Relating Enterprise, Application and Infrastructure Architects

Eoin Woods
Artechra, UK

Nick Rozanski
Artechra, UK

ABSTRACT

The architect takes a high-profile role in many IT departments today. In fact, it can be quite difficult in some organizations to find a senior member of IT technical staff whose job title does *not* include the word "architect." However there is little consensus in the academic community or amongst practitioners as to the responsibilities of the many different types of architect we encounter – or indeed, what they should even be called.

In this article we propose a simple, widely applicable, taxonomy of architects, namely *enterprise architects*, *application architects* and *infrastructure architects*. We define their distinguishing characteristics, their responsibilities, the stakeholders with whom they engage, and the tools and techniques they use. We show how this taxonomy can be applied to most, if not all, practicing architects in the information systems domain, and explain how it helps us understand how such architects work together to help deliver the organization's business goals.

INTRODUCTION

In recent years we have seen the role of the IT architect take a much higher profile in enterprise computing. In fact, today it can be quite difficult to find a senior member of IT technical staff who is not referred to in some way by the title "architect", for example an "enterprise architect", an "integration architect" or a "Unix architect". However the lack of widely accepted definitions of the responsibilities of the different types of architect causes quite a lot of confusion.

In this article, we attempt to remedy this state of affairs. We believe that there are some defining characteristics that all architects share, and that it is possible to identify a small set of architecture roles into which all of the different architecture jobs can be placed and which can be used as a test for a role actually being an architecture role (rather than that of, say, a technical expert).

Our taxonomy of architects comprises the *enterprise* architect, the *application* architect and the *infrastructure* architect. We characterize and classify these groups in terms of two specific aspects of their jobs: the *breadth of focus* that they need (e.g. the number of systems they are interested in) and the mix of *domain and technology knowledge* that the job requires.

RELATED WORK

When we started thinking about the relationships between the various sorts of IT architect, we tried to find existing approaches for classifying the blizzard of job titles in this field. However, although there are plenty of references to the various job titles in formal and online literature, we found relatively few attempts to discuss how the different roles related to each other.

What we did find were a number of classification schemes used in career development or certification programs (namely those from IASA, IBM, CapGemini and the Open Group) and a number of discussions of the responsibilities of software architects specifically.

As part of its attempt to establish and formalize a profession of software architecture, the International Association of Software Architects (IASA) defines five distinct specializations (IASA, 2011) within software (or IT) architecture, namely enterprise architecture, software architecture, infrastructure architecture, information architecture and business architecture. IASA defines these roles as follows.

- Enterprise Architecture describes the terminology, the composition of enterprise components, and their relationships with the external environment, and the guiding principles for the requirement, design, and evolution of an enterprise.
- The Software Architecture of a system is the set of structures needed to reason about the system, which comprise software elements, relations among them, and properties of both.
- Infrastructure Architecture describes the structure and behavior of the technology infrastructure of an enterprise, solution or system. It covers the client and server nodes of the hardware configuration, the infrastructure applications that run on them, the infrastructure services they offer to applications, the protocols and networks that connect applications and nodes. It addresses issues such as performance and resilience, storage and backup.
- Information Architecture is the art of expressing a model or concept of information used in activities that require explicit details of complex systems. Among these activities are library systems, Content Management Systems, web development, user interactions, database development, programming, technical writing, enterprise architecture, and critical system software design
- Business Architecture is a part of an enterprise architecture related to architectural organization of business, and the documents and diagrams that describe that architectural organization.

Interestingly, one local chapter of IASA, IASA Sweden, independently looked at the same question of specialization within software architecture (Akenine, 2008) and identified a slightly different set of roles, namely enterprise architecture, business architecture, application architecture and software architecture.

IBM defines IT architecture as one of the career development professions within the company (Yi Qun, 2009) and again identifies a number of specializations within it, namely enterprise architecture, application architecture, information architecture, infrastructure architecture, integration architecture and operations architecture. These specializations naturally reflect the type of work in which most of IBM's architects are involved.

