

Whole Enterprise Architecture

Exploring the context of an enterprise



SAMPLE COPY

Tom Graves

Whole Enterprise Architecture

Exploring the context of an enterprise

Tom Graves



Copyright © 2023 Tom Graves

All rights reserved. No part of this publication may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the publisher, except in the case of brief quotations embodied in critical reviews and certain other non-commercial uses permitted by copyright law. For permission requests, write to the publisher, addressed "Attention: Permissions Coordinator," at the address below.

Published by
JC3DVIS

www.jc3dvis.co.uk

First published November 2023
ISBN 9781739125455 (Paperback)
First Edition

Legal disclaimer

This book is presented solely for educational purposes. The authors and publisher are not offering it as legal, accounting, or other professional services advice. While best efforts have been used in preparing this book, the authors and publisher make no representations or warranties of any kind and assume no liabilities of any kind with respect to the accuracy or completeness of the contents and specifically disclaim any implied warranties of merchantability or fitness of use for a particular purpose. Neither the authors nor the publisher shall be held liable or responsible to any person or entity with respect to any loss or incidental or consequential damages caused, or alleged to have been caused, directly or indirectly, by the information contained herein.

CGI illustrations, design and editing by JC3DVIS

Whole Enterprise Architecture

Exploring the context of an enterprise

Tom Graves

Contents

- 1 | 1. What is Whole Enterprise Architecture?

Part 1: EA as a career

- 4 | 1. Creating a career in Enterprise Architecture
- 12 | 2. The unique contribution of EA
- 13 | 3. How much should an Enterprise-Architect know and do?
- 17 | 4. The deliverables of EA
- 18 | 5. How I do *Whole EA*
- 24 | 6. Guiding the dance of change
- 27 | 7. Architecture as boxes, lines and glue
- 29 | 8. Business architecture and EA
- 31 | 9. How do we make sense of EA?
- 33 | 10. The glue between strategy and execution
- 36 | 11. Two roles for Enterprise Architects
- 38 | 12. Analyst, anarchist, architect
- 41 | 13. Applied scientist or alchemist?
- 43 | 14. EA and solution-architecture
- 46 | 15. What futurists do, and why

Part 2: EA basics

- 52 | 16. Keeping EA simple
- 54 | 17. Innovation
- 56 | 18. Back to basics
- 58 | 19. What is architecture?
- 59 | 20. Vision, role, mission, goal
- 60 | 21. Inside and out
- 63 | 22. Capability
- 67 | 23. Fractals and EA
- 68 | 24. How to use the Five Elements
- 76 | 25. The relationship is the asset

Part 3: Why Whole EA?

- 82 | 26. Financial-architecture and EA
- 85 | 27. Digital transformation
- 88 | 28. Anti-clients
- 92 | 29. Designing for 'humans'

Part 4: Whole EA perspectives

- 98 | 30. Assumptions
- 100 | 31. Competition
- 104 | 32. Digital Transformation check-lists
- 106 | 33. Which EA tools to use?

Part 5: How not to fail in EA

- 109 | 34. Beware of 'policy-based evidence'
- 112 | 35. Certification

Part 6: Appendix

- 118 | 36. A simple check-list
- 124 | 37. What's next?

What is *Whole Enterprise Architecture*?

To some, enterprise architecture is a bit of a mystery. *“Something to do with IT and enterprises?”* It can seem full of jargon and complex diagrams. But at its heart is about making things better and working *‘on purpose’*. It is about connecting the dots and understanding the big picture.

In this book an **enterprise** is defined as:

A bold endeavour, an undertaking. For example, running an airport might be described as one type of enterprise, while the passengers’ enterprise would be to get to their holiday as quickly and safely as possible.

And **architecture** is defined as:

The structure and story of how everything works together as a whole. In an airport for example, a building architect would tackle the physical aspects of the building, the air-conditioning, doors etc. A solution architect would tackle the IT aspects of the airport, such as the software needed to run the airport, assisting: visas, luggage, information flow, people flow, governance and much more, while an enterprise architect would tackle all of the parts of an enterprise, connecting the boxes.

