

White Paper

What is Architecture?

Simon Watts

Synopsis

Knowing what to expect from architecture and knowing what to demand from it are both things that business wrestles with. This article breaks architecture down into its key deliverables and presents a framework for understanding what can be achieved, specifying what is required and a way forward for delivering against expectations.

This document contains information proprietary to Miley Watts & Associates Ltd, and may not be reproduced, disclosed or used in whole or part without express written permission of Miley Watts & Associates Ltd.

If you would like to discuss any of the ideas described in this document in more detail then please contact us at enquiries@mileywatts.com and we will be happy to have a conversation.

Many times we hear the question 'What is Architecture?' often accompanied by the response that it is just design by a different name.

Yes, architecture is design but it has specific responsibility and differs from other forms of IT design because of this.

Architecture is a combined discipline comprising analysis, design, philosophical thinking, planning, programme management and governance to set out the IT strategy. It is not a purely technical pursuit but must hold its own in both the business and technical communities. As Barry Skingle, then Chief Architect of Reuters, once said 'Architecture is the means by which business takes control of technology'.

Understanding what architecture is and the benefits it can bring are essential to both a hiring manager and a business looking for improvements in its IT strategy. Other things that drive a proper understanding of architecture include effective market entry or addressing an acquisition with speed and agility.

Rather than 'What is Architecture?' the real question is 'What does/can architecture deliver?'. This should be the main question addressed by any business when thinking about its IT strategy.

Architecture delivers a number of things for an organisation, which we can summarise as:

- Realisable IT Strategy
- Principles for IT Governance
- A Framework for IT Delivery

As a whole, architecture provides a realisation of the business goals within the context of IT.

In this article, we'll look at what architecture delivers, what it is not and the characteristics that define it.

1. Realisable (Implementable) IT Strategy

Architecture is meaningless unless it is realisable therefore it must draw from business strategy. It must be grounded in realistic and affordable approaches. It must acknowledge the organisation and will be influenced by several factors including:

- Market
- Current Assets
- Technology Advances

Architecture predicts the implemented state, and sets the measure of success.

It defines goals/objectives and purpose that will fulfil the business strategy. This gives the architecture its scope.

2. Principles for Governance

Architecture must deliver something that is governable and consequently must explain how to govern itself. To achieve this, architecture should exhibit a clear derivation from business objectives and purpose without which it can offer no justification. It should acknowledge current methods (internal or external e.g. RUP¹) and practices though it may also introduce new ones.

Architecture must set principles, which govern:

- The overall framework
- Detailed design
- Design choices
- Implementation
- Use of patterns and standards
- Divergence and convergence plans
- Management and oversight

Finally, it must set policies and guidelines, which act as the backdrop for other decisions and actions.

3. Providing a Framework for IT Delivery

Within the context of purpose and objectives (the strategy) and constrained by the principles, architecture is a design for the future.

Based on solid analysis of business intent and current assets it deconstructs the business strategy into moving parts (called services nowadays).

The architectural framework defines scope – what is in and what is out, what is permissible and what is not.

It must identify abstract/organic/natural systems (or domains and sub-domains) and decompose them to understand their nature, function and operation.

Vitally it must define its boundaries and describe the behaviour and interaction that occurs at these points. Understanding the boundaries delivers sustainable, evolving and extensible environments.

The framework provides context for design choices and constrains them according to corporate policy or technical suitability.

Architecture selects or constructs patterns that are relevant to the domain and stipulates their use in preference to other approaches, which provides consistency and ultimately a skill set and body of knowledge usefully employed by many in the organisation.

4. Scope

When defining architecture an early activity is to clarify its scope. To do this the architect must first understand what comes before – namely the business context dealt with earlier in this article. Then it is a matter of figuring out the significant factors that

¹ Rational Unified Process

will shape the architecture. This boils down to stating what the architecture is and isn't responsible for, what dimensions it must deal with and where its boundaries lie.

