Is It Time for an Architectural Renovation?

Any software built several years ago would have been developed based upon the needs of consumers and architectural standards at that time. The more rigid systems of “back then” are now transforming into agile, online apps, and today we are in the age of sliding, swiping, air gestures and being mobile. As with any software enhancements over time, expansions and changes are sometimes made without knowledge of the underlying software architecture, so the architecture can become unstable. Organizations looking to modernize their application portfolios frequently require an architectural renovation to reset the foundation for their next-generation solutions.

Software architecture requires periodic renovation

Software architecture is similar to building architecture; actually, the former derives a number of concepts from the latter to explain planning, design, construction, and certification processes. An organization's architect is similar to a City Planner, who is responsible for ensuring resources are effectively applied to the organization's needs. Applications are built based on defined building blocks, applying patterns so they conform to code and blueprints, depending on the nature of the business and a budget that maximizes return on investment.

A city goes through a number of evolutions as the needs of citizens change. Growth and prosperity bring opportunities that call for expansion, demolition, new construction, additional resources, and revision of building codes to accommodate the changes. Similar changes happen within an organization's software architecture. The organization may have a compilation of software applications built from the ground up, acquired software that has gone through multiple customization cycles, commercial off-the-shelf software configured to meet specific business needs, software applications built on platforms, and frameworks across heterogeneous technology stacks. As in building architecture, software enhancements, modifications, and extensions sometimes go in without architectural (city) approval. This compromises building integrity. Similarly, software integrity is impacted every time a structural change—a non-compliant change or a deviation from set standards—is made.

The need for periodic architecture review does not go away, however, and unchecked design will negatively impact the organization's business sooner or later.

Over a period of time, these non-sanctioned changes will increase risk to the organization's business. They introduce application instability, increase application maintenance costs, reduce the capacity to release new features, and lead to a number of unsustainable technology stacks. At some point, businesses either need to renovate their software architecture or tear it down and replace it completely.

Many organizations have a form of checks and balances to help mitigate instability. Some have an architectural review function within their Change Control Board (CCB) and in others, a formal Architectural Review Board (ARB) assesses proposed changes prior to their implementation. Unfortunately, with the constraints on businesses to reduce spend and increase productivity, resources are often diverted to other strategic initiatives, and the ARB is compromised. The need for periodic architecture review does not go away, however, and unchecked design will negatively impact the organization's business sooner or later.

Creating a new building framework

The importance of establishing a generalized architecture

Companies will frequently conduct an architecture assessment following an application portfolio rationalization. Once the organization has a good picture of the applications it should maintain, retire, or add to meet future state requirements, technology leaders can more readily determine how well the current, underlying architecture supports those objectives.

Often, organizations use architectures that are not generalized or documented, which makes it difficult for domain subject matter experts, business analysts, technical teams, infrastructure teams, and most importantly, the architects to contribute in a meaningful and consistent manner. They must overcome knowledge gaps due to the lack of a common model, which can derail discussions around architectural synthesis. In contrast, having a Generalized Enterprise Reference Architecture Model (GERAM) defines a blueprint that reference architectures can use, including the lifecycle elements (e.g. as-is, to-be, needs-are, could-be architectures), the modeling frameworks used in builds, and the associated modeling constructs. This generalized model recognizes contributions of various existing architectures, as well as emerging ones, that align to the needs of the business. Having this defined will help technologists articulate a complete collection of methods and modeling tools they can employ for an enterprise engineering effort. It also enables stakeholders to focus on developing deliverables aligned to application tasks and activities, rather than a contentious debate on architecture form and factor.

Knowledge sharing—or lack thereof—can also have an impact on architectural integrity. Quite often, knowledge of specific architectures resides within a select individual's head. Having a documented, generalized architecture can help mitigate risk should this person leave the organization. Additionally, this challenge can be addressed by grouping various architectural objects on the basis of the category or domain they belong to (e.g. grouping merchandising elements as part of a larger, retail systems architecture). These objects could be resources, people, tools, etc. that are available within the enterprise or could be accessed by virtue of having partners, suppliers, vendors and 3rd party service providers.

The generalized architecture model should also capture the Structure, Behavior, Value, and Knowledge dimensions that collectively define the enterprise. Each of these dimensions detail important aspects of the organization: for instance, the Structure outlines the various architectural objects in an enterprise, where they are located and how they are connected; the Behavior dimension describes the interaction of these objects, the states they pass through and how these objects behave over time; the Value dimension captures the capability each object brings to the organization relative to the associated business risk it introduces and the cost of maintaining it. The Knowledge dimension is particularly important, as it provides an enterprise with a mature knowledge management approach to architecture that is complete, precise, certain and predictable.

Quite often, knowledge of specific architectures resides within a select individual's head. Having a documented, generalized architecture can help mitigate risk should this person leave the organization.

By categorizing important architectural components, an enterprise can be modeled in a manner that makes it easier for all parties to have a uniform understanding of the way the enterprise works today or how it should be working.

