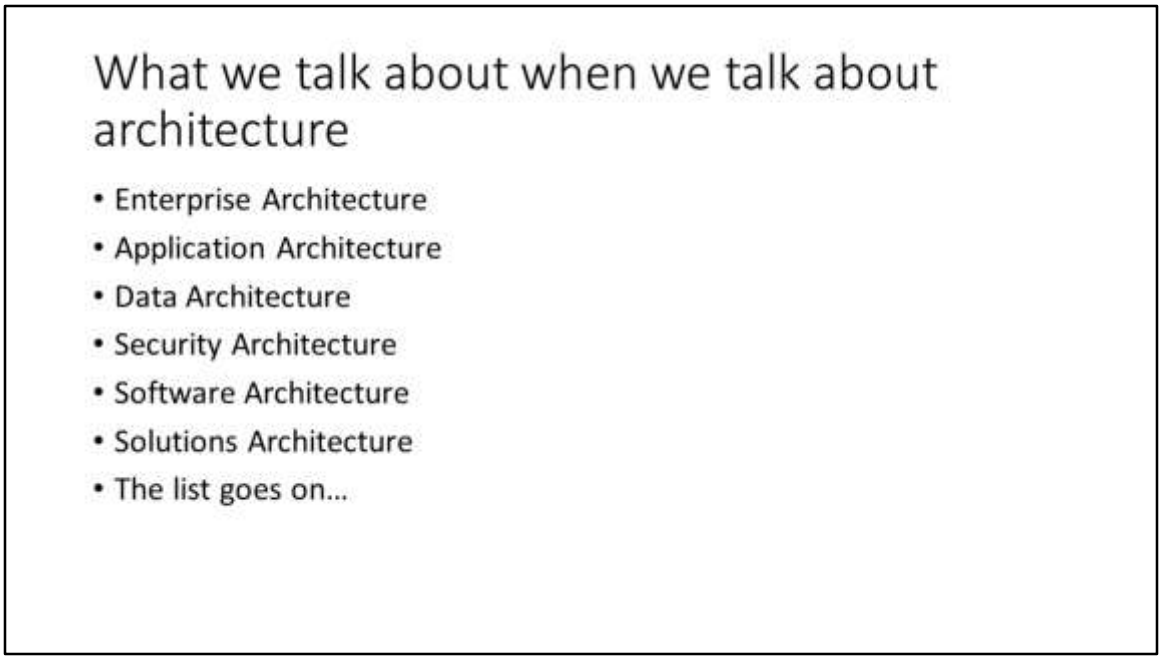
Architecture Research

Responsibility, Business Value, Accountability, and Future State

Timothy Adams
3/29/2018

Thoughts on Architecture



There are numerous domains of architecture, each requiring specific skills and with a different focus. There is no true hierarchy to these architectures. No one domain is comprehensive, enveloping the rest.

However, thinking in terms ranging from general to specific, abstract to concrete, and broad to narrow, Enterprise architecture best matches the former traits, Solutions architecture the latter.

The true value of architecture as a concept lies in bridging the gap from the nebulous to the concrete and tying business strategy to specific business outcomes in a way that benefits the organization.

Because a prerequisite to executing a plan as an architect lies in understanding the value it delivers and the overall system in which that value stream resides, this research will focus more broadly on enterprise architecture for its ability to enable others to succeed.

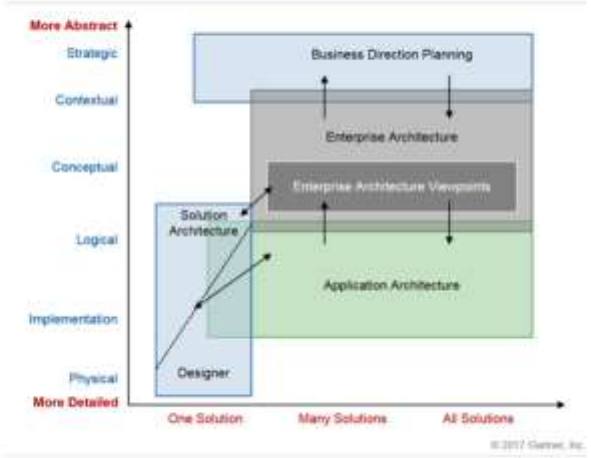
“There is nothing so useless, as doing with great efficiency, something that should not be done at all.” – Peter F. Drucker

There are accepted industry definitions and frameworks for Enterprise Architecture. Fundamentally, at an enterprise level, architects are responsible for understanding organization assets and understanding the systems and contexts of those assets to help the business make good decisions about realizing value from them.

