often describe the content of this repository as a '**hologram**', built up over time via interlinks between many different types of content, including:

- whiteboards and hand-drawn diagrams
- handwritten notes or journals
- photos, videos or audio-recordings
- URLs, index-references and other content-pointers
- spreadsheets, databases and analytics data

- models, auto-generated diagrams, simulation-configurations and other electronic elements

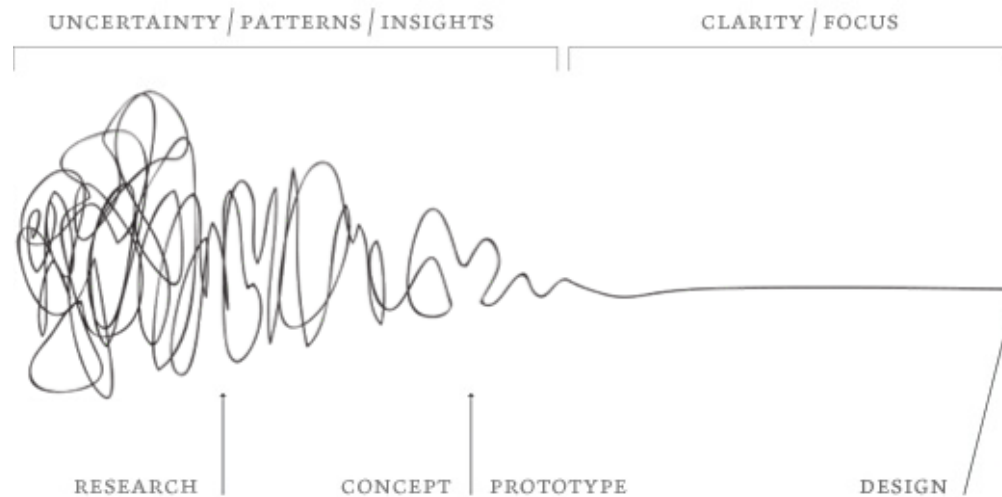
We also need our toolsets to help us keep track of where we are in our explorations, and prevent us from falling too far down the rabbit-hole of recursion. That's important.

In short, what we need most of all is that ***our toolsets should provide practical help in all aspects of the work we do as architects***. For example, perhaps the hardest part of the architecture-process with which we need help is in the complexities of the People phase of the architecture – otherwise all-too-accurately known as the 'Storming' phase. As [JB Sarrodie](#) put it in a Tweet the other day:

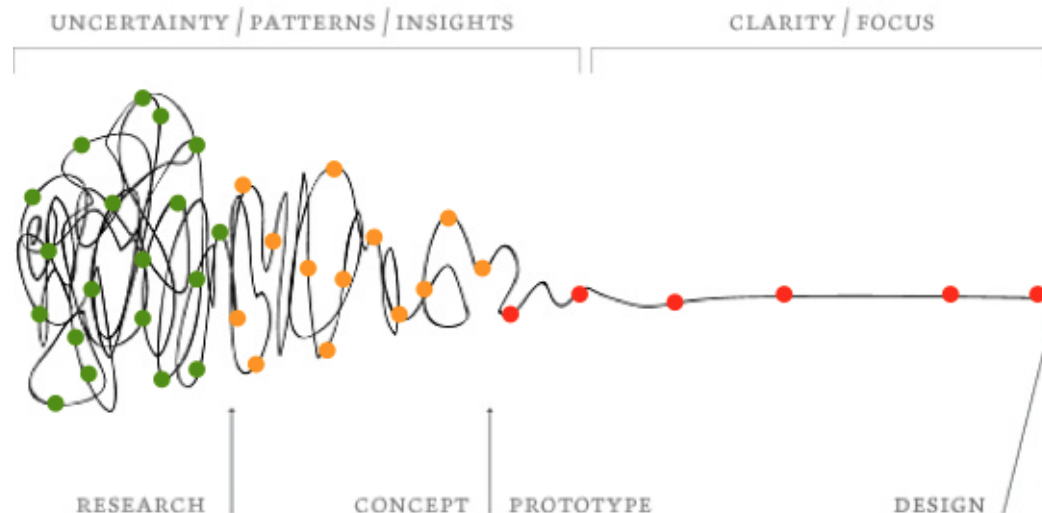
*“Too many think an #entarch produces nothing if he just speaks with people, while it's the whole point [of the work].
The final document (if needed) is not the end, but the by-product we offer when all the real work is done.”*

(There's more on this in the slidedeck '[Attracting, retaining and getting the best from your architects](#)'.)

