A Day in the Life of an Enterprise Architect

Mike Walker
July 2007

Introduction
Enterprise architecture has grown from being just a set of small pilots to being a fully sponsored and supported initiative within enterprises. With the growing demands to reduce costs, increase agility, and standardize IT environments, there has been a surge of enterprise architecture activity. According to Gartner and the MIT institute the growing complexities that span across process, information and software are among the three top CIO concerns. Additional pressures come from regulatory bodies that impose compliance guidelines on the industry (such as the Clinger-Cohen Act of 1996 or FFIEC Guidance). Compliance has been a catalyst for the formation of an enterprise architecture practice. This article will walk you through the daily challenges that an enterprise architect faces. By doing so, we hope to provide a unique perspective on this growing role in the IT industry.

We will take these observations of the enterprise architect role and contrast them with more defined roles in IT, such as the role of the developer. The enterprise architect (EA) role is both complex and dynamic. After reading this article, you will have a clearer understanding of the enterprise architecture role and the challenges faced by those who fill it. If you are an enterprise architect or aspiring to be an enterprise architect, this is a must-read for you. The guidance found in this article is tailored for mid-size to large enterprises. If you fit other criteria, some or all of the guidance may apply. Prescriptive guidance around enterprise architecture strategies, tooling, and governance will not be covered in length.

The term “enterprise architecture” can have many different meanings, and we often find that there are only particular aspects examined. To compound the issue, information technology professionals scope enterprise architecture differently. Some IT professionals view enterprise architecture as a more structured activity with many moving parts such as architecture strategy, architecture patterns, principles, architecture design, business architecture, and IT governance. Others may just focus on the technical aspects rather than the business, operational, or strategic aspects. Many view enterprise architecture as an IT-only issue. In many cases, it is important that the enterprise architecture group speak in both business and IT terms. EA should be neutral in their functions.

How this Article is Structured
This article will walk you through the life of an Enterprise Architect. To do so, we will surface the following aspects of the role:
• **The Role of an EA** – In this section we will talk about what it means to be an enterprise architect. The skills and the responsibilities that are required to be an EA.

• **How EA’s are Different** – We will contrast with various other roles to give a clearer picture of how EA’s fit within the architecture landscape.

• **The Tools they Use** – We will get an understanding of what tools are needed in an EA function. A scenario will also be given to understand the importance of “connected” EA tooling.

• **How They Make Decisions** – We will dive into what happens in the mind of an EA when making complex decisions.

• **How they Fit into the Organization** – We will show where EA’s fit into the organization and how they influence our processes and decisions.

• **Challenges they Face** – In this section we will talk about the challenges that EA’s face on a daily basis.

Additionally, we have collected quotes specifically for this article. They come from the two largest software companies in the world, Microsoft and IBM. Both IBM and Microsoft have significant impact, interest and experience in this area.

**The Role of an Enterprise Architect**

On a daily basis, an EAs activities can change quickly and dramatically. For the sake of this article, we will not go into organizational models of enterprise architecture organizations. Instead, we will explore the role and responsibilities of an enterprise architect. Understanding the role of an enterprise architect will help us understand the challenges an EA faces on a daily basis.

The addition of the word "enterprise" extends the coverage of architecture to span both IT and the business realm — including the mission, value chain, business strategy, business functions and business processes, which together describe and define the role of the enterprise. The business-level description may or may not include the application of IT. Enterprise also implies the entire breadth of the enterprise, as in enterprisewide architecture. This is added to the traditional architectural coverage of information, or "solution" architecture and technical architecture.

*Figure 1: Enterprise Architecture as defined by Gartner*

Shown above is Gartner’s definition of enterprise architecture. Unfortunately, this in one of many; there is no consistent set of definitions for enterprise architecture, which leads to further confusion in the industry. For the sake of this article, we will use Gartner’s definition as our interpretation of enterprise architecture.
There are a broad range of essential skills that every EA should have. Obviously, an EA must be well-educated in technology. Ideally, EAs should come from a highly technical background. Even though enterprise architects deal with many other factors besides technology, it is still important to keep technical skills fresh. Engaging on projects at a micro level of detail can serve as a great refresher from time to time.