- Enterprise Architecture is concerned with the definition of a high level enterprise-wide IT Architecture focusing on the mapping of IT capabilities to business needs.
- Application Architecture is concerned with the design of applications required to automate business processes and resolve business issues.
- Information Architecture is focused on the elements required to structure the information and data aspects of solutions and also required to design, build, test, install, operate and maintain the system of solution information.
- Infrastructure Architecture is concerned with the design of infrastructures including servers, storage, workstations, middleware, non- application software, networks, and physical facilities that support the applications and business processes required by the client.
- Integration Architecture is focused on the design of solutions which enable existing applications, packaged software offerings, networks, and systems to work together within an enterprise or among enterprises.
- Operations Architecture is focused on defining plans, strategies and architectures for the installation, operation, migration and management of complex information systems

Cap Gemini's IAF (Integrated Architecture Framework) (van't Woutn, 2010) defines the following types of architecture:

- Enterprise architecture, which supports enterprise wide decision making and planning, and shapes the enterprise landscape;
- Domain architecture, which supports business unit level decision making and planning, and shapes the domain landscape;
- Solution architecture, which provides architectural guidance to programs and projects (that is projects or large programs of work which are involved with the design, installation, commissioning and integration software systems – usually packaged solutions – to provide an entire functional solution for an organization, rather than just a piece of software or a standalone system);
- Software architecture, which provides architectural guidance to software development (that is the process of developing a bespoke system, service or major reusable component from scratch, rather than installing and integrating an existing large scale reusable application, such as a CRM system).

There have also been a number of attempts to characterize the work of the software architect (an "application architect" in our classification), such as (Fowler, 2003), (Vickers, 2004) and (Kruchten, 2008). All of these descriptions focus on the single role of an architect creating a single system, rather than the relationship of software architects to other types of architect. All three references stress the architect as a technical leader, with a system wide view, working in or closely alongside the software development team(s) and providing requirements and context to the development team and communicating with those outside the development organization who need to understand the system being built. This aligns very closely with our experience of software architecture.

Finally, the book (Taylor, Medvidovic & Dashofy, 2010) also briefly discusses the role of the software architect (in largely the same terms as the three references above) and also mentions the relationship between software architects and other related roles such as systems engineers and requirements engineers. This discussion appears to relate to the systems engineering domain though, whereas our classification is aimed squarely at enterprise systems development.

MOTIVATION FOR OUR WORK

In our careers as consultants and employees, the authors of this chapter have worked as architects and with architects in tens of organizations across a number of industries including banking, insurance, financial asset management, law enforcement, retail and high technology. In these roles we have frequently needed to cooperate with many other architects and so, to allow productive cooperation, it has been important to understand the focus, scope and competence of each. The problem we have often faced is someone introducing themselves using one of the many of architect titles that are in common use, but this not helping to define the key characteristics of their role.

That said, it's clear that in each organizational context, these various job titles are meaningful and clear, describing the various environment specific nuances of the particular jobs within a particular organization. It's when working across organizations that the job titles cause problems.

When we considered this situation, it seemed to be a fruitless exercise to try to create a complicated and intricate system of architectural job classifications. People wouldn't be able to remember its subtleties and would probably confuse different aspects of it. And in any case, we would never be able to capture all of the details of the variety of architecture jobs that exist in the industry today.

Instead, we realized that what we needed was a clear and very simple classification of architects that divided them into broad groups according to their main concerns and which could be clearly defined using

a couple of straightforward rules that would be common to all organizations. This would be enough to allow the existing jobs to be mapped to these simple classifications but would provide enough precision to allow architects to quickly understand what each other's primary focus would be.

We started off by trying to find an existing standard classification to use; we had no particular desire to create another, and we assumed that this was a common problem. So we were expecting to find that industry organizations would already have defined suitable classifications. However, we found a number of problems when we tried to use the existing classifications that we found in the industry.

- Several of the more comprehensive classifications are proprietary (e.g. IBM's and CAP Gemini's) and so information about them is not freely available;
- The existing classifications don't relate easily to each other (for example it's not clear where IBM's Infrastructure Architecture would fit in CAP Gemini's classifications);
- There do not appear to be clear principles to allow a new architectural job to be mapped to the roles within the classification systems and so it is not clear how to classify responsibilities as belonging to specific roles. For example, Infrastructure and Operations architecture in IBM's classification and Enterprise Architecture and Information Architecture in IASA's classification appear to have a strong relationship between them, if not a direct overlap in the concerns that they address.

These difficulties with the existing sets of classifications led us to consider how we would categorize IT architects and we came to the conclusion that in our experience, the primary concerns of an architect are driven by their scope (single system or organization wide) and their focus (the problem domain or the technology domain). When we considered these two dimensions, we rapidly converged on the three roles we define here and found that all of the architectural roles we were aware of could be mapped cleanly into these three groups and that the groups seem to form very coherent collections of roles with a lot of common interests.