So **Whole enterprise architecture** is defined as:

Understanding the overall story of the enterprise about everything in the enterprise, not just one aspect such as IT. Understanding the connections, the *‘whys’*. Being able to see the big-picture, understanding the mess and unnecessary duplication. It’s about the context of the enterprise.

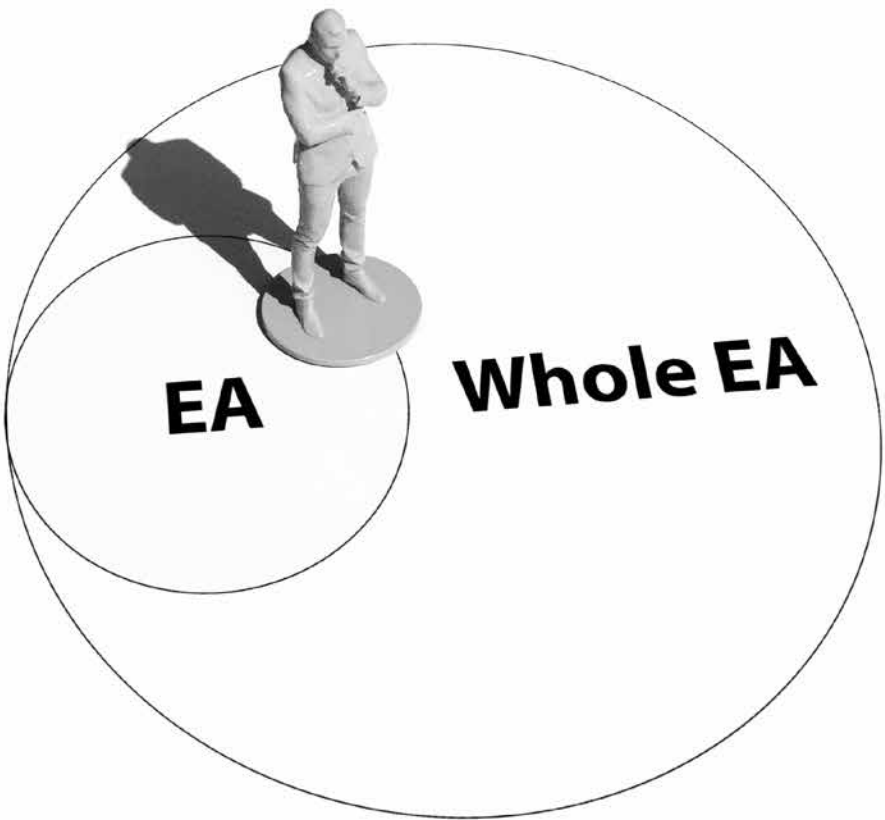
Throughout this book we will use an example of an airport to demonstrate various concepts, as it is a vastly complex enterprise, yet familiar enough to most people.

This book acts as a bridge to a huge library of work produced over thirty years in the field of enterprise architecture.

Most of the chapters reference more *‘in-depth’* articles which can be found in a set of anthologies **at www.leanpub.com/u/tetradian** .



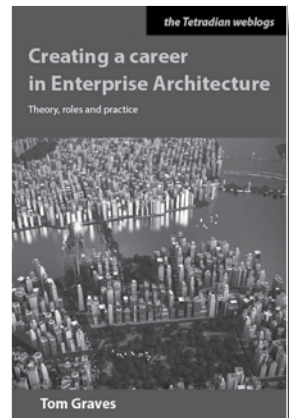
Tom Graves has been an independent consultant for more than four decades, in business transformation, enterprise architecture and knowledge management. His clients in Europe, Australasia and the Americas cover a broad range of industries including small-business, banking, utilities, manufacturing, logistics, engineering, media, telecoms, research, defence and government. He has a special interest in whole-enterprise architectures for non-profit, social, government and commercial enterprises.



Part 1:

Enterprise Architecture as a career

This section of the book is an abridged version of
'Creating a career in Enterprise Architecture'
www.leanpub.com/tp-career



1: Creating a career in EA

What is Enterprise-Architecture?