Technology skills are not the only skills you need to be an EA. There are other skills that are essential, including:

- **Motivational.** EAs must be able to motivate and inspire. A large part of the job is to influence or evangelize a set of ideals in the enterprise.
- **Negotiation.** There will be times at the decision-making table when an EA must negotiate to get things accomplished. Most EAs are individual contributors and do not have organizational power.
- **Critical Thinking.** Being able to think quickly and on your toes is often required.
- **Problem Solving.** EAs often face a set of complex and unique problems, so they must be able to evaluate and solve problems.
- **Big Thinking.** Avoiding tunnel vision and being able to look at a problem from multiple angles to test your own rationale is crucial to an EAs role.
- **Business savvy.** Knowing the industry in which you work is essential, to help you understand how the technology can really affect the business. Being in tune with the business gives EAs much-needed credibility.
- **Process Orientation.** Thinking in terms of process is essential for an EA. Building repeatable and reusable processes as both as artifacts from the work they do to how they work themselves.
- **People Skills.** An EA’s job requires interacting with people constantly, so not having those skills could mean trouble.

Skill sets aside, the role that an EA plays is multi-faceted. The most common function an EA will perform is that of large-scale program oversight. Programs are multiple related projects represented as a package. Managing programs generally requires a person that is able to handle multiple aspects of a project at one time.

"When I hire for Enterprise Architects, I look for individuals who have an exceptional ability to communicate, deal with political situations, and take on big bold organizational challenges. If all s/he brings to the table are strong architectural abilities, I pass on that individual and keep looking”.

- Kathy Watanabe, Microsoft Chief IT Architect

An EAs role spans more than just strategic or large-scale programs. EAs can also cover the following aspects of a job:
- **Governance Committees.** These committees make decisions on what the organizational standards and policies should be in an enterprise, such as the supported protocols for business partners. There would be very clear business, security and technology requirements that the governance committee would use to determine the repeatable patterns, and processes should be employed across the organization.

- **Architecture Review Boards.** Another group of people that convene to approve architecture to ensure that it is sound for the organization.

- **Technology Life Cycles.** This function determines how individual technologies or standards will change or be versioned, such as deciding that a Get Customer service will change every six months and will support two new lines of business every year. They can also determine whether technologies or standards will stay or be transitioned out of the organization.

- **Portfolio Management.** Enterprise architects are often linked to and accountable for part or all of the health of the IT portfolio, although they are usually contributors to the IT portfolio, not owners. As shown below in figure x.x this is often referred to this aspect is referred to as Asset Portfolio Management (APM). EA’s add vital information about the technology and provide insights into the technology life cycles that govern them.

- **Architecture Strategy.** This activity encompasses all or strategic areas of IT. It often includes gathering a current state (asking “What do I have?”), a transition state (asking “What iterative steps do I need to get to my goal and what does that look like?”), and a future state (asking “What are the changing aspirations of the organization?”).

- **Strategic Project Support.** Often, EAs are brought into important strategic projects to aid in ensuring the success of the design process.

A good way to compare enterprise architects with other types of architects is to look at them in terms of breadth vs. depth. This helps us to see the relationship between depth in expertise, such as depth of understanding in Windows Infrastructure Architecture, and the span of responsibilities that the architect covers, such as breadth of understanding in a Line of Business (LOB).

**How Enterprise Architects are Different**

As shown in Figure 3, enterprise architects cover the breadth of IT and, to a degree, the business concerns as well. When compared to other roles, you can see that the expertise of an EA is very different.

Let’s take a look at how these roles are different from each other based on breadth vs. depth:

- **Enterprise Architect.** Enterprise architects span across LOB and IT domains such as security, infrastructure, information, or development.

- **Domain Architect.** Domain architects focus on a specific domain and have deep expertise that area. Typically these architects only focus on specific areas, Examples of these include: Business Architect, Security Architect, Information Architect, Infrastructure Architect, Communications Architect, etc.
- **Developer.** Developers focus on one solution at a time and have a deep expertise in development.