THE ROLE OF THE ARCHITECT

The role of IT architect has grown slowly to take its current position in enterprise computing. The first references to the idea of software architecture emerged at the now famous NATO conference on software engineering in 1969. However the idea of a software or IT architect did not become widespread until the 1990s, when it became more common to hear of people described as "architects," and the first mainstream publications talking about architecture were published, such as the IEEE Software special issue on software architecture in 1995 (IEEE, 1995). Since then, awareness and understanding of the role has accelerated, to the point where today you can find architects in almost any enterprise with a significant IT capability.

However, while today architects appear to be in every organization, and there is some similarity in what they call themselves, there is still widespread confusion over what IT architecture is, what IT architects do (or should do) and what types of IT architect exist. In this article, we hope to answer some of these questions, by clearly defining the different roles that IT architects play, the focus of each, and the activities that they typically perform.

While there are different types of IT architects, there is a core set of characteristics that they all share. We have found these to be as follows:

- *Design Centric Work* - architecture is a design centric activity, although different architects are responsible for designing different types of artifact. However, designing something, be it a system, an infrastructure, a process or a service is core to the activity of being an architect.
- *Stakeholder Focus* – being an architect involves a lot of focus on stakeholders, much more so than more traditional technical and design oriented jobs. This is because a key part of the

architect's role is to bring clarity and consensus to problems that are often poorly understood or poorly communicated by the different stakeholders.

- *A Focus on System-wide Concerns* – rather than focusing on the detailed design of one particular element of a system or problem, architecture inherently involves taking a broad view and focusing on the overall organization, the common design aspects shared across it and the overall qualities of the artifact being designed.
- *Balancing Concerns* – architectural design tends to involve searching for an acceptable solution amongst a range of possible options, which meet the concerns of different stakeholders to a different degree. It is often a case of achieving the least worst option, given stakeholder needs and biases, rather than the optimal engineering solution (which is usually the goal for more detailed or bounded design contexts).
- *Leadership* – most architecture jobs involve some sort of technical leadership, even if this isn't through a traditional line-management relationship. Architects usually need to define standards, norms and direction for their specialist area within their organizations, and ensure that they are adhered to.
- *Managing Uncertainty* – architects are usually involved at the very earliest stages of software development projects, when ideas are hazy, priorities and requirements are ill-defined and the overall goal may not yet be known. Architects need to keep this under control, to encourage creativity and inspiration but also to focus stakeholders on reaching consensus on what needs to be done.

Looking for these characteristics in someone's work helps us to identify that they are performing "architecture" work, even if their job title may not reveal this.

This leads us to naturally consider what we mean by *architecture* in this context. When defining architecture in its most general form, we follow ISO Standard 42010 (ISO, 2007) and define it as the set of design decision which define the essence and core characteristics of the system, or more formally "the fundamental conception of a system in its environment embodied in elements, their relationships to each other and to the environment, and principles guiding system design and evolution".

This definition naturally leads us to require a definition of a *system*, which is the artifact which requires an architecture and which the architect is concerned with designing. Our scope here is information systems development, but different architects design different types of information system and so our definition is somewhat generic. It's quite similar to definitions used by the SEI and others (Ellison & Moore, 2003), and is that a system is the combination of hardware, software and human activity that supports one or more functions within an organization, such as management, operations or decision-making. So the concept of a system can span an organization-wide set of applications that automate an entire-business process through a generic service that provides a facility used by other systems, to an individual small information system that automates one part of a department's work.

So if architects simply design systems, then why do we need to consider different types of architect? The reason for this is that while the idea of creating an architecture is quite simple, creating an effective architecture for a system requires a great deal of specialist domain knowledge and experience. The systems that require architecture work will vary greatly between different environments, and even in a defined domain, such as enterprise information systems, there are significant differences between the architecture required for different systems within the domain. It is also often important to have different types of architect cooperate in order to achieve a successful outcome, so understanding the role of each and the relationships between them is an important factor in achieving this.

A CLASSIFICATION FOR ARCHITECTS

Having held a number of architectural roles ourselves, and observed architects at work in many other organizations, we have found that architects fall into one of three broad groups: enterprise architect, application architect and infrastructure architect. These three groups encompass all of the architectural roles that we have seen in practice.

To help explain the differences between these roles, we classify them along two axes, namely their *business / technology focus* (the extent to which they focus on the problem domain as opposed to the solution domain), and the *breadth of their architectural portfolio* (the number of systems that they have architectural responsibility for).