Employing a thorough modeling approach

To create a generalized model, organizations must establish for themselves a modeling technique, method and capability. There are many modeling methods and technologies, and the challenge is to ensure that modeling is consistently applied and is sustainable. It can be helpful to assess the enterprises' modeling capability to confirm the right constructs and structures are available for a sustainable architecture representation. Modeling should also be flexible, as new technologies, techniques and methods can offer significant productivity gains.

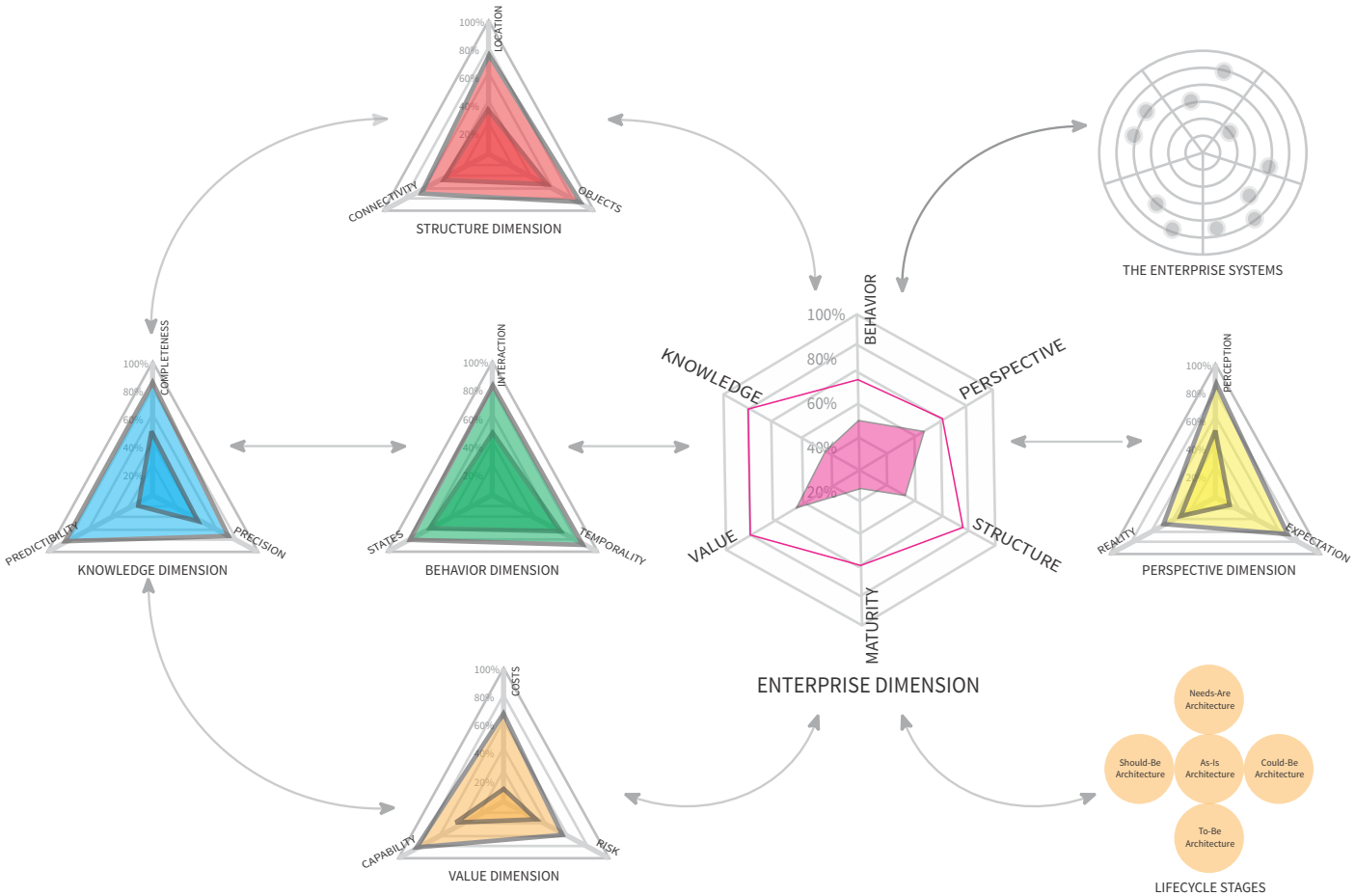
There are seven stages of modeling:

1. **Capture and Represent** – the specific measures of effectiveness relative to the As-Is, Could-be states of an architecture's objects and relationships.
2. **Store and Access** – the specifics on how architectural models and views are stored in a centralized location and accessible to all architectural modelers.
3. **Visualize and Understand** – the capabilities to visualize various architectural models from the perspective of enterprise flows, layouts, and configurations.
4. **Analyze and Design** – the ability to model simulations, analytical models, process models, etc., and adjust and design accordingly.
5. **Verify and Validate** – the specific measures related to traceable models that help the organization validate its architecture models and views.
6. **Evaluate and Select** – the specific architectural models and views that provide insights, improvements and perspectives that are valuable to the organization.
7. **Implement and Monitor** – the specific capabilities for deploying and monitoring the manner in which architectural models and views are actually used in an organization.

