

# Which capabilities do I need to build my EA practice?

By Roland Woldt

Defining an EA capability model seems to be difficult, because the term “Enterprise Architecture” is such a loosely defined term, which has changed over the last several decades. This leads to confusion if EA is “just the tech stuff that I don’t understand” or if businesspeople rather look for BPM (Business Process Management) instead.

This document is the “partner White Paper” to the more how-to-focused “How to Get Your Architecture Off to a Good Start” White Paper, and it aims to establish the basis for services that the practice will provide, or how to grow the maturity of the practice and the practitioners.

## Three areas of a successful architecture implementation

One takeaway of dozens of EA implementations is that three major areas determine a successful implementation – the content that architects create, the setup of a proper governance, and adoption capabilities. Please note, that the technical implementation (“getting the tool to users”) does not show up as a key capability, which is something that some people find interesting and counterintuitive.

When you look at the “three-legged stool” that you need to balance for your organization as the implementer, we see three areas: Content, Governance, and Adoption.

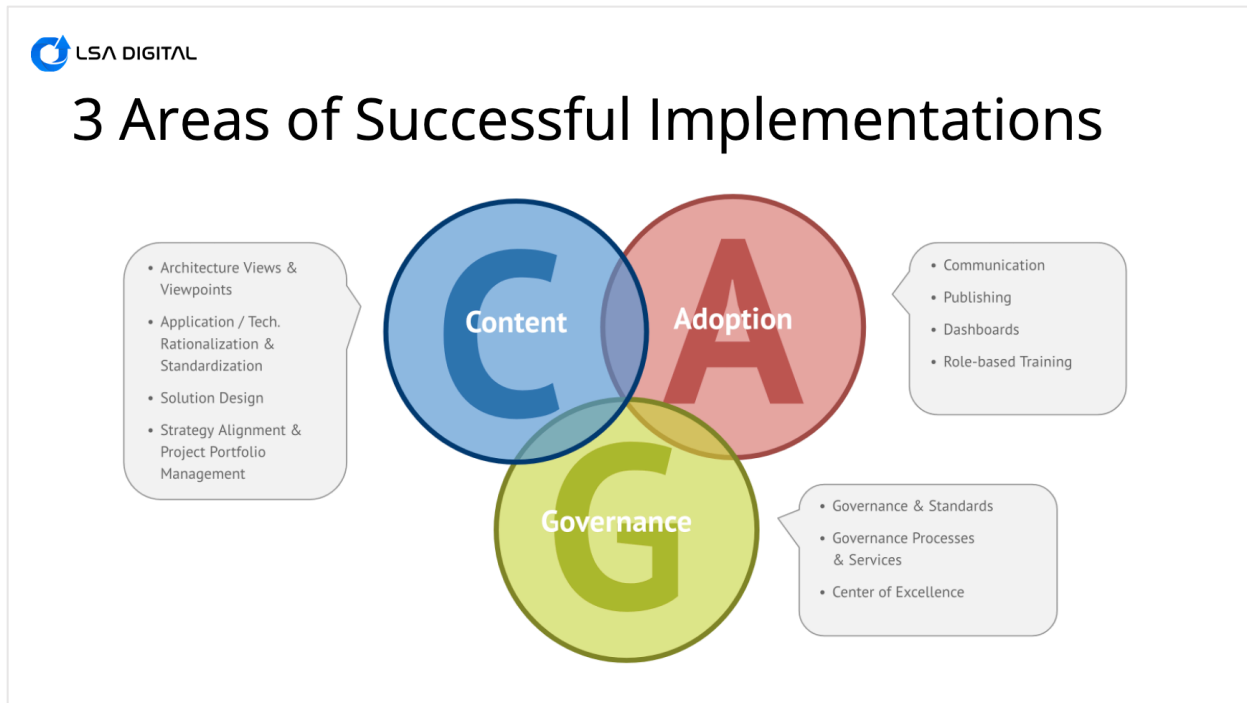


Figure 1: Three areas of successful architecture implementations, supported by an EA capability model

- Content:** this is most likely the area that brought you to identifying architecture as a key success factor for your program and got you started. It is the problem that you want to solve, and typically you see topics like the ones mentioned in the graphic above. Additional topics might be Process Mining, Robotics Process Automation, or Simulation.