Although the definition of architectural scope is a recursive activity it is important to begin the serious thinking here because it is instrumental in shaping everything that comes after. Then as the architecture develops one can return to this part and refine to reflect the more extensive thinking that has taken place.

Defining the scope is really a matter of making assertions in four ways:

1. A list of factors, characteristics, behaviour and function that is the responsibility of this architecture (what is in scope).
2. A list of what is **not** in scope. This can contain anything that seems relevant or worth mentioning. It should, at a minimum, state things that might be assumed to be this architecture's responsibility but aren't and provide some rationale for such.
3. The list of dimensions that are of concern.
4. A list of boundaries, both internal and external to the organisation, that will be touched by the architecture. This list can draw from more than one layer of abstraction – for example it can include organisational boundaries such as departments, customer interaction points and deployed systems as well as patterns to be employed.

The first two can be derived or inferred from the earlier work on the business context.

Dimensions that can be considered include:

- Time
- Organisation
- Business purpose
- Market forces
- Business strategy
- Geography
- Geopolitical
- Infrastructure

Each dimension may manifest in more than one way – many of which may be pertinent to this architecture.

As an example let's take a quick look at time. An initial list of the ways time rears its head for an architecture might include the following:

- The longevity of the architecture

- How long the architecture will take to realise
- Any temporal aspects of the services
- The temporal nature of the data

Thinking a little further we can also add to the list such considerations as:

- Operational delay in the infrastructure
- Time slices/units relevant to data flows
- Roadmap phases
- Temporal obligations or opportunities in the market

Our recommended way to proceed is to consider each dimension and list the ways it may manifest and impact this architecture. This can be refined later but this list will help shape later thinking.

Architecture has boundaries, as all design work does. The question that must be answered is what those boundaries are. The architectural boundaries should be enumerated and elaborated to understand their impact.

Boundaries will either be natural or artificial. Examples of natural boundaries are:

- The edges of the business
- The edges of departments
- The edges of systems
- The edges of the customer
- Time

These are natural boundaries – in other words, something that exists in the real world that cannot be controlled by the architecture but that the architecture depends on.

There are also artificial boundaries which include:

- Business strategy and its interpretation
- The architect's own thinking, knowledge and experience (or lack thereof!). What can and can't be conceived represents a boundary
- Assumptions (stated or unstated)
- Corporate memory

Boundaries can be identified by naming the point at which control ceases and interaction with something outside takes place.

With the scope set, the architectural principles should be revisited.

Principles provide a supporting framework for design decisions both within this level of architecture and in the detailed design and implementation that comes after. Throughout, the principles serve as a manifesto for what will be adhered to (providing governance); a method of measuring conformance; right thinking; and acceptability (providing grounded reviews of components within the architecture).

Depending on the context and the strategy of the organisation, other product lines of questioning may include:

Will the architectural boundaries always remain the same?

Is there actually a repeating pattern as you move up and down the layers of abstraction?

How does this architecture differ from detailed or high level design? Therefore what is out of scope for the architecture but is in scope for the next level down? What defines decisions that must be made at lower levels of abstraction?

How should each level of abstraction be defined? Implementation detail? Constraints? Levels of interest? Static versus moving parts? Which technology decisions must be made? Now?

What can/should be done about natural and artificial boundaries? How is the architecture influenced by market forces or controlled by them? Can it initiate change in a natural boundary or should the architecture assume that its boundaries are immutable? What should the architecture say about what goes on outside its boundaries?

Should the architecture try to make everything conform or allow diversity? How?

Should the architecture describe life cycles?

What does the roadmap elaboration add?

After all that let's see if we can put together a better definition of what architecture is.

Architecture is a descriptive model beginning with statements of purpose, objectives and the business function it will serve. It defines scope and subsequently explains what exists within the architecture (the moving parts) and how both the whole and key parts behave under a variety of conditions. It states the principles by which it will be governed, patterns which are relevant or should be employed or are natural phenomena that must be acknowledged, what its boundaries are, what its scope is, what its scope is not, what its purpose is, what its measure of success is and how this can be recognised.

No architecture is meaningful until this has been done.