IT-architecture is a cross-disciplinary specialism: the enterprise IT architect will bridge between the various IT specialties, but the focus essentially remains centred around IT alone. By contrast, to the enterprise-architect, everywhere and nowhere is *'the centre'*: they must be a generalist, interested in everything. So I would encourage you to lift your eyes from the screen and the imaginary worlds within IT-systems, and look around you. IT systems describe a digital world, but they are also very physical: they exist in a real world beyond data alone. A real, messy, chaotic world, where computers need power and to be kept cool, placed somewhere safe from dust, rain and more. And a human world, where real people have real emotions and do the real work, and where passion for code (*and, sometimes, a confused passion for 'control'*) is what creates all of this in the first place.

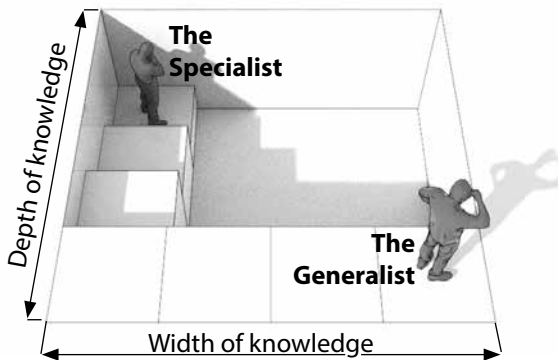
Within enterprise-architecture itself, look wider than just IT-oriented frameworks¹, to models² that encompass more of the *overall* enterprise. I would suggest to keep remembering that enterprise architecture literally means *'the architecture of the enterprise'*, not merely the architecture of the enterprise-IT. The IT exists within the context and needs of the broader organisation; and the organisation exists within the context and needs of a far broader shared-enterprise. An organisation is bounded by rules, roles and responsibilities, but an enterprise is essentially a human construction, bounded by aspirations, commitments, hopes, and fears. We create an enterprise-architecture *for* an organisation, but *about* that broader enterprise. And *'quality'* in all its forms is what arises from that broader enterprise: creating all those quality-oriented issues in architecture such as reliability, efficiency, safety, sustainability, security and more. If you remember to keep that idea of the broader enterprise in mind whilst you work on even the smallest item of code, it will help to show you what an *'enterprise'* really is, and so the nature of enterprise-architecture itself. The last point, perhaps, is to respect that all of this does take time, many years. But you're already a long way down that track: say, after more than ten years in *'the trade'* you will certainly have learned a great deal about the difference between academic theory and real-world practice!

1. Framework

A EA framework is a collection of methods, best practices, and tools to tackle a specific type of problem, often affected by change. An early example is DODAF, which in basic terms is a checklist of recommended items to explore and how to tackle them, such as unwanted duplication of equipment.

2. Models

For example in an airport, where experts model how the people flow through the airport, what order to do they do things, what depends on what and how they interact. In this example we are modelling the customer journey.



The Specialist versus the Generalist

Are you a specialist or generalist?

The relationship between specialist and generalist is rarely an easy one... but if you want to be involved in EA, then it's a relationship that you're going to need to resolve. And, perhaps most of all, resolve it within yourself...

Long ago, it was not all that unusual for an individual to become a '*complete generalist*': someone with deep skill in everything, or at least, everything that was known at the time.

But these days, no-one could do this: the vast scope of information, knowledge and skills that could apply in our world means that any one person would need many lifetimes to learn them all. And yet we have to be able to get things done right here, right now.

Think of this as two axes: horizontal, for width of knowledge, and vertical, for depth of knowledge. The usual solution is to specialise, and then perhaps to over-specialise. The danger of over-specialism, though, is that we risk losing the ability to '*connect the dots*': we get better and better at '*doing things right*', but we lose the ability to know if we're '*doing the right things*'. So we need some means to link all those specialists together: and that's where the generalist comes into the picture. ***Enterprise-architects are specialists at being generalists.***

In most current cultures, specialism is still prized far more than generalism. One of the reasons is that specialists visibly *do* things, whereas generalists don't seem to *do* much at all. It is the connections *between* things which are difficult to describe or to value.