In this group you will find the most advocates of the ‘give them the tool and go out of the way’ or ‘the tool is just Visio on steroids’ approaches. Don’t adopt this mentality – it will lead to a low adoption rate and ultimately makes your program fail.
- Governance:** this area defines the ‘rules of engagement’ for the program and are already mentioned in the related article. In a nutshell – you must define what your participants shall do (Process Governance), how they are organized (Organizational Governance), and which standards shall be followed (Technical Governance). This area describes the ‘inside-out’ perspective – how do we get things done?

In the past many EA groups have seen this as their main focus – maybe because they could not discuss the content topics? – which led to the impression that they built their ivory towers (or

were the 'Committee of No' vs. the desired 'Committee of Know').

- **Adoption:** As in the most failing enterprise-wide implementations this is the area that gets the least attention or will be treated as an afterthought by otherwise more technology-oriented architects. However, it will be the key driver for a successful implementation that is driven by your strategy. It takes the stakeholders, user personas, and -overall- the 'outside-in' perspective of your program into consideration. Our recommendation is to involve your org change management and user experience specialists early in the program, and let them drive the outside view, while others take care of the governance and technical aspects.

Depending on your current point of view or situation, you might focus on one area over another, but keep in mind that you want to balance these. During your implementation you might have to shift the focus according to where you are on your journey – in the beginning there will be more focus on governance and adoption, while content will become more in focus once all participants in your program have understood what is expected from them.

## What are capabilities?

First, let's talk about what capabilities are not: they are not skills and knowledge that an individual has. They are also not features and functionalities that a software has.

Therefore, it is good to make a distinction between *business capabilities* and *technical capabilities*. The latter are things that an application can do and fits more the second scenario above. In earlier versions of our favorite EA tool this object was called "IT Function", and I think this is a better term than (technical) capability. It makes it even clearer in the context of an application rationalization, where you want to find redundant functionalities to streamline your application portfolio and avoid multiple apps that do the same, or you might want to find missing functionality that you can provide with another app/extension or build by yourself.

The more interesting term in our opinion is the business capability. This term means the definition of the "what" an organization shall be able to do. It is comprised of things shown in the middle of the graphic below: organizations, skills, knowledge, locations, applications, processes, data, and more. Together, these items provide a good high-level description of what shall be implemented, updated or retired.

Business capabilities are a core concept of every architecture – they are derived from the strategy (which is not only is comprised of the items below, but also includes the services and products, the business model, and other things), and should align to it. The business capabilities are then driving the programs on the roadmap and can be used to be a litmus test for each project request by asking “which capabilities do you create/update/retire?”. If the answer is “none of those” then the justification for that project funding request is on a very weak foundation – in an ideal state, everything should be aligned with the capabilities and the results of the projects should be measured and show the accomplishments of the strategic objectives.

