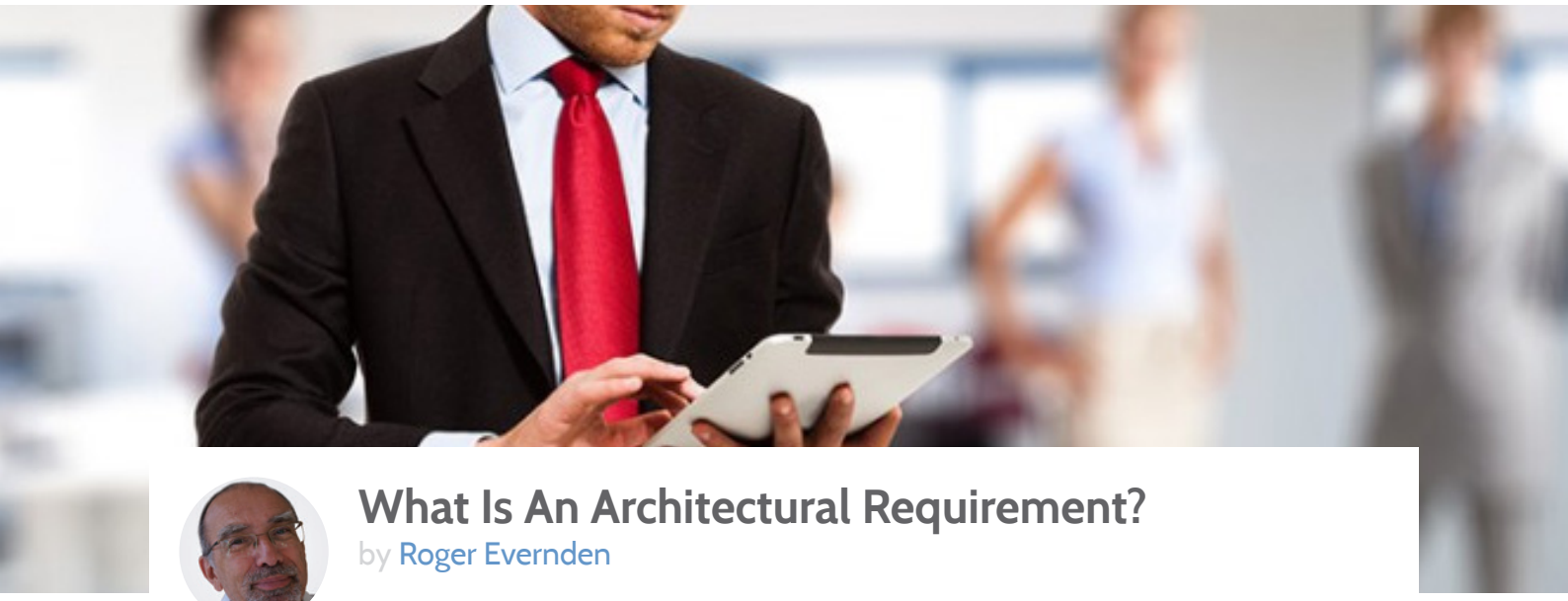

What Is An Architectural Requirement?

by Roger Evernden

It sounds a simple question – What is an Architectural Requirement? Surely it's simply a need to change the architecture in some way. And surely there's a clear reason for making the changes. But it's not that simple. Sometimes an architectural requirement is confused with an IT requirement. Sometimes it's seen as the same thing as a business necessity.

Let me illustrate with an example. A client told me recently they had been asked to recommend a new Internet platform; they wanted to know if this was an Architectural Requirement? More importantly, they wanted to know how to make it a more effective Architectural Requirement.

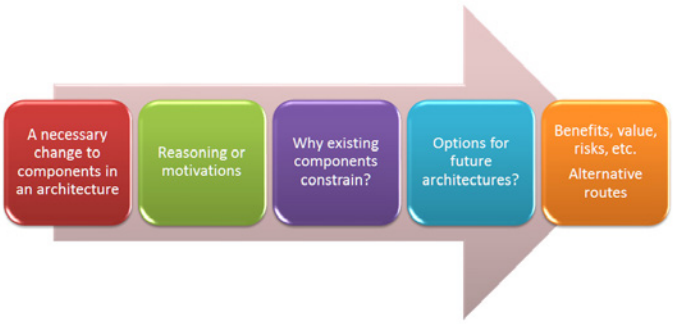

Figure 1: Architectural Requirement Diagram

Key Characteristics of an Architecture Requirement

TOGAF has a whole phase [illegible] corresponding chapter (17) [illegible] devoted to Architecture Requirements Management. But [illegible] an architecture requirement is [illegible] basic process for managing their documentation.