To quote the *Tao Te Ching*:

*"...therefore profit comes from what is there;
usefulness from what is not there."*

No matter how useful the generalism of enterprise-architecture may be, the visible *'profit'* will usually *seems* to come only from the specialists. It is really important to recognise that ***specialist generalists are depth-specialists, they specialise in the skills required for broadscope generalism.*** The focus is not so much on content, as a discipline-specialist would, but on how different disciplines link together.

Some of the skills-challenges here include:

- thinking in multiple domain¹ *'languages'* at the same time, and translating between them as required
- thinking and designing in terms of interdependent systems rather than single independent specialities
- thinking in and working with multiple time-perspectives, in some cases ranging from sub-microseconds to millennia
- identifying, and rapidly learning, the key principles and practices of new domains, and requirements for and implications of linking between them
- clarifying and communicating contexts, constraints, designs and design-issues
- searching for simplicity
- mastering the *'soft-skills'*² needed for negotiation and suchlike, in what will always be a challenging and highly *'political'* area of work

One challenge will be around learning all those many different domains: *'just enough detail'* to be useful, yet no more than that, because you simply won't have time to do any more than that.

For every discipline that your work will touch, you'll need to learn enough about it not only to be able to converse credibly with any of the specialists in that area, but do so in ways that will enable you to make connections between all of those domains, connections that the specialists probably won't even know could exist. There are some personal challenges, too: such as the embarrassment of having to admit to others that *"I don't know"*, for example.

1. Domain

For some the meaning of the word 'domain' might be obvious, but for others maybe not. A domain might be described as an area of knowledge. For example an airport security expert's domain would be 'security'.

So domain language might be specific terms or shorthand, such as in cyber-security: 'Incident handling'.

2. Soft skills

Skills which are needed for all professions, such as teamwork, while 'Hard skills' are only required for certain industries.

Where do we start with EA?

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?

In essence, we start from scratch.

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

- the politics and pragmatics¹ 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. Probably the single most important concern is to get *'buy-in'* at senior level, certainly from the respective Chief executive officer for the main focus area (e.g. *the Chief information officer, for enterprise IT-architecture*), but preferably from the CEO and entire executive. If you don't have that *'buy-in'*, you'll be going nowhere: you need to get the executive on-side.

As others state, 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 context in which the organisation operates, and where the organisation fits within that context. 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. I usually look through sources such as the organisation's website, publications, advertisements, intranet and annual-reports. There's usually enough information there to build some preliminary models with which started: or at least, enough for people to tell us that the models are wrong, which is one way of getting them engaged in telling us their ideas about what it should be!

1. Pragmatics

Linguistics uses this complex term to broadly describe different methods of communication, including non-verbal communication.

For example a meeting can suddenly change its whole mood, depending on how one person might talk to another.

While we're doing this, we need to be looking for any allies, people who are already committed to other themes that connect with EA, and would be likely to see the value of connecting between those areas of interest. This is really important if we've only just started with the organisation, because 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 often take 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. 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. Other standards will be generic for the industry as a whole, but they're usually not hard to find.

What we are also looking for are all the other standards, guidelines and workarounds 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 would have little chance to know what these are.

We 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 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 to do architecture'* are probably the only part of the work that will remain unchanged: we need to be able and willing to challenge ourselves on just about everything other than that.

Almost all of that above is about enterprise-architecture as structure. The other side is about about architecture as story. 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. Which means that it's well worth while to go *'looking for the story'*.

I typically look for all of those interweaving stories that hold everything together. Some of these stories are straightforward enough: every journey through a business-process is a story; every customer-experience or 'value-journey'¹ 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 is not valued in practice within the organisation. About workarounds or special-cases that no-one has documented but without which the store or office would not work. Those stories are often really important from a structure-perspective², too. 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, how, where* and *when* of the conventional EA.

The need for a quick-start is very real. We need quick results, but above all we need to get the interest and, if possible, real excitement, going right from day one. We have to make sure that EA matters to everyone, in their context, their workspace, because without that engagement and excitement, this will go nowhere.