And we ***need that help in a fully joined-up form***. If we look at this in terms of the Squiggle:



...then ***we need toolset-support along the entirety of the Squiggle*** – not merely one or two arbitrarily-selected subsets of it. But whilst it's probably too much to expect that a single toolset that would ever be able to cover *all* of the Squiggle, what tools we have at present seem to be scattered almost at random along its metaphorical length – and almost none of them can [work in a joined-up way](#) with any others at all...



Even worse, almost all that any of our so-called 'enterprise-architecture' toolsets will help us with at present is JB Sarrodie's mere 'by-product' of all the largely-

- ‘[The toolset-ecosystem](#)’ – from multi-screen situation-room right down to handheld dumb-phone
- ‘[Platform, toolset and tool](#)’ – how all of this can link together to create a new type of toolset

The reality is that ***there’s a huge market for toolsets that can handle those requirements in the way that we need*** – and yet it often seems that no-one has even *started* to tackle that market yet. The first toolset-vendor who really gets what I’ve been saying here will probably gain huge ‘first-mover’ advantages on a truly vast yet as-yet untapped market – a point worth noting, perhaps?

Application

From the above, we can derive suggested text for the ‘Practices and toolsets’ section of a possible future standard for whole-enterprise architecture. Some of the graphics above might also be included along with the text.

– *Scope of practice*: The practice of whole-enterprise-architecture may address any business-question applicable within any or all of the entirety of a chosen shared-enterprise. (Note that enterprises may be subsets, supersets or intersecting-sets with any other enterprise, so, in an extreme case, the effective scope may be potentially infinite.)

– *Nature of practice*: Practice should be based on the fractal, recursive metamethod outlined in Part 3 of this series, optionally using other content-frames and plug-in methods as guided by the metaframework outlined in Part 4 of this series, and in accordance and alignment with the core-concepts outlined in Part 2 of this series.

– ‘*Start Anywhere*’ principle: If the enterprise is ‘an ecosystem with a purpose’, then by definition everything in that enterprise is interlinked with and interdependent on everything else within that enterprise. Hence to improve the effectiveness of the enterprise, we can *start anywhere*: it shouldn’t matter where we start, as long as we *do*

get started.

- ‘*No Endpoint*’ principle: A counterpoint to the ‘Start Anywhere’ principle, there is no start to the enterprise and its architecture, yet also no end – we can *always* make the enterprise more effective, we can *always* do it better.

- ‘*Usefulness*’ principle: Given that the scope for whole-enterprise architecture is potentially infinite, and that both start- and end-points for architecture practice are inherently arbitrary, the practical constraint on any item of architecture-work is its *usefulness* – that it contributes usefully to the overall architecture within a reasonable expenditure of effort and time. The typical guidance for this is that each item of architecture-work should address a specific *business-question*, each indicating its own scope, scale, success-criteria and suchlike. Timescales for such work may vary from minutes to years, dependent on complexity and need, guided by the ‘Just enough’ principle of the [Enterprise-Architects’ Mantra](#).

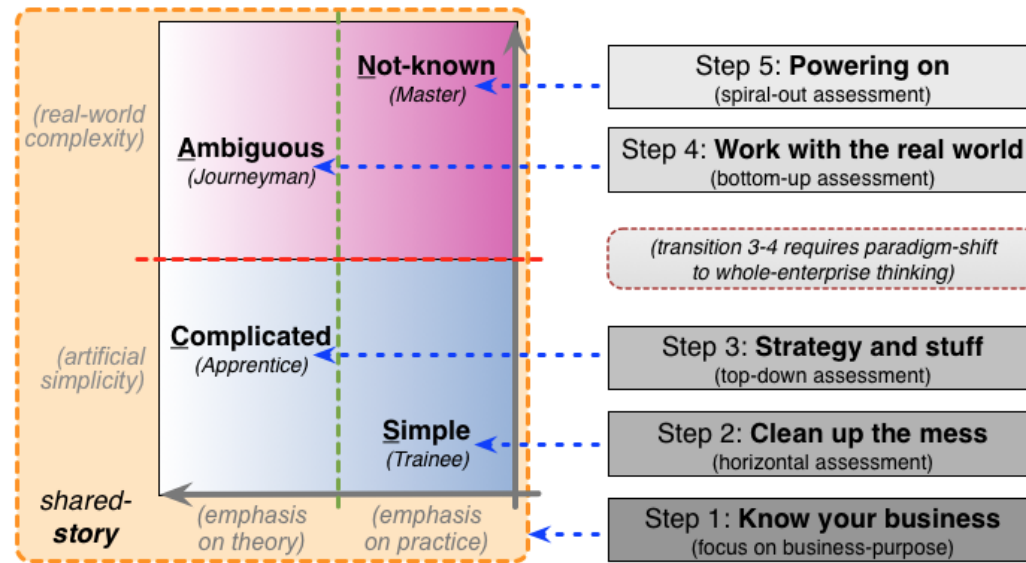
- ‘*Hologram*’ principle: If the enterprise represents an interlinked, interdependent ecosystem, then any enterprise-architecture work should reflect those interlinks and interdependencies. In doing so, such architecture work will, over time, create a kind of ‘hologram’ of the enterprise, within which any new item of work will be able to stand on its own, yet also add in some way to the detail of the descriptions of *every* part of the enterprise. (One aspect of the metaframework outlined in Part 4 is that its structure will inherently support such a ‘hologram’ concept of the enterprise.) In this way, the architecture-hologram should be able to support *any* type of view or sub-model of any aspect of the enterprise, to the available level of detail currently available within the hologram-model.