These seven stages help ensure that a comprehensive architecture model is established and maintained over time.

Ensuring the architectural model includes all necessary dimensions

In today's digital economy, companies particularly want to align their architectures with user-oriented dimensions and data analytics to make the architecture actionable. Architects need to be involved in the evaluation and assessment of these dimensions. By understanding the current architecture relative to finer elements (e.g. by Function, Information, Decision, Physical) at multiple levels (e.g. conceptual, structural, and realizational), companies can do a more thorough evaluation of how well their architecture supports business needs. The GRAI-GIM modeling construct enables this approach by looking at the As-is and To-be states of software architecture across a number of meaningful dimensions.



Adapted from "Aspects of Enterprise Modeling" in
Enterprise Modelling - The Readiness of the Organization by Ron Hysom
From the Handbook on Enterprise Architecture, Springer Press, 2003

GRAI-GIM Model

Another well-established model is the Zachman model. It has six dimensions designed to provide a fine-grain view of the enterprise on attributes such as Function, Data, People, Time, Network and Motivation. Each of the dimensions is then described at five different levels: contextual, conceptual, logical, physical and representational. Using a simplified scoring model, technologists can gain visibility into architectural strengths, challenges, and gaps in a comprehensive manner. Trending the data collected over time can also provide insight into emerging gaps, risks and costs of model deviations.

	FUNCTION	DATA	NETWORK	TIME	PEOPLE	MOTIVATION	
SCOPE	List of processes that the Business performs on	List of things important to the Business	List of processes that the network performs on	List of events significant to the Business	List of groups important to the Business	List of business goals and strategies	CONTEXTUAL
BUSINESS MODEL	Business Process Model	Semantic Model	Business Logistics Systems	Program Schedule	Workflow Model	Business Plan	CONCEPTUAL
SYSTEM MODEL	Application Architecture	Logical Data Model	Distributed Systems Architecture	Processing Structure	Human Interface Architecture	Business Rules Model	LOGICAL
TECHNOLOGY MODEL	System Design	Physical Data Model	Technology Architecture	Control Structure	Presentation Architecture	Rules Design	PHYSICAL
DATA REPRESENTATION	Component Model Center	Data Definition	Network Architecture	Timing Definition Details	Security Architecture	Rules Specification	DATA REPRESENTATION

Zachman Model

Regardless of which technique is used, an assessment should highlight what is working well, what is not working well, and the gaps, challenges, risks and issues that are (or will) impact the business. Isolating the findings by the different dimensions can help break down the problem in a more meaningful manner, so it can be architecturally addressed by decomposing specific functional categories.

An architecture assessment should also provide recommendations on what needs to be done from a People, Process, Tool and Governance perspective. This will help organizations implement corrective measures that ensure challenges and gaps are addressed immediately or in the near future.

Generating architectural models from code

Organizations will sometimes conduct code assessments to review architectural quality, particularly when there are no architectural models in place or when the current model is no longer relevant due to significant changes in business direction. By reverse engineering the code, technologists can derive the primitives, and then use those primitives and abstractions to get a sense of the current architecture's integrity. A number of reverse engineering tools are available in the market to support this approach and are often used with respect to legacy application modernization. Although this approach is not a substitute for an architectural model, it will provide a good starting point. Code analysis also highlights some of the technical debt that needs to be addressed and can reveal other details the company was not anticipating.

In summary, as companies participate in the Digital Economy, they need to have a current state architecture that is relevant to the strategic direction of the business and aligned to the technology capabilities of the enterprise. Having a contemporary software architecture—one that is Actionable versus Reactive—will help the organization adapt swiftly to business changes and maintain resilience to competitive threats, so the company is well-positioned to retain customer loyalty and create exciting, new revenue models moving forward.

Architecture Assessment Services from Ness

Within 6 – 8 weeks, Ness Digital Engineering can complete a comprehensive architecture assessment to help your organization build a solid foundation for its next-generation solutions. We evaluate business, technology, and integration architecture, as well as the enterprise technology stack. Our team employs industry-proven methods, such as Zachman and TOGAF, to unearth challenges, shortcomings, and opportunities for improvement around technology use, process and people. The deliverables include a diagnostic report and actionable recommendations on what can be done in 6 months, 12 months, etc. to ensure your applications meet the needs of your business.

About Ness Digital Engineering

Ness Digital Engineering designs, builds, and integrates digital platforms and enterprise software that help organizations engage customers, differentiate their brands, and drive profitable growth. Our customer experience designers, software engineers, data experts, and business consultants partner with clients to develop roadmaps that identify ongoing opportunities to increase the value of their digital solutions and enterprise systems. Through agile development of minimum viable products (MVPs), our clients can test new ideas in the market and continually adapt to changing business conditions—giving our clients the leverage to lead market disruption in their industries and compete more effectively to grow their business.