Figure 3: Architecture roles, breadth vs. depth illustration

Figure 3 shows an enterprise architect in comparison to other architects. There are many different architecture operating models. The roles shown above are a sample of a structure that demonstrates how each of these roles might align.
**Figure 4 – Many different organizational structures**

Figure 4 shows two examples of diverse architecture organizational models, separated by a line. In some cases these organizational models are not hierarchal in nature. Rather, they address the pre-EA operating models. Even though this is not ideal in all cases sometimes it is necessary to gradually introduce change to avoid pushback in the organization. Each of these can have an impact on the role in which an EA operates, and each carries contrary organizational and operating challenges.

Clearly, there isn’t just one criterion for making IT decisions. Decisions are made from multiple perspectives. Often these decisions are not made based on the merits of the technology in question but on organizational aspects such as cost, personnel supportability issues, and IT standardization.

**Tools Enterprise Architects Use**

Since EAs have a multi-faceted role, in this article we will focus on some of the challenges and not all of them holistically as the goal of the article is to highlight the day in the life of the enterprise architect.

One of the larger issues with the role of an EA is having the right tools that enable them to succeed. Since EAs work at a few layers above developers, you will not see developer tools in their arsenal. What you will find are modeling and communication tools.

You can expect to see the following applications in the Microsoft® Windows® Start Menu’s top applications launched:

- **Microsoft Word or Microsoft Excel®,** to document and report on architectures
- **Modeling tools (usually Microsoft Visio®),** to create models of the architectures
- **Microsoft PowerPoint®,** to communicate architectures
- **Management tools,** which could consist of internal applications or portfolio management tools

The rest of the tools usually depend on the role of the EA and the standard applications that are used in the organization for these functions.
The goal now should be to leverage the capabilities of these tools to optimize your processes. We want to focus on the tooling because the processes and roles are often different from organization to organization.

**Scenario: Using Microsoft Tools for Enterprise Architecture**

There are many ways you can leverage your existing tools without much effort or cost that will give you a substantial gain on your EA productivity. In Figure 6, you can see a sample solution to the documentation of enterprise architectures by utilizing standard Microsoft Office tools are used to document architectures.
In this very high level scenario, you could be working on an enterprise effort where you are capturing the current state of your enterprise or a project effort. You can create Microsoft Word templates that are used for capturing architectures. By doing so, you have defined elements that are consumable by a downstream process for consumption and workflow.

Many processes can be developed from this once the data is in a consumable format. You have created a level of consistency in how you document architectures, making it easier to leverage information that has been documented when making IT decisions.

By centralizing the information, you have created a repository for all of your architecture information. This facilitates governance functions, such as security, policy management, information management, approvals, and alerts. Reporting and metrics are a downstream benefit that can be used after all the information is classified.
How They Make Decisions

Decisions made by Enterprise Architects are complex and span multiple domains. EAs are not only pulled into multiple directions but are also required to switch between levels of detail as well. It is common to go from macro to micro and back again between meetings.

When making architecture decisions, either on a project level or enterprise level, there are always many factors that go into the decision-making process. At times, it consists of a set of compromises to support a more important factor.

![Diagram showing architecture trade-offs between Technical and Organizational factors with subcategories like IT Operation, Technology Decisions, Culture, Supportable, and Business Drivers.]

*Figure 7: Architecture trade-offs*

As shown in Figure 7, the organizational factors can be significant in how technology decisions are made.

In figure 8, you can see that there are three major classifications of the separation of decision making that is asked of an EA. Depending on your organizational structure, these may look a little different. However, each one of these is important for EAs to advise on.
How They Fit in the Organization

Figure 8: Three major classifications of the separation of decision making

Organizational Policy
Policies that span the organization are often used to facilitate IT governance. By defining a set of policies or rules, the organization is aware of which technology practices are right for their organization. When the time comes to do a formal review, the IT community will also know what will be the basis of their evaluation.

EAs often participate in the committees that are responsible for building the overarching policies and standards that will be used throughout the enterprise for decision support. The expertise of an EA in these exercises is extremely valuable, as they have insight into both the long-term strategy and their organization’s IT landscape.