Some architects focus primarily on the business of the organization that they work for (the problem domain) being experts on one or more aspects of the business that they are helping to automate. Other architects focus primarily on the technology used to solve business problems (the solution domain), and are experts in one or more broad technology areas, such as messaging, networking, user interface development or data storage and retrieval.

Some architects need to find a balance between these two areas, as their role involves understanding some aspect of the problem domain to allow them to automate it, but also understanding the technology of the solution domain well enough to be involved in the detailed design and implementation of the solutions they create.

When considering the breadth dimension, some architects are responsible for a large number of systems (possibly all of the applications in the organization), but at a necessarily shallow level, whereas other architects are responsible for a single application, but at a very deep level of involvement. We also find that the number of systems that an architect is responsible for determines the time horizon and abstraction level that she works at. Architects that work across many systems tend to have a longer term, more abstract, focus than those architects responsible for a small number of systems, who are involved in shorter term single-system change and many more of the details of how a system works and is operated.

We illustrate our architectural classification in the informal diagram in Figure 1.

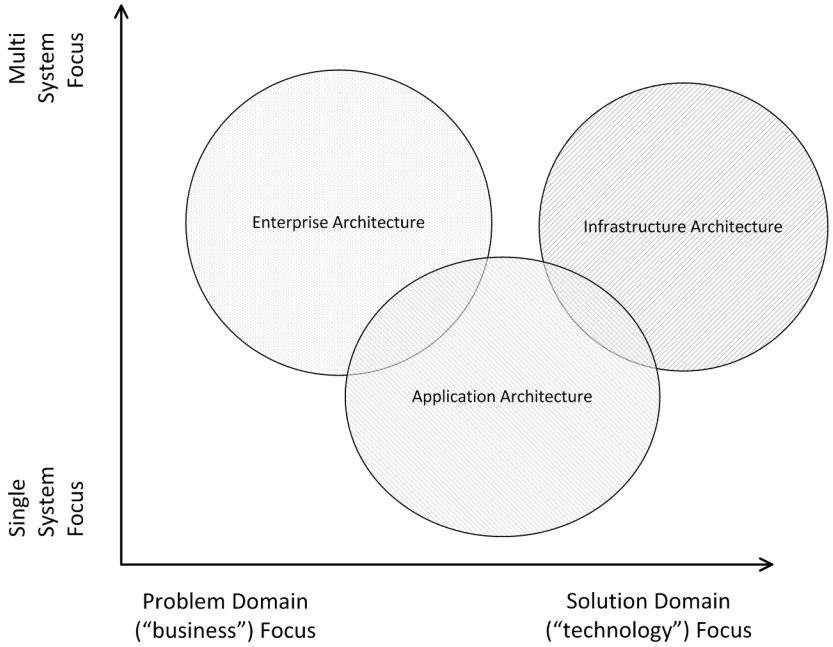

Figure 1 – Classification of Architectural Roles

DEFINITIONS

An *enterprise architect* is responsible for defining the capabilities, functional components, information assets and systems required to support the activities of a business unit or of the whole organization. The knowledge that she needs to perform her job is a blend of domain and technology knowledge, and is often biased towards broad domain knowledge.

Common organizational synonyms for enterprise architect include functional architect, business architect, strategic or strategy architect, domain architect and stream architect.

An *application architect* is someone who focuses on a small number of systems (often only one or two) and is the design authority for those systems, their capabilities and their internal structure. The knowledge she needs is a balanced blend of technology and domain knowledge, combining deep practical knowledge of the technologies needed to build and run her systems, with enough detailed domain knowledge to ensure that her systems effectively meet business needs.

Common synonyms for application architect include software architect, solutions architect, application architect, systems architect and technical architect.

An *infrastructure architect* is someone who, like an enterprise architect, has a scope ranging over many systems, probably all of those in an organization, but where the knowledge she needs to perform her job is nearly all specific technology knowledge, with a limited amount of domain knowledge being required.

Common organizational synonyms for infrastructure architect include technical architect (again), technology architect, and specialist job titles such as database architect, middleware architect, network architect or storage architect.

It is worth noting that our definition of Enterprise Architect also encompasses a couple of commonly used architectural titles, namely Information Architect (or Data Architect) and Business Architect. This is not to say that these are not valid types of architect in some environments, but rather that we have found that they are best classified as sub-specializations of enterprise architecture rather than distinct classifications in their own right (just as storage or network architecture are specializations of infrastructure architecture). We have observed significant overlap between the work performed by different types of enterprise architect but have found that they all share the key characteristics that we attribute to enterprise architecture, namely a domain rather than technical focus and responsibilities that extend their influence over a large number of systems. Hence, we decided to unite information architecture and business architecture with enterprise architecture, rather than trying to separate them.