Do not try to start off straight away with any of the 'heavyweight' frameworks⁴. They do have very real value, in later stages of EA (*though often only in specific areas of EA*). For this earliest stage, we need something simpler.

1. Value-journey

With an airport example, from the customer's perspective the whole airport is only a small part of their value-journey. For them they want to spend as little time as possible in the airport, as their destination is what is important to them.

2. Structure³-perspective

In the structure of an airport, if someone wanted to improve security checks, exploring all the elements such as x-ray scanners and software currently used. From this one could begin to work out what might need upgrading, for example.

3. The 'structure' of the architecture of the enterprise.

In an airport example, the structure might be made of parts, such as security, which works in a certain way and slots together with other parts in the airport.

4. Heavyweight framework

A framework which tries to give every type of detail, but it won't work in every context, while lightweight frameworks are often a set of check-lists.

I usually describe EA, development in terms of six distinct steps:

Step 0: Get started. (*the initial setup to do EA*)

Step 1: Establishing what business are we in. (*The big-picture*)

Step 2: Clean up. (*horizontal optimisation*¹)

Step 3: From strategy and execution. (*top-down*²)

Step 4: The plan versus the real world. (*bottom-up*³)

Step 5: Resolving pain-points⁴. (*spiral-outwards*⁵)

'Heavyweight' frameworks tend to come into their own in **Step 2** and **Step 3**. But before that happens, we need to have done the **Step 1** work of establishing the enterprise-context, and before that, we need to have established, in **Step 0**, the reason and desire for doing EA at all.

Better frameworks emphasise that the very first step after someone in the organisation decides to do something about EA, is a first-stage training, education, above all communication.

Getting to know the background is also crucial, though most of it will happen after that first '**Step-0 stage**'. These must include concerns such as identifying vision, values and mission. Also being clear about the crucial difference between '*the organisation*' and '*the enterprise*', because they're not the same.

Another really important point here: don't fall into the trap of describing *enterprise IT-architecture (EITA)* as '*enterprise-architecture*' (EA).

1. Horizontal optimisation

In an airport, making sure each part of the airport works as efficiently as possible, for example reducing duplication in security.

2. Top-down

Decisions made by the heads of an organisation which filter down to all staff.

3. Bottom-up

Staff communicating with higher ranking staff if the plans are actually working in the real world.

4. Pain-points

An airport example might be, where people are trying to get through border control, but the security system goes down.

5. Spiral outwards

In an airport, rather than picking a top-down view, we could pick anywhere, such as baggage claim and looking up, down, sideways to see how any point in the airport connects to everything else.

Everything up to this point has been, and must be, about *real whole-enterprise architecture*, because we must establish the overall scope before we can focus in on any specific part such as *EITA*, security, business-architecture or anything else.

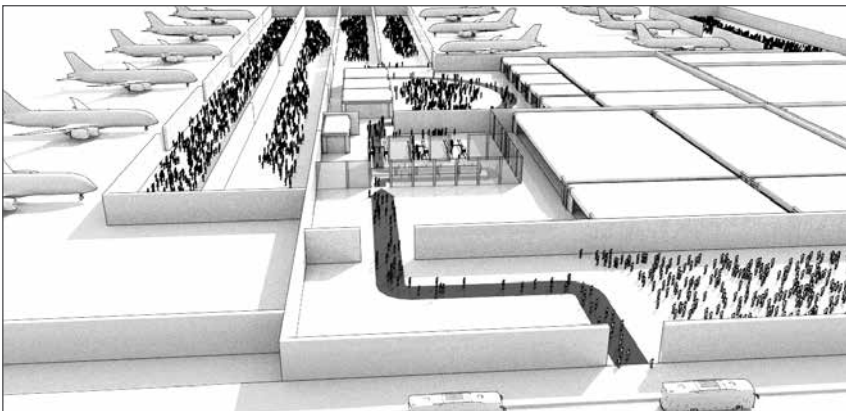
If we constrain the scope too early, we're then left with no adequate means to connect to the other area architectures, which again would guarantee architectural failure, especially over the longer term.