Organizations have seen significant benefits by going through this exercise, including:

- IT cost savings
- Improved vendor relations
- Focused IT
- Empowerment of architects and developers

IT Strategy
Strategy creation is a core function of an Enterprise Architect. There are three base activities that an EA will perform:
• **Defining the future state** (sometimes referred to as the “to-be” architecture). This provides a road map for your enterprise to follow. This will also aid in building your transition state architectures.

• **Capturing the current state architecture** (sometimes referred to as the “as-is” architecture). This activity helps you figure out “what you have”. By doing this, you can figure out what is working, identify duplications in the enterprise, or measure the health of key business processes that are supported by your architectures.

• **Building the transition architecture**. Transition state architecture is the process by which we connect the current to the future state by creating an iterative roadmap to get to the desired future state.

In Figure 9, you can see the IT Strategy Life Cycle. It is important to note that the following activities are not listed in order.

> “Enterprise Architects manage the technology strategy and implementation as it relates to the business strategy and objectives.”

> - John Adams, IBM Executive Architect

Capturing current state architectures is essential for grounding your efforts to figure out what needs to be changed or transitioned. However, it’s best not to capture current state as the first step. It may seem like a good idea, but it often causes EAs to lose sight of the end goal and get lost in the details.

As shown in Figure 9, the lifecycle for IT strategy is iterative. This means that when you capture your current state once, it isn’t necessarily complete. Be careful not to bite off more than you can chew in each step—divide work into small and consumable chunks to ensure success.
When building future state architectures, we take the current state into account. When the future state models are complete, they can act as a road map that should be used when making technology decisions. It is an indispensable tool for your architecture thought processes and decisions, because it provides context and perspective about the progress of the architectures in your organization.

Transition state architectures enable future state models and plans. It provides changes to the organization in easily digestible iterative changes. Instead of the “Big Bang” approach where everything needs to change at once, this provides a set of steps to move into the direction of the future state gradually.

As a downstream benefit, it can also provide a test bed for new ideas that may not pan out, so that they can be tried with minimal business disruption and financial impact to the business.

The future state is documented mostly in models. As shown in Figure 10, this makes it easier to communicate the complexities of the architectures.
When building out future state models, it is important to remember that these are projections of the future. Businesses change and adapt to industry trends, and they can go in multiple unforeseen directions. It is important to be agile with your EA organization by building a defined life cycle around reviewing the future state architectures.

Program and Project Decisions
When an EA engages with project teams at this level of decision making, we see a solo view of decision making. Project teams that are isolated to only a specific LOB often develop these narrow perspectives. This can inhibit an EA’s efforts to unify areas across the enterprise.

Often EAs run into statements such as, “We have to have this package” or “We want to build it our way because it works.” Having such a limited view on the decision without considering the bigger picture is one of the largest reasons why more than 75% of IT work consists of maintaining solutions.

When EAs make decisions at this level, reviewing the work done on the organizational policies and IT strategy can save the day. One effective way to ensure traceability is to create a tool that aligns these components together. This has multiple benefits:

- Ensures IT decisions are made consistently across the enterprise
- Creates repeatable process
- Facilitates traceability to the overall strategy of the organization
- Empowers architects to make the right decisions
- Removes the conflict of interest in decision making
- Enables auditability and accountability of decision making
- Eases governance
Whether or not you formalize this process, it is important for EAs to take a step back and not focus on the tree, but look around the forest and ask some questions, such as:

- Is this a unique solution, or is there something similar already in my organization that can be re-purposed?
- How does this problem align to Buy vs. Build policies?
- Does this apply just to this domain or multiple domains?

**Challenges they Face**

There are many aspects of an EA’s job that do not involve technology. Often there are organizational considerations or barriers. As mentioned previously, EAs require essential people skills to be successful. Technical savvy is not enough; EAs must also possess great communication skills, be homed in on the business, and be great decision makers.

Why are organizational barriers so prevalent in this role?