- ‘*Dynamics*’ principle: Whole-enterprise architecture takes place within a context that is inherently complex in many different ways, all undergoing continual dynamic change, at many different, interleaving rates of change. In such a reality, it is essential to work *with* such inherent dynamic-instability, rather than pretend that it does not exist: for example, it needs to be understood that concepts such as ‘current-state’ or ‘future-state’ are, at

best, *useful-fictions*.

– *Nature of required skills*: Whole-enterprise architecture is dependent on a specific set of skills – in particular, on *cross-disciplinary generalism*, and ultimately on the skills of *those who specialise in 'being a generalist'*. Since everyone is responsible for the architecture of their enterprise, everyone will need *some* awareness of the architecture and how best to support it within their everyday work: these skills are minimal, yet extremely important. At the other extreme, the skills and experience needed to operate at a professional level and at a full whole-of-enterprise scope are very extensive indeed, and will typically take decades to acquire – though acquired via any number of potential routes. (Requirements for training and education will be outlined in Part 6 of this series.) The key danger to the architecture is '*anything-centrism*' – most often caused by '*specialism-trolls*', who persistently attempt to elevate the importance of their own domain arbitrarily above all others.

– *Role of maturity-model*: Given the complexities of the work, a maturity-model may be useful to assist in guiding priorities for architecture-work, and also to assist in skills-development by prioritising work in terms of likely development of architecture-skill over time. The maturity-model summarised in Part 3 and Part 4 of this series can assist with both of these requirements:



– *Role of toolsets:* The architecture-metamethod expects and requires some kind of repository within which to store information both to guide the work and arising from the work, and ideally also to support guidance of the work itself. This repository may be implemented in any number of possible ways, from a handwritten notebook to a large purpose-built toolset. If at all feasible, tools and toolsets used for this purpose should be able to connect with each, to support the ‘hologram principle’ above. (It is unfortunate that the most of the tools and toolsets currently available for these tasks are woefully inadequate for the real needs of whole-enterprise architecture: it is to be hoped that this gap will be filled within the relatively near future.)

The above provides a quick overview of how we can guide whole-enterprise architecture practice, using the metamethod, metaframework and core-concepts described earlier in the series. Once again, I do know that it’s ‘blurry’ and ambiguous in many places, and nowhere near complete-enough for a formal standards-proposal, but it should be sufficient to act as a strawman for further discussion, exploration and critique.

The other remaining concern for a potential practice-standard is the resultant needs for training, education, certification and suchlike. We’ll explore those in the upcoming final part of this series.

Before we go there, though, any questions or comments so far?

Related posts:

1. [Towards a whole-enterprise architecture standard – 1: Introduction](#)
2. [Towards a whole-enterprise architecture standard – 2: Core](#)
3. [Towards a whole-enterprise architecture standard – 3: Method](#)
4. [Towards a whole-enterprise architecture standard – 4: Content](#)
5. [Work-in-progress – two more books](#)

◀ [Towards a whole-enterprise architecture standard – 4: Content](#)

[Towards a whole-enterprise architecture standard – 6: Training](#) ▶

Tagged with: Business, business architecture, complexity, decision-making, effectiveness, enterprise, Enterprise architecture, framework, Knowledge, metaframework, methodology, paradigm, sense-making, standards, strategy, toolsets

Posted in Business, Complexity / Structure, Enterprise architecture, Knowledge

One comment on “Towards a whole-enterprise architecture standard – 5: Practices and toolsets”



Doug McDavid says:

16 May 2016 at 4:41 pm

Maybe I have an old-fashioned hangup on well-crafted schemata. It seems that what you're proposing has more of the character of an EA data lake. Or maybe i'm misreading?

[Reply](#)

Leave a Reply

Your email address will not be published. Required fields are marked *

Name *

E-mail *

Website

Comment

Post Comment