In the following sections we explore these roles more deeply, by defining the defining characteristics of each role, their responsibilities, and the methods and tools that are used.

ENTERPRISE ARCHITECTS

Defining Characteristics

The enterprise architect has a primary focus on the problem domain (i.e. the business) rather than the solution domain (i.e. the technology) and is responsible for some aspect of the organization's enterprise architecture which, based on the MIT CISR's definition (Weill, 2007), we would define as the set of principles, standards and blueprints for the organization's business processes and IT infrastructure, that reflect the integration and standardization requirements of the company's operating model. As such, enterprise architects create a functional model of the whole enterprise, and map this model onto the physical systems that implement these functions (sometimes referred to as a "system landscape model"), but are not directly involved in the development of application systems or the implementation of the infrastructure platform (which are the domains of the application and infrastructure architects, respectively).

Enterprise architects are primarily interested in what IT needs to provide to the business in order to support its needs and aligning the provision of IT to the direction and needs of the business is a core focus. That said, some enterprise architects are responsible for purely functional aspects of enterprise architecture,

such as the responsibilities and relationships between applications, while others have more technical responsibilities, such as standards, direction or governance over particular types of IT provision (such as data management or enterprise network provision). So, as suggested in Figure 1, even within this role, there is some variation in the problem versus solution domain focus.

The enterprise architect has a cross-system focus, being interested in how all of the systems in the organization combine in order to provide the business with its technology needs. This results in the role being centered around the responsibilities of the systems, their interfaces and how they are integrated, rather than in their internal workings. She often has a role in overseeing major decisions relating to individual systems (an activity usually referred to as "governance"), but rarely gets involved in the details of how they are constructed.

The enterprise architect necessarily has a long-term, quite strategic, view of IT provision in the organization, not being involved in the day-to-day running and improvement of the production systems. Her goal is to guide the IT department in a long-term direction that is congruent with the long term needs of the organization.

The enterprise architect is usually more externally facing than the application architect. She will often work with standards organizations, regulators and other industry bodies to guide or understand industry-wide strategic initiatives and define and agree industry standards and regulation.

In our experience enterprise architects can be drawn from a variety of backgrounds. The need to understand the needs of the organization and have a cross system focus tends to mean that they are former system or infrastructure architects, or sometimes, former business analysts (who are probably responsible for the functional aspects of enterprise architecture, rather than the more technical areas).

Purpose of Role

The purpose of the role of the enterprise architect is to:

- Ensure that the business processes of the organization and well understood and that there is a clear definition of how they are supported by the organization's information systems.
- Ensure that each information system (application) in the organization has a clearly defined set of responsibilities and interfaces.
- Define the structure, location and primary characteristics of the organizations data, both in motion (messaging) and at rest (stored data), including data ownership, rules for data duplication, "golden sources" (owners) of each significant part of the organization's data and so on.
- To ensure that the organization's strategy is understood and is used as an input to IT decision making, so aligning IT and organizational decision making,
- To maximize the return on investment of the IT budget by ensuring that business priorities are understood and are used to select the right areas for IT budget spending.
- To ensure that the organization's future state IT environment is defined and understood and that there is a clear roadmap that defines a credible set of steps to move from the current state to that proposed future state.
- To ensure that IT change projects align with the organization's priorities, strategy and future state architecture.

Responsibilities

The responsibilities of the enterprise architect are varied, but usually include all or some of the following:

- Functional business domain modeling of the organization and the specification of application systems required for it.
- Corporate data modeling, to create enterprise-level data models and data dictionaries, which provide standard definitions of concepts to be shared across the organization.
- Developing organization-wide models of data ownership and strategies for dealing with distributed and decentralized data.
- Creating current and future state models of the enterprise and its systems, and associated gap analysis and roadmap creation to guide the organization towards the desired future state.
- Creation and management of enterprise inventories of key IT assets such as applications, business processes or reusable software elements.
- Contributing to the organization's technology strategy.
- Defining the organization's application integration architecture (the significant information and control flows between key systems, and the data standards to be used) to allow applications to be integrated in a systematic, standardized and flexible way.
- Definition of organizational technical standards.
- Assessing, overseeing and providing governance over IT change, to ensure that it aligns with the organization's standards and strategy.