Other [illegible] deliverables and artifacts that document an architecture requirement. For example, [illegible] include the Architecture Vision, Architecture Definition Document, Architecture Requirements Specification, and things like Business Scenarios, Principles, or Architecture Building Blocks.

A glib answer to our question might be to say that an Architectural Requirement is defined by the sum total of these deliverables and artifacts! This is certainly how TOGAF would have us document an Architectural Requirement, but it doesn't really explain the difference between an architectural and a non-architectural requirement.

Let's make it easier by listing the key characteristics of an architecture requirement:

- It should describe a necessary change to components in an architecture. This might be [illegible] components, removing [illegible] replacing or improving components, or changing the way in which they work together. What is going to change?
- It should explain the reasons and motivations behind the change. Why does it need to change?
- It should explain why the existing components are inadequate, limiting or constraining. What problems, issues or concerns are caused by the current architecture?
- It should describe the available options [illegible] that address all concerns. How do alternate target architectures eliminate the problems of the current architecture?

- It should explain the benefits, value, risks, costs, opportunities, constraints, and future options associated with each alternative. How do we decide between one alternative and another?
- It should outline any alternative routes to close the gaps and get from the current to the target architecture. How do we make the transition or transformation from what we've got now to what we need in the future?

To go back to our example: the client asked to recommend a new Internet platform. Their architecture requirement could be summarized as:

- We want to update the architecture on which the Internet platform is based. This requires replacement of the technology platform, allowing improvements to application functionality and more real-time data processing, which would enable better integration of c[illegible] and generate more prof[illegible] This provides a simple e[illegible] outcomes in a way that stakeholders can understand, [illegible] components in the architecture.
- [illegible] architectures are outdated and [illegible] in comparison with our competit[illegible] resulting in a massive loss of market sh[illegible] dissatisfied customers. The rationale here is clear to all stakeholders – the future of the company is at stake.
- The current techno[illegible] developed in an ad[illegible] components do no[illegible] foundation for the [illegible] customers expect. Furthermore, many of the components are no longer adequately supported by vendors and there is a lot of unnecessary duplication, which makes maintenance costly and frequent disruptions to availability. The problem is explained in terms of the constraints imposed by architecture.
- There are two options available to us. We can outsource the underlying technical platforms, using standard technologies provided on the cloud as a service. Or we can build an in-house capability using vendor-provided technologies. The alternatives are outlined by describing the end-result, but it is also easy to demonstrate the architectural differences between the two options.
- Using outsourced services and well-defined standards would allow us to incorporate services from multiple vendors – giving greater business flexibility and functionality. However, this approach would require greater coordination at the business process level, which might be more complicated than management would tolerate. An in-house capability might be a simpler solution, but would probably restrict us to a single vendor, which in turn might limit business process and product flexibility. The factors that will influence the choice between the alternatives are clearly summarized.
- Initially we would need to replace the underlying technology platform – with either an outsourced service-based environment or an in-house vendor solution. This would [illegible] architecture, reducing costs, improving availability and reducing maintenance overheads. This [illegible] intelligence and usefulness of information by better [illegible] allowing better marketing and generating higher sales. Finally we would be able to make parallel improvements to application functionality and business processes – making it easier to customers to buy products and services [illegible] with our sales teams [illegible] clearly shows key [illegible] deliver.

[illegible] greatly simplified. [illegible] detail, and in particular the arguments would need to be firmly based on complete descriptions of the current and target architectures, and the possible road-maps to make the changes. But hopefully this simplified overview has emphasized the things that form an Architectural Requirement.

One Final Point

Remember that this is an "architectural" requirement; if you don't relate everything to the relevant aspects of the enterprise architecture, then it is simply a "requirement"!