- **Appetite.** Appetite refers to what an organization can support or is willing to support. For example, if one of the organizational principles is to buy software rather than build it, then proposing building a solution may be contrary to what the organization can support. The ability to support solutions is a factor that should always be considered when proposing solutions. Another aspect of appetite is politics within the organization that will need to be considered.

- **Maturity.** Often EAs have progressive ideas for moving an organization forward with more advanced concepts or architectures. However, the organization may not be ready for those ideas yet, due to factors such as education, infrastructure, and software capabilities. Getting an idea about how mature the organization is will help EAs propose iterative approaches to advanced concepts and solutions.

- **Incentive.** Localized interest—asking “What’s in it for me?” on a personal or team level—is a common organizational barrier. EAs are typically individual contributors and have little or no organizational powers. When working with teams whose incentives are based on their own organizational goals, it can be difficult to propose ideas that span the organizational unit.

In addition to complex decision making, there are organizational forces that challenge an EA. These forces often have a significant impact on an EA’s group, which can lead to poor perception or cause an EA’s group to disband into the organization.

There are three core organizational barriers that EAs face:

- **Buy-In.** Even in non-hierarchal organizations, this is often a challenge. When there are no management ties between you and the personnel you wish to influence, it becomes increasingly challenging for EAs to accomplish their goals. Gaining buy-in from your internal organization on strategies, projects, or their participation on key activities can be daunting.

- **Resources.** Even if a LOB manager buys in on a strategy, shelling out the cash or personnel to support it can be a different story altogether. In many cases, since EAs span across the
enterprise, there are resource needs not only from that business owner but from multiple functional areas. To be clear, EAs generally do not have:

- **Budget.** In most instances, EA groups have no formal budget, so to be a successful EA you must be good at justifying when you need a share of a group’s financial resources. This could be business justification or from an operational perspective.
- **Personnel.** Projects need personnel resources as well as financing, so EAs often have to go through another justification effort to get IT and business groups to provide personnel for projects.

- **Influencing Current or Existing Projects.** Getting in on the front end of a project is one thing, but influencing a project already in motion can be very difficult. In an ongoing project, teams are often reluctant to take an EA’s input or will bypass EA efforts because of the potential impacts to timelines and incurred cost that changes to their architecture may have.

Shown in Figure 12 is Forrester Research’s representation of the Enterprise Architect’s Dilemma. This illustration shares some of the basic themes highlighted above. It provides a good high-level understanding of the organizational barriers.

**Figure 12: Forrester’s illustration on the Enterprise Architect’s Dilemma**

**Obtaining Buy-In**

When obtaining buy-in, it’s usually more important to be able to speak to your audience in terms they understand than to demonstrate the worth of the technology. Most times, this means speaking in
business terms. By speaking in business terms, you can qualify and quantify the problem, and you can clearly show that the change isn’t just a change for technology’s sake but rather for a business purpose.

“In a typical work week, in order to ensure the success of our Enterprise Architecture program, I spend 50% of my time addressing the organizational barriers.”

- Kathy Watanabe, Microsoft Chief IT Architect

There are very few tools that address this concern directly; a combination of resources and tools are most often used, such as:

- Project Dashboards
- IT Environment Health and Change Management Statistics
- Business Analytics
- Current State Architecture Analysis

It is important to collaborate and validate information with stakeholders. This ensures buy-in from the stakeholders. Portal tools that allow strategies and the corresponding collateral (such as justification or metrics) to be published can aid here.

Other tools such as Excel and PowerPoint can also help demonstrate the need for a project.

**Getting the Resources**
Getting a director or vice president to agree that you have a good idea doesn’t always equate to getting cash and valuable resources from them. Justifying why you need those resources can be a big challenge. EAs must be able to fully explain why someone else should shift resources, priorities, and budget.

There should be full transparency in the process when getting buy-in, preventing any surprises when it comes time to ask for money and personnel.