Methods and Tools

The enterprise architect typically uses methods and tools such as:

- Functional modeling tools, supporting UML or enterprise architecture specific approaches such as Archimate or TOGAF. Due to their ubiquity, desktop office products (such as Word, Visio and PowerPoint) are also often used to create less formal "box and line" diagrams.
- EA frameworks, such as TOGAF or Zachman, to guide the creation of sets of architectural models.
- Specialist EA tool sets such as Troux, ProVision or Promis Netmodeller, which provide enterprise architecture repositories and modeling tools.
- Metadata modeling tools and repositories such as ASG Rochade or Adaptive Metadata Manager.
- IT inventory products such as ARIS IT Inventory or HP Asset Manager, which allow databases of IT assets to be defined, created and managed.

Due to the high cost of some of these tools, enterprise architects often develop their own home-grown equivalents, such as building an IT inventory using a spreadsheet or relational database.

APPLICATION ARCHITECTS

Defining Characteristics

The application architect has a more even balance between problem domain and solution domain focus than her enterprise architect colleagues, as this is crucial to creating effective applications that meet the needs of the business in a cost-effective way.

The application architect tends to focus on a single system, which she is deeply involved in defining, building and often operating. Her main concern is its function and internal design, and in particular the system wide concerns and mechanisms that it embodies. She is knowledgeable about other systems in the organization at a more abstract level, typically understanding them as "black boxes" that hers interacts with.

The time horizon of an application architect is the short to medium term. She knows where her application is headed but is more concerned with how it should be built and changed today.

The application architect is almost always grown from senior, highly capable software developers, so she has very strong technical knowledge of the process of system building.

Purpose of Role

The purpose of the application architect's role is to:

- Ensure that their application(s) have clearly defined scope and key requirements.
- Ensure that their application(s) have a well-defined, clearly implemented and suitable architecture that allows them to meet their objectives and requirements.
- Ensure that their application(s) are capable of meeting their key requirements, particularly the quality properties (non-functional requirements) required of them.
- Create a suitable architectural description for their application(s), according to the complexity of the application(s) and the needs of the environment they are working in.
- Ensure the technical integrity of the implementation their application(s), so that they are implemented in line with the architecture, follow sound software development practice, and are fit for purpose.
- Minimize implementation risk for new applications or application changes, by the use of techniques such as expert judgment, proof-of-concept implementation, technology testing and incremental, risk-driven development.
- Provide a source of expert, independent advice to provide validation and review of other applications in the organization.

Responsibilities

The responsibilities of the application architect include the following:

- Application architectural design, defining the architectural structures of the applications she is responsible for, and creating design models to support this activity.
- Creation of technical standards to be applied across her system in order to ensure commonality of approach and technical integrity.
- Creating prototype and proof-of-concept software in order to validate design ideas, investigate and assess implementation options or guide the work of software implementation teams.
- Detailed software and database design, where the design work is particularly critical to achieving important system qualities (for example perhaps being involved in the design of part of a database that is likely to have a direct effect on system performance).
- Reviewing the work of others, particularly design, software implementation and testing artifacts.
- Performing architectural assessment of her own systems and those of others.

Methods and Tools

The application architect uses the following methods and tools:

- Architecture and design approaches such as Domain Driven Design (Evans, 2003) or Attribute Driven Design (Woicik et al, 2006), the application and development of styles and patterns and the use of architectural viewpoints and perspectives (ISO, 2007).

- Software modeling techniques such as UML, Entity Relationship Modeling, informal notations such as "box and line" diagrams and again, the use of architectural viewpoints. When using formally defined notations such as UML or ERDs, the application architect quite often uses a modeling tool (such as Rational Software Modeler, MagicDraw or PowerDesigner) to help her capture her models.
- Software development and analysis tools when developing or analyzing software (such as IDEs, code analysis tools, testing tools, build and configuration management tools).
- When performing assessments, some application architects will use predefined methods like ATAM (Clements, Kazman & Klein, 2001), whereas others will define their own lightweight approaches like TARA (Woods, 2011) or use an ad hoc approach for the situation in hand.

INFRASTRUCTURE ARCHITECTS

Defining Characteristics

The infrastructure architect tends to have a broad organizational focus, but a narrow technical focus, being an expert in one or more specialist technology areas which she is responsible for right across the organization.

The focus of an infrastructure architect is strongly technical, rather than on the business domain of the organization she works in (indeed, she may well be a consultant to that organization). While she needs to have a broad understanding of the priorities of the business area she is serving, her primary skill is deep expertise in one or more technology domains.