This higher level definition can generalize across organizational domains and scale to different scopes. At any of those levels, the specific responsibilities of an architect will change, as will the deliverables expected of them. Different people having different talents and areas of knowledge may specialize in the intersection of domain and scope, but perhaps there is a notion of a full-stack architect that can somewhat navigate some wider version of them.

I might posit that as scope moves from the more specific to the more general, the distinction between domains becomes less important. Assuming deliverables are different at different levels of scope, there may be an organic way to collaborate across domains by teaming up to produce deliverables that operate at higher levels of scope/abstraction.

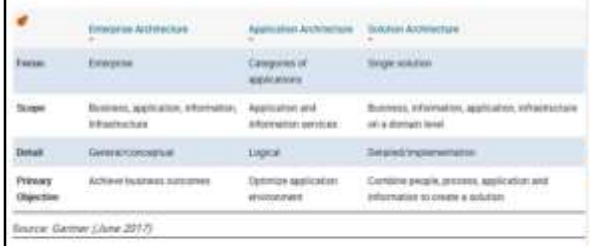
One model for how these pieces fit together?



This visualization does a good job of showing how different categories of architectures might map across the contexts of an organization as well as how they might interact.

I might consider this more of a shifting view of what would be architecturally relevant at which organizational context, rather than a distinct specialization. The following slide shows the focus for each of the defined areas.

What do they focus on? What might they do?



	Enterprise Architecture	Application Architecture	Solution Architecture
Focus	Enterprise	Categories of applications	Single solution
Scope	Business, application, information, infrastructure	Application and information services	Business, information, application, infrastructure on a domain level
Detail	Generic/conceptual	Logical	Detailed/implementation
Primary Objective	Achieve business outcomes	Optimize application environment	Combine people, process, application and information to create a solution

Source: Gartner (June 2017)

Here are some generalizable areas of architectural concerns: things architects might focus on and examples of what they might do.

This should operate as a guide to what is architecturally relevant as one navigates organizational context rather than as a guide to organizational silos. There is an opportunity for an architect to specialize based on ability, skills, education, or experience. However, those assignments would optimally be determined just-in-time by jobs to be done rather than pushed onto the team by external factors not relevant to the business value being delivered.

Thoughts on Architects

The slides in this section have mostly been culled from recent IEEE articles about software architects and their roles in an organization. As such, the focus seems more narrow than the guidance on Enterprise Architecture. However, most of it seems general enough to be applicable across domains. There is generally a smaller body of work specific to data architects and their roles.

The point of this section is to identify how the role of the architect is currently practiced. So, the focus is on responsibilities and value added over the application of specific skills.

The Software Architect's Role in the Digital Age

- Role evolving from mostly technical to include business, social, and cultural aspects.
- Connect business and engineering to align company strategy and technology strategy ("ride the elevator from the penthouse to the engine room").
- Engaged in development, operations, and maintenance.
- Connect and coordinate. Manage complexity and spread knowledge.
- Contribute to quality attributes: business speed and value, cost and risk, technical debt management.

The two key takeaways here are:

1. The role is moving beyond a limited technical focus to include more communication and connection to help deliver business facing value.
2. The role has expanded beyond up-front design work to include the whole of the software development lifecycle and product lifecycle.

The Software Architect's Role in Practice

- Transforming from decision makers to advisors, coordinators, and knowledge managers.
- Emphasis on group decisions: developers make local decisions, groups form consensus global decisions with architects as consultants.
- Architects guide team members through standards/constraints.
- Identify important architectural decisions.
- Transferred knowledge to dev team.
- Explore technologies. Explore potential solutions.
- Document decisions (of value).
- Collect and disseminate patterns and guidelines.

The role of the architect has become less of an authoritarian decision maker to more of a coach. It still retains an ability to make a final decision but there is a bias towards soft skills and helping people make good decisions for themselves.

While there is still a responsibility to document decisions placed on the architect, there is more room to decide which decisions need to be documented (and how to document them) based on the value added by that activity.

Finally, there is the addition of exploring new technology and solutions.

Architect as shepherd:

- Architects must guide and harmonize the entire community of project stakeholders.
- They must bridge the gap between the development organization and the software architecture.
- Play a key role in change scenarios:
  - Adapt the organization and architecture to accommodate the changes.
  - Continuously change the architecture and organization to keep them aligned
  - Understand and mediate existing and new requirements, stakeholders, and concerns
- Architects must balance technical solutions with challenging social and organizational issues.
- Ultimately they must build relationships between people such that the goals of the development community are aligned with good-quality software architectures.