**Influencing Current Projects and Affecting Existing Solutions**
Influencing existing solutions or projects that have already been started is very different from obtaining buy-in and resources for a prospective new project. The primary difference is that in these situations you are affecting work that has already been set in motion, such as a project that has started in a stage of the Software Development Life Cycle or a solution in the IT portfolio. Obtaining buy-in and resourcing is often performed in the planning stages when resource allocation and funding has not been finalized yet. In this case, work has started or is complete. This is where negotiation skills are vital.
At times convincing project teams that you add value can be a challenge. Especially in organizations that have an immature Enterprise Architecture practice

- John Adams, IBM Executive Architect

First, let’s focus on how an EA can influence the SDLC processes by interweaving into a current project. Each organization usually has slightly different terminology to represent an SDLC; we will use the Microsoft SDLC terminology for this article to keep consistency of terms. Figure 13 shows the SDLC process and where there are potential touch points in the process. Again, the specific naming of the phases may be different in your environment.

![SDLC Diagram]

Figure 13: Potential Enterprise Architecture touch points in the SDLC process

Even though an EA could participate in multiple phases of the SDLC, it doesn’t necessarily mean that they should. EAs most often engage at the end of the envisioning phase and the start of the design phase.

Let’s walk through each phase and highlight how an EA can help:

- **Envision.** During the envisioning phase, there are some critical processes that determine both the course of the project and the application assets that are to be purchased or developed. These processes concern how the solution will fit into the environment, how it should be architected, and whether an organization should build, buy, or buy and customize a solution. If
vendors will need to be selected, this is a critical stage for an EA to provide input before in-depth vendor management occurs.

- **Design.** Typically EAs will rely on domain architects and developers to get down to the micro level of detail. However, EAs need to be able to see how all the pieces fit in the enterprise and review the architecture as a whole once it is complete.

- **Build.** During this phase of application coding, EAs shouldn’t be involved unless there is an explicit need.

- **Stabilize.** During testing cycles, it is inevitable that the application will need to be changed in some way or another. If there are significant changes to the architecture, it may warrant an EA to review the changes and aid the project team to steer them in the right direction.

- **Deploy.** When going through the operational aspects of deploying software to environments and instilling change management, EAs shouldn’t be involved unless there is an explicit need.

- **Production.** During this phase, an EA can begin to work with post-production resources responsible for the maintenance of the application. They can build out how the technology will be incorporated in the IT portfolio with a subsequent life cycle, how it will be versioned, build out of left over architecture decisions, if there are any service endpoints or management of endpoint.

Changing existing solutions that have already been deployed into a live production environment can be a real challenge. Statements such as “it’s not broken, so why should I fix it?” or simply “it works and I’m not paying for customizations” are all commonly used, but they are valid statements to address nonetheless.

When EAs move into the post-production phase of an application’s life cycle, where there are already systems put into place, they will find several areas of concern. In Figure 14, you will see that there is a great deal to consider.
There are impacts to multiple areas, each of which has its own unique set of challenges. The three major areas that should be considered are:

- **Production Processes.** These processes support the promotion of software to production environments, change management, and support of solutions after they are in production.
- **Production Systems.** Systems that are in the production environment are often not isolated; they are deployed and configured into environments that have dependencies and restrictions.
- **Production Teams.** Teams that support and deploy these solutions have unique processes and procedures. There is both a process and an organizational perspective on this.

**Production Processes**

EAs should be mindful of production processes because they affect the cost, quality, and resiliency of software. EAs can have a positive impact on these processes by being involved in the following core production processes:

- **Configuration Management.** EAs can optimize these efforts, both in the design of the architecture and in providing insight into the rest of the organization, possibly standardizing this process.
• **Change Management.** EAs are typically not involved in this process, but they need to be mindful of the impacts to solutions, since they could have many different relationships with other solutions and altering a solution could create downstream challenges.

• **Incident Management.** EAs do not generally engage in this process either, but they need to be mindful of it, because incident management data can be of great value. The data collected here can correlate with other data to help EAs gauge how much an architecture costs.

**Production Systems**

EAs perform a set of activities that involve existing production systems quite often. By doing so, they serve multiple roles, both in participation and leadership for the following activities:

• **Future State Architecture.** When EAs determine a direction for a set of business problems, a solutions road map and architecture envisioning occurs.

• **Current State Review.** As mentioned previously, EAs generally understand the solution architectures more deeply. This process involves an EA engaging with a LOB owner or post-production maintenance teams.