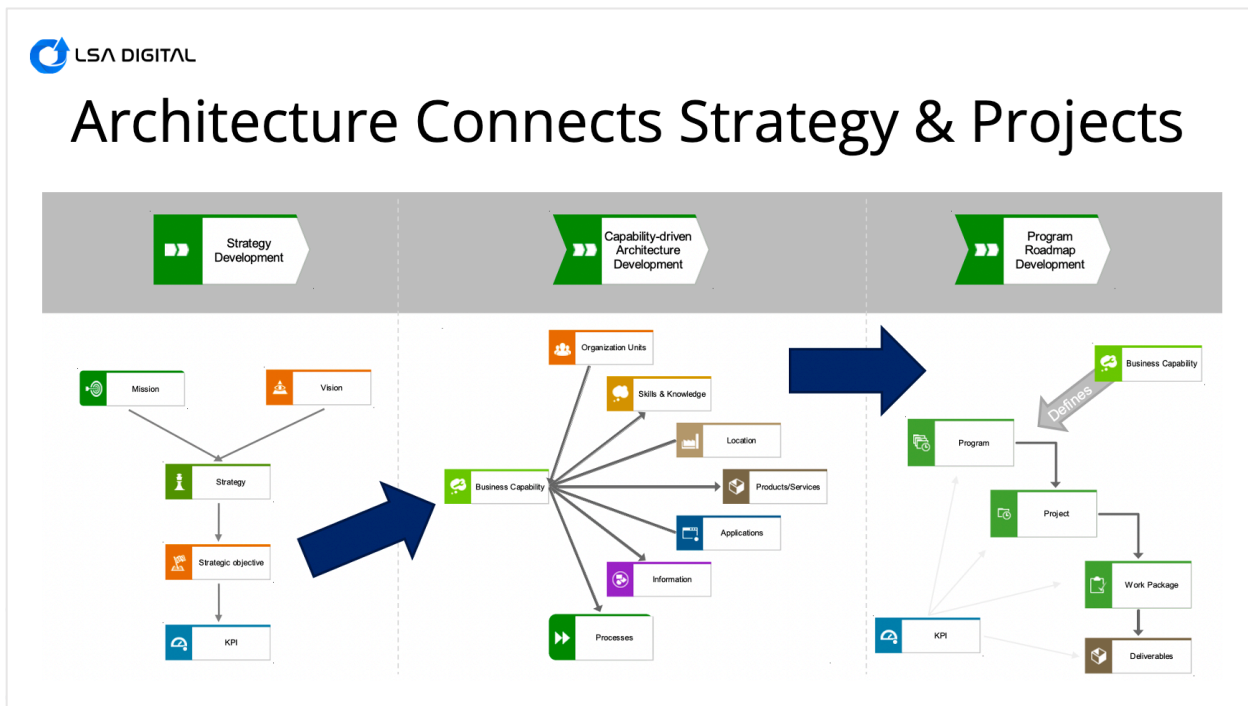


Figure 2: EA capabilities bridge strategy and projects

Implementing projects requires a bit more detail, though. When you look at your organization, you will find different states of maturity, or needs and requirements which might be a result of your organizational structure (some org units might have more people than others, and therefore might have the need for more automation and workflow, while the smaller regions might be able to get away with a more manual approach).

To show this and have objects available in your planning and alignment during the architecture development, the concept of *capability configurations* comes into play. Capability configurations are subsets of the main business capabilities and might be defined either as different fidelity states of a capability (think “gold/silver/bronze” in the example above), or various states of the capability being laid out on a timeline. Either way, the capability configuration is what needs to be defined to describe the scope of a project (and, of course, shows the specific processes, affected apps, data, etc. in more detail).

An architect will have to integrate this back into the overall capability model, answering the question of “how far are we in implementing the overall capability model?”. This will then give the feedback to the strategists about what the organization might be able to do soon, and what the impact on products/ services or program delivery might be.

## EA capability model in more detail

So, what does that mean for your architecture practice? In our experience they are grouped into five big buckets of business capabilities shown in the graphic below:



# Architecture Capabilities

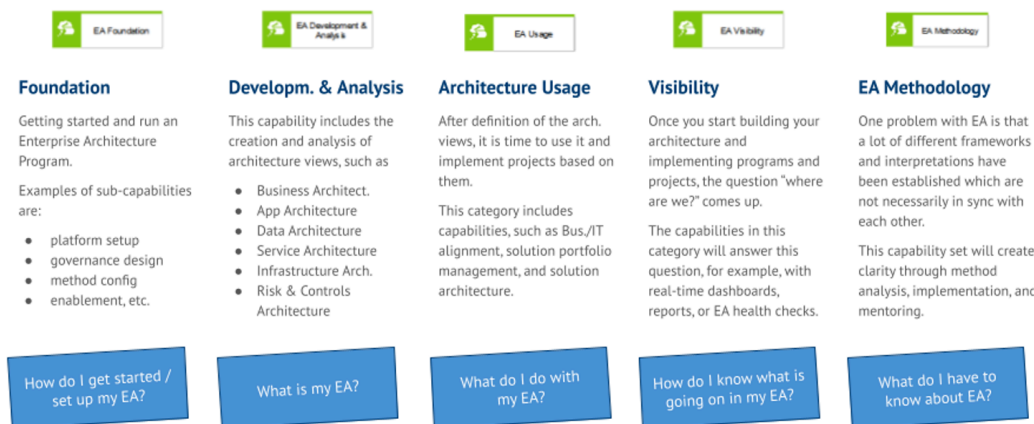


Figure 3: EA Capability Model on a high level

- **EA Foundation:** This bucket includes all capabilities that are needed to answer the question how to set up and maintain an architecture program. This includes the technical aspects of your chosen platform components, and governance, but also the people side of things – the governance and adoption circles in the Venn diagram above, as well as the high-level content structure (which framework to follow, the structure of the meta model, and so on).
- **Architecture Development and Analysis:** this is the first area of content, and it includes the capabilities that around the various views of your architecture. If you are following the TOGAF framework for example, this covers the Architecture Development Method phases A-D. Besides modeling the content in Enterprise Baseline to get an understanding of the as-is (“what runs?”), this capability bucket also includes the various analysis techniques, such as application rationalization, technology standardization, or process mining. It basically aims to answer the question “what is my EA?”.
- **Architecture Usage:** Once you understand what your as-is architecture looks like -most likely on a very high level when you get started- the logical question is “what I do with it now?”. This capability area includes things like creating solution architectures (in the Projects/Solutions area of the architecture tool database), managing the solution/project portfolio, or foster the alignment of business and IT. In TOGAF speech this would include Phases E to G and the requirements management (Phase H is part of the governance definitions in the overall approach and already included in the EA Foundation area).
- **EA Visibility:** one aspect of the architecture implementations that I have seen clients struggle with is getting an answer on how far they are in their architecture efforts. On the one hand, this is caused by the lack of a strategy, and therefore objectives that are not well defined, which then leads to arbitrary KPIs that will be measured, and on the other hand, the pure focus on non-capability-oriented measures, or “formal” measures. An example for such a KPI is how many trainings have been delivered as part of a go-live – this is interesting information but does not tell the story if the implemented solution brings in the business results that should be achieved. That would be better measured six months after go-live when looking at the business results.  
In addition to this, a lot of programs did not do their stakeholder needs analysis homework and determined what the information needs of these stakeholders are – this leads to generic dashboards that are mostly not relevant to an individual stakeholder and want to do too much

(but are easier to build).

- **EA Methodology:** when one gets started with the topic of EA there is a big learning effort in basically understanding the strange language that architects use, which also contributes to the “ivory tower” perception of them. However, a lot of EA concepts and approaches are very valuable and therefore the question in this capability bucket is “what do I have to know about EA?”. This covers things like which frameworks or notations to use, but also assessing if a new concept comes up (for example: “do I still need an architect when I do Agile?”), and if the answer to these questions is that it is in fact relevant, then you need to plan how to implement this (and the impact on existing work), and actually do the changes.

One suggestion is to capture all the architecture capabilities needed in a multi-level EA capability model that can act as the catalog for all the things that you need to define an implement to get your architecture program up and running – nicely laid out on a timeline later. The capability model that we brought to clients as reference content in the past included the five major capability buckets above, which were then decomposed into 24 level 2 capabilities that included ~110 level 3 capabilities. Our suggestion when creating an EA capability model is to limit yourself to 3-4 levels at maximum, so that you don’t lose the high-level picture and get lost in the details (which are defined in your cap configs anyways) – this is the time for broad strokes.

We then used the capability model to determine the services that the EA organization should provide when developing an EA strategy. It can also be used for assessing the EA maturity, which will also be a topic for a future article.

Please note that none of these areas include or prescribe how you implement the EA capabilities. For that you will have to have a look at your tools, standards, maturity, and other factors, and then derive the capability configurations needed to grow the architecture maturity in your organization. But that topic will be discussed in another White Paper.

Do you like what you see, have more questions, or simply want to chat with us? Please reach out at [info@lsadigital.com](mailto:info@lsadigital.com)!