Enterprise Architecture is often compared with building architecture. And in some ways the two are similar... but in other ways they are not! This comparison is quite common – in fact it came up again in a recent discussion with a newly formed EA team, so I thought it would be useful blog topic.

When students are introduced to TOGAF, the analogy with building architecture is used to explain what enterprise architecture is all about. The parallels help to position EA within an organization, but there are also many characteristics of EA that can’t be explained!

Let’s start with a little historical background. The word architecture was first used around the middle of the 16th century. It is derived from the Latin word architectura, which in turn came from a Greek ἀρχιτέκτων or arkhitekton. Arkhitekton has two parts to it – ἀρχι-, meaning "chief", and τέκτων meaning a builder, carpenter, or mason. In this original form it clearly meant the person responsible for overseeing the design and construction of a building. So although we have a similar meaning when we talk about enterprise architecture, we will more usually find several “chiefs” and they are probably not going to be builders, carpenters or masons!

More recently architecture has been used in a broader sense to cover the complex or carefully designed structure of something. More general uses might include the architecture of a computer system, or the architecture of the human brain.... which brings us to the difference between building and enterprise architecture.

Now I’m sure I’m not alone in thinking that as our world gets more and more complex, so architectural thinking becomes more useful in making sense of it and possibly in organizing and managing it; which means that architectural thinking can apply to pretty much anything with a designed structure. And if you look back over the last 40 years of enterprise architecture you’ll find that it has evolved from a discipline that was largely aimed at helping develop IT systems to something much broader. Today with have business, information, social, and environmental architects; and each of these different types can be under the banner of enterprise.

We need different words to appreciate the two basic types of architecture – so instead of talking about building versus enterprise architectures, I think it is better to refer to “hard” versus “soft” architectures.

• Hard refers to architectures that deal with more tangible, concrete or material outputs – the most obvious being buildings, but this might also include planes, cars or the physical aspects of utility and communications networks.

• Soft refers to architectures that cover less tangible outputs; this includes outputs that are conceptual or logical, ones that are difficult to see or touch, or ones that are non-figurative and abstract. Soft architectures therefore include software systems, organization structures, business rules, capabilities, and processes.
Let's start by looking at the ways in which hard and soft architectures are similar:

- They both provide a complete, holistic, big picture view of things.
- They can both provide some consistency and coherence across the components or building blocks, allowing them to fit or work together more effectively.
- They both provide multiple viewpoints and views, to explain the big picture in ways that make sense to everyone involved.
- They both deal with short-, medium- and long-term timescales.

There are other similarities, but these should give you an idea of the commonalities between these approaches.

**Hard Architectures**
- Building blocks and components are tangible – bricks, mortar, window & door frames, etc.
- Relationships between components are physical and manufactured
- The focus is on shapes and spaces.
- Specifications focus on physical attributes (strength, durability, etc.), material costs, etc.
- Requires experience and knowledge in building materials, construction techniques, architectural styles and so on.

**Soft Architectures**
- Building blocks and components are intangible – processes, services, software, rules, etc.
- Relationships between components are digital, logical or virtual, and often generated.
- The focus is on information.
- Specifications focus on attributes such as reliability, availability, security, agility, flexibility, etc.
- Requires experience and knowledge in techniques such as taxonomies, classifications, patterns, layering, separation of concerns, etc.

Register today for a free Good e-Learning account to view this resource