• **Strategic Initiatives.** EAs can shape strategic initiatives that result when other forces besides a formal planning process trigger evaluation of current solution architectures.

EAs encounter both technology and operational aspects when reviewing and re-architecting solutions. It’s important to keep in mind that these concerns are not just related to software, but can include a mix of hardware, communications, and software aspects. These aspects stem from a set of enterprise functions, which include:

• **Shared Services.** EAs consider whether or not particular solutions should use shared services.

• **Solution Dependencies.** Solutions often communicate with other solutions for additional functionality. Unless the current state architecture is fully mapped, there is a seemingly endless amount of interdependencies throughout enterprises.

• **Environments.** EAs often consider unified management and consolidation of platform environments (such as Windows 2003).

• **Constraints.** EAs take in limitations or constraints to architectures for various reasons. Some COTS-based solutions limit the API usage, for example, while other custom-developed solutions are built not to be extensible.

**Production Teams**

Various post-production maintenance teams are required to do most work on existing architectures, because design documents are created during the SDLC process that can quickly become outdated. Unless the architecture is fully documented through the post-production life cycle, EAs rely on these teams. Teams that are engaged usually consist of:

• Maintenance Team
• Operations Team
• User Support Team
These teams offer perspective into multiple domains of consideration when making architecture decisions.

**Dealing with Governance**

Governance facilitates the orderly development of software in the enterprise by using a set of processes and operating models, and by policy development. Governance defines the processes by which software should be built. There are multiple aspects to the development of policy that derive from business drivers, industry trends, competitive forces, and corporate and IT strategy. This article will not go into great detail into this subject except to discuss how it intersects with the day of an enterprise architect.

After an EA builds policies and standards and starts to vet them in the enterprise, a governance model emerges. Governance can be tricky, because organizational culture plays a huge part in this process. Additionally, each organization can have its own model, as shown in Figure 15 - TOGAF Architecture Governance Framework. It highlights the major structural elements required for an architecture governance initiative.

![Figure 15: TOGAF Architecture Governance Framework - Organizational Structure](image-url)
Governance models are solely based on organizational culture and maturity. This is not a bad thing; it’s actually very good for many reasons. These include:

- Adapting to the culture of the enterprise
- Changing perceptions in the organization
- Allowing resources in the organization to ramp up their educationally

IT governance is an activity that EAs are engaged in regularly. It is important to note that day-to-day governance activities have a subtle impact on the overall success of an organization, in the long-term results that yield returns.

Aside from the more formal governance model, EAs are usually involved in several governance activities. These activities include:

- **Architecture Reviews.** These are usually in the form of committees responsible for evaluating architectures. In this, EAs can address technology standards adherence.
- **Building of Policies and Standards.** Policies and standards are generally built by forming a committee that has representation from key areas.
- **Creating Design Patterns.** Documenting how to employ the policies and standards is essential to convey the architecture concepts. It also provides the development community with building blocks to build compliant architectures.

Falling into the trap of needing to command and control is a common mistake. Fostering collaboration with the teams is more effective than simply telling them what to do. When EAs mandate change unilaterally, they are viewed as “speaking from an ivory tower” and lose credibility. EAs should enable and empower the community to make the right decisions. This gives EAs creditability and support, because they are not viewed as micro-managing.

The end goal should be to raise awareness with developers so that they will pay attention to the architecture decisions they make when they implement solutions.

**Conclusion**

An enterprise architect’s role is multi-faceted and extremely dynamic. Not only must they keep track of IT concerns, but also those of the business. Through the work performed on strategic initiative, EAs strive to make alignment between IT and the business more transparent.

Being an individual contributor with no alignment to the business or technology LOB group, this often necessitates that an EA must call more on negotiation skills than technology skills. In this way, EAs must be able to use an ability to influence the organization to do the right thing.
It is important to remember that enterprise architecture is about fostering community by providing reusable frameworks for decision making, repeatable processes, and assistance on mission-critical projects.

References
MSDN Enterprise Architecture Center at http://msdn.microsoft.com/architecture/EA