The infrastructure architect necessarily has a cross-system focus, defining technology standards and provision at an organizational (or divisional) level, in order to meet the needs of many applications in a consistent manner. Her relationship with individual application areas is often quite formal, via management mechanisms such as service level agreements (SLAs).

The time horizon of an infrastructure architect is long term, as many of her projects will take a long time to implement, and the return on the investment is likewise only seen over the long term. She is interested in achieving simplicity, standardization, stability and cost effective technology provision, none of which is amenable to a short-term, tactical approach. This often means defining long-term roadmaps for her domain and having long-term relationships with vendors and suppliers, to maximize the value to the organization.

The infrastructure architect often influences the "quality properties" to be exhibited by applications, such as security (designing and implementing enterprise-wide mechanisms for authentication and confidentiality); availability (designing and implementing enterprise-wide mechanisms for high availability or disaster recovery); or scalability (designing and implementing enterprise-wide mechanisms for shared computational infrastructure or on-demand provision of hardware resources).

Purpose of Role

The purpose of the infrastructure architect's role is to:

- Provide a clear definition of the IT infrastructure platform for the organization, and how it is to be used to meet the needs of the organization's applications.
- Define clear standards for infrastructure technology acquisition and use in the organization.
- Select appropriate in-house or third-party technology products to act as the components of the infrastructure platform, and to define how each component is used within the platform.
- Provide in-house expertise in one or more areas of infrastructure technology.
- Act as an informed point of contact with third party infrastructure product vendors and consultants, to ensure that the third parties that the organization contracts with work to meet its needs.

Responsibilities

The responsibilities of the infrastructure architect include the following:

- Infrastructure environment design, to define how shared infrastructure will work and provide a service to the organization.
- Contributing to technology strategy in her area of expertise, defining the current and future state for it and how it relates to the rest of the organization's IT strategy.
- Creating technical standards in areas relevant to her expertise, guiding the organization's adoption and use of particular technologies.
- Reviewing projects that are making use of parts of the infrastructure environment, to guide them towards appropriate and realistic use of the infrastructure and ensuring that particular applications do not cause problems for other users of infrastructure services.
- Project consultancy, to help organizations adopt and make most effective use of the infrastructure services available.
- Liaising with technology vendors who supply the organization's IT infrastructure components, and managing the relationships with important vendors.
- Keeping abreast of new technology developments, understanding how they could benefit the organization, and promoting their adoption where this is of benefit.
- Leading infrastructure implementation projects.

Methods and Tools

The infrastructure architect is not particularly well served by the rather sparse set of methods and tools available to them, particularly in comparison with enterprise architects, but typically uses the following methods and tools:

- Informal modeling tools, such as desktop office products like PowerPoint, Excel and Visio.
- Parts of enterprise architecture frameworks (such as TOGAF and Zachman) that are relevant to the definition of the areas of focus for the infrastructure architect.
- Domain or vendor specific methods or tools such as those available for network design or capacity management and modeling.
- Documentation and communication tools such as wikis, SharePoint sites and PowerPoint presentations.

RELATING ENTERPRISE, APPLICATION AND INFRASTRUCTURE ARCHITECTS

Having defined and explained the three IT architecture roles, we now consider how they relate to each other and the influence that each role has on the other two.

The enterprise architect provides an application architect with a context for her work, and provides an infrastructure architect with strategies, priorities and requirements for enterprise wide infrastructure.

In turn, the application architect provides an enterprise architect with requirements that they have for enterprise infrastructure or standardization, based on the needs of their applications; inputs to the enterprise architecture plans and roadmaps; and validation of the enterprise architecture in practice. The infrastructure architect provides an enterprise architect with assistance in formulating enterprise wide technology plans and specific information about the cost and practicality of the enterprise architecture group's plans.

Application and infrastructure architects also interact frequently, with the application architect usually acting as the customer of the infrastructure architects, who provide standard services or infrastructure configurations, with known characteristics and qualities, which she can use as the foundation for her systems.

These interactions are illustrated by the informal diagram in Figure 2 and explained below.