Reinforcing the concept of the architect as a coach or a guide, this also adds a responsibility to manage change. It also provides some clear ideas about how architects might exhibit leadership though soft skills in their roles.

What about data architects?

- Design and maintains the architecture of data science applications
- Creates data models
- Creates data process workflows
- Designs shared information environments involving models or concepts
- Develops data models for optimal performance in databases
- Designs data structures for data interchange
- Develops data standards and converts data to controlled vocabularies
- Structures the technology that manages data models

This gets a little more specific into the types of things some data architects might do. All of the previous leadership and high level responsibilities of the role likely still apply. However, these specifics might be important to consider in the context of the data architects domain when considering success factors in the next slide, where it is important to match the architects skills to their role based on the phase of the project.

What makes an architect successful? ("It depends.")

- Software architecture = "...the structures needed to reason about the system. Each structure comprises elements, the relations among them, and the properties of the elements and relations."
- Or the bridge between the system's business goals and its realization.
- Success is matching the architects skills to the role:

FIGURE 1. Software architects' three roles over a system's lifetime.

Initial designer	Extender	Sustainer
• Create and defend conceptual integrity	• Compromise conceptual integrity to accelerate value creation	• Assess and communicate conceptual integrity in new environment
• Understanding of all system structures and abstractions	• Understanding of as-built system, including undocumented side effects	• Understanding of system's environment and of competitive systems
• Proficiency with architecture design patterns	• Proficiency with architecture integration patterns	• Knowledge of how the system creates value for users or customers

FIGURE 2. Comparison of software architects' required skills across the three roles.

As specific to software architects as we get and focused on a specific system. This lives at the solutions architecture level from earlier, but shows the expanded role of the architect from simply an initial designer to a maintainer and extender as well. Even at this level, it is becoming more important for the architect to understand the business goals the system is intended to meet and to help make trade-offs between the value delivered and the architecture of that system.

References

1. Waterman, Michael. "Agility, Risk, and Uncertainty, Part 1: Designing an Agile Architecture." *IEEE Software* 35.2 (2018): 99-101.
2. Klein, John. "What Makes an Architect Successful?." *IEEE Software* 33.1 (2016): 20-22.
3. Weinreich, Rainer, and Iris Groher. "The Architect's Role in Practice: From Decision Maker to Knowledge Manager?." *IEEE Software* 33.6 (2016): 63-69.
4. Hohpe, Gregor, et al. "The Software Architect's Role in the Digital Age." *IEEE Software* 33.6 (2016): 30-39.
5. Tamburri, Damian A., Rick Kazman, and Hamed Fahimi. "The Architect's Role in Community Shepherding." *IEEE Software* 33.6 (2016): 70-79.
6. da Silva Amorim, Simone, et al. "The Architect's Role in Software Ecosystems Health." *Proceedings of the 2nd Workshop on Social, Human, and Economic Aspects of Software*. ACM, 2017.
7. Slot, Raymond, Guido Dedene, and Rik Maes. "Business value of solution architecture." *Working Conference on Practice-Driven Research on Enterprise Transformation*. Springer, Berlin, Heidelberg, 2009.
8. Lavallée, Mathieu, and Pierre Robillard. "Are We Working Well with Others? How the Multi Team Systems Impact Software Quality." *e-Informatica Software Engineering Journal* 12.1 (2018): 117-132.
9. Gartner: Leadership Vision for 2019: Enterprise Architecture and Technology Innovation Leader, **Published:** 23 January 2018 **ID:** G00348586, *Analyst(s): Betsy Burton | Marcus Blosch | Brian Burke*
10. Gartner: Avoid the 13 Worst EA Practices and Ensure Your Success in the Digital Business Era, **Published:** 15 January 2018 **ID:** G00343827, *Analyst(s): Saul Brand | Betsy Burton | Marcus Blosch*
11. Gartner: Storytelling for Enterprise Architecture: How to Influence and Persuade Leaders of EA Value in Decision Making, **Published:** 29 January 2018 **ID:** G00350356, *Analyst(s): Philip Allega | Bruce Robertson | Ed Gabrys*
12. Gartner: Stage Planning a Business-Outcome-Driven Enterprise Architecture, **Published:** 16 March 2017 **ID:** G00321942, *Analyst(s): Brian Burke | Betsy Burton*
13. Saltz, Jeffrey S., and Nancy W. Grady. "The ambiguity of data science team roles and the need for a data science workforce framework." *Big Data (Big Data), 2017 IEEE International Conference on*. IEEE, 2017.