This, by the way, is another reason why we don't try to use 'heavyweight' frameworks for EA until such time as we do want to focus specifically on the IT related areas.

The other theme, around principles and standards, is something that we should not worry about too much until we've already gone some way down the track. At that point the only thing we need to say about principles and standards is that we are indeed going to need principles and standards, and how to apply those principles and standards to real-world practice. That's it.

Any competent architect will know that yes, we will definitely need security and governance principles. But in the early stages all you need to do is add them to a list of examples. It doesn't need anything more than that, for that stage. Later on, yes, you'll need a lot more detail. But we don't try to do that too early: all it would do is confuse people, drowning them in too-much-detail and putting them off, just at the point when we need to gain their engagement.

So, quick summary: find a way to live with the frustration, because it's going to be there for a quite a while, whatever you do; and do settle down to do everything step-by-step, because it is the only way that works.



Simple frameworks can help you begin to evaluate complex enterprises, such as how an airport could function more effectively.



2: The unique contribution of EA

What do enterprise-architects actually do?

It's not that enterprise-architects should attempt to do anything to do with business: that's micro-management, not architecture. Instead, what architects do is connect: we join the dots, link between the boxes, build bridges between the silos, get people talking with each other, to help create a clear sense of the whole as whole. To be honest, architecture doesn't do much of anything that's visible on the surface: and most its deliverables don't make much sense in those terms, either. Most of the *'doing'* within an enterprise is the role and purview of area-specialists, whereas architects are cross-functional generalists whose real role is to connect between things. *Architecture connects*: that's its real purpose.

An architecture needs to be able to connect between anything and everything that's in scope for the context of that architecture. It doesn't attempt to do everything that's in scope: but it does need to understand everything that's in scope, in *just enough detail*, and with awareness of what and how *it depends* on what, in order to unify and connect between everything and everything else that's in scope for the context of that architecture.

If something is in scope for an architecture, it's *'in scope'* because something else depends on it being there: so if the architecture can't connect between everything that's in scope, the architecture as a whole could be placed at risk. To be viable, an architecture must be able to connect everything in scope with everything else in scope. It's the connections that are the real focus of interest for the enterprise-architecture, not necessarily the *'somethings'* themselves.

In reality, everything depends in some way, either directly or indirectly, upon everything else: so the only way that works is to recognise that everywhere and nowhere is *'the centre of the architecture'*, all at the same time. As soon as we make out that some one area is *'the centre'* of the architecture by definition we'll have broken the unity and symmetry of the architecture, which means we'd have also set it up for failure in some *'unexpected'* way.

The unique contribution of architecture is that it connects, helps make whole, helps link strategy to execution, intent to action, action to value, and so on. Enterprise-architecture is just another expression of the same idea: architecture at an enterprise scope, architecture whose scope is *'the enterprise'*. Yet specialists will only work within their own specific areas, often without much if any sense of connection with anything else: so the unique contribution of EA is that it can connect everything and anything across all of the enterprise, to create that whole as unified-whole.



3: How much should an EA know & do?

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

Architecture itself must always face toward the *'big picture'* view; there's also always a large component of real, practical, concrete design, because architecture only becomes useful when it does touch the real world.

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. To make that happen, the architect needs to obtain all of those things from the clients, and so 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 world-view 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 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.

How much industry/enterprise knowledge does the EA need?

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 *'area-specific' specialist knowledge*
- architecture depends on *'link-between-areas' generalist knowledge*

So we need both types of knowledge, which is why it takes a long time to become competent as an architect. But area-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 basics of an area 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 converse intelligently and usefully in their own specific terms, but also to link all of the *'areas'* 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.

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. 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 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 capability¹ of their own and no way to use the work that's been done.

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 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 will operate and exist.

Two two key points:

- the relevant enterprise is always larger than the 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, anti-clients¹, etc*). The in-depth knowledge of our own area 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 strangeness 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 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 area.

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 present. I 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.

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 an airport, 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 passengers; other airports and other direct competitors; '*up-side*' and '*down-side*'. In other words, all the usual interweaving of the transaction-economy.