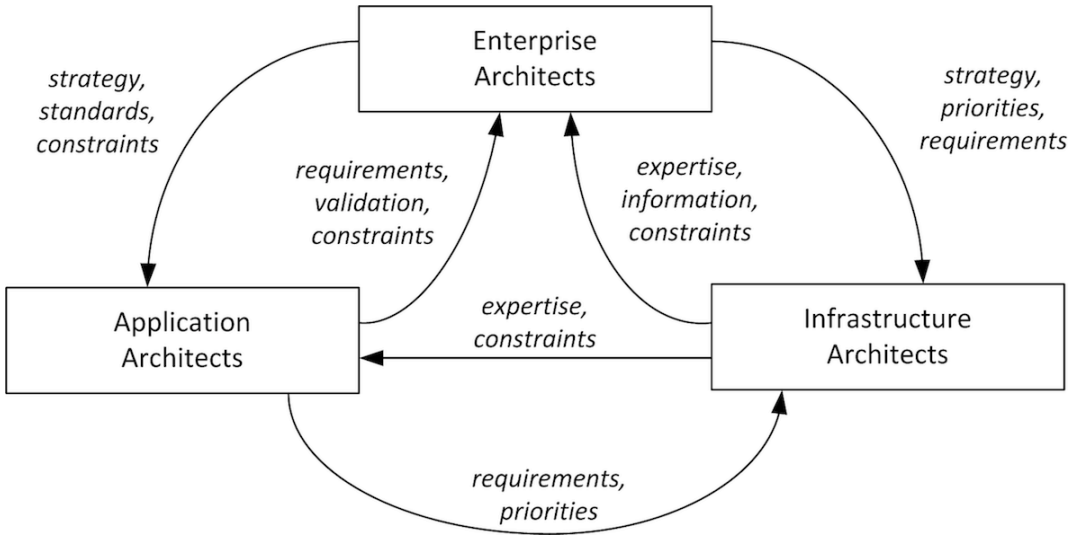

Figure 2 – Inter-Architect Relationships

To be more specific, the enterprise architect provides an application architect with the following artifacts:

- An organization-wide application architecture, to provide context and define responsibilities for her application(s).
- A corporate data model, to guide database and domain model design and application integration.
- An integration architecture and associated standards and patterns, to guide application integration.
- Standards for achieving common critical qualities, such as security, high availability and disaster recovery in a standard way within the organization (often using common organizational services).
- A technology strategy and technical standards, which also guide application design.

In return, the application architect provides an enterprise architect with the following artifacts:

- A practical, usage-based validation of the feasibility and usefulness of the enterprise architecture based on her use of it and the requirements of the applications that she is responsible for.
- An implementation of some parts of the enterprise architecture, as embodied in her applications.
- Content for and validation of corporate standards as they are developed, based on the experience of trying to build applications in the organization.

The enterprise architect provides an infrastructure architect with the following artifacts:

- A technology strategy, into which the infrastructure architect's work will fit.
- A set of business priorities, to help the infrastructure architect focus on and prioritize the right aspects of her work.

- Requirements for shared infrastructure, based on the long term needs of the organization.
- Rollout plans for major programs, to allow the necessary infrastructure services to be designed and implemented.

The infrastructure architect provides an enterprise architect with the following artifacts:

- Evaluations and certifications of products within her area of expertise, to help the enterprise architect confidently select products, knowing that they can be successfully applied within the organization.
- Technical feasibility reviews of enterprise architecture proposals, particularly those relating to shared infrastructure services.
- Cost estimates for the infrastructure aspects of proposed enterprise architecture led programs.
- Information about the vendors who supply the technology in her area of expertise, the relative positioning of different vendors and more general market information such as adoption trends.
- Briefings on emerging technologies to help the enterprise architect stay up to date.

The infrastructure architect can provide an application architect with the following artifacts:

- Application design reviews, from an infrastructure perspective.
- Consultancy or informal training on the various technologies and infrastructure services that the infrastructure architecture is responsible for.
- Design work for the infrastructure aspects of new applications.
- Assistance with selecting appropriate infrastructure products and services for particular applications and providing an understanding of the likely or known risks, problems or limitations that use of particular products or services is likely to imply.
- Cost estimates for the acquisition and use of infrastructure products and services.

An application architect is largely a consumer of the work of the infrastructure architect, but does provide input to infrastructure architecture work via the following means:

- The application architect can supply an infrastructure architect with a deployment view of her system to help the infrastructure architect understand its requirements and plan for it.
- She can supply a rollout schedule to allow the required infrastructure provision to be timed correctly.
- Her non-functional requirements are a useful input to the infrastructure architect's designs, to help identify the right infrastructure services and products to use.
- The budget for the application architect's application is a constraint on the infrastructure provision that can be specified.

DEFINING THE RELATIONSHIPS

In this section, we delve more deeply into the relationships between the three types of IT architect that we described earlier. We do this through the mechanism of a UML class diagram, shown in Figure 3, which models the key architect roles under consideration, the roles of others in the organization that they interact with, and the artifacts through which this interaction occurs.