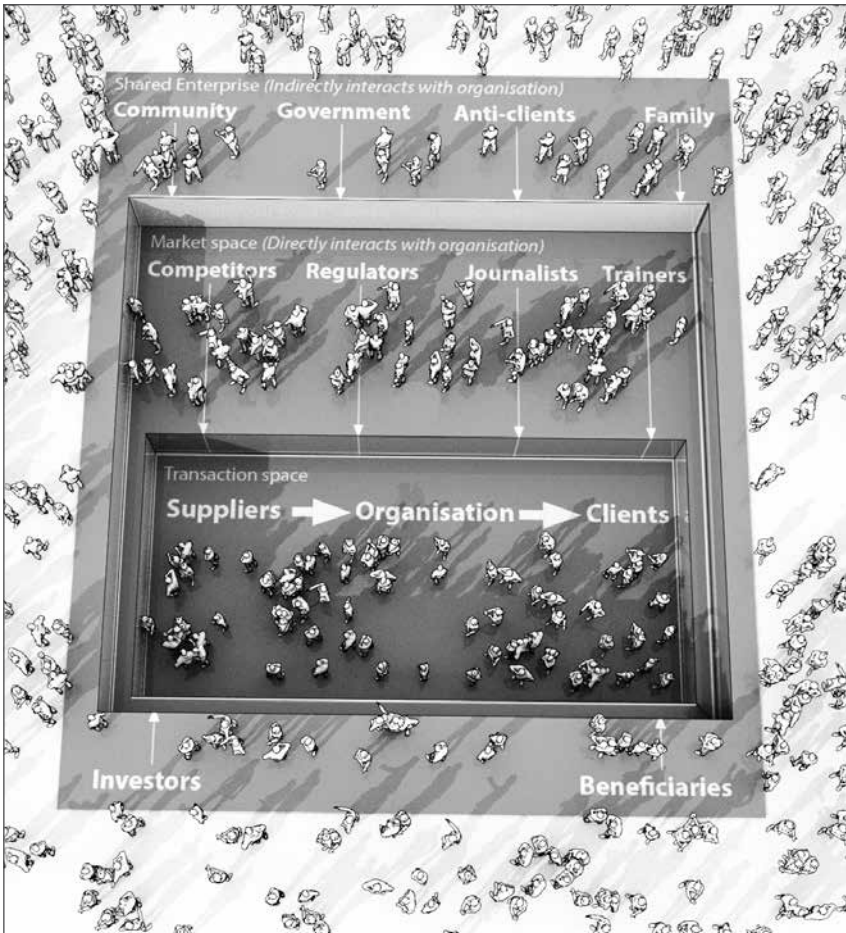
But we also need to understand what's happening beyond our immediate market. Especially where it interweaves with the attention-economy and trust/reputation-economy: hence the importance of non-clients, anti-clients. And other intersecting service-providers such as border control, fuel, medical services, and the community in general. What are some of the entirely different forms of travel that could sideline airports entirely?

1. Anti-clients

For an airport anti-clients can be all those who are opposed to the airport in some way. Local residents or disgruntled passengers. How they fit into the overall enterprise is shown overleaf, with the Whole EA tool.

Anti-clients are discussed further in chapter 28.

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, if only we'd had the sense to look out to sea...



A simplified version of the Whole EA tool designed by Tom Graves and Michael Smith allows you to explore the enterprise in which an organisation sits.



Taken from the chapter: **How much should an enterprise-architect know and do?**

Whole Enterprise Architecture

Exploring the context of an enterprise

Enterprise Architecture and *IT Architecture* are often seen as one and the same.

But do you fully understand the overall enterprise of which your organisation is only a part? *Whole Enterprise Architecture* brings together Tom Graves' thirty years of experience working in *EA* in multiple industries around the world, to help you spot and tackle the common pitfalls which plague organisations facing change.

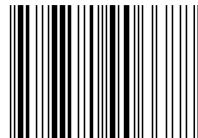
Presented in a series of approachable articles with diagrams, this book will help all organisations facing change to explore and resolve it effectively.

SAMPLE COPY



www.tetradian.com

ISBN 9789781906688



9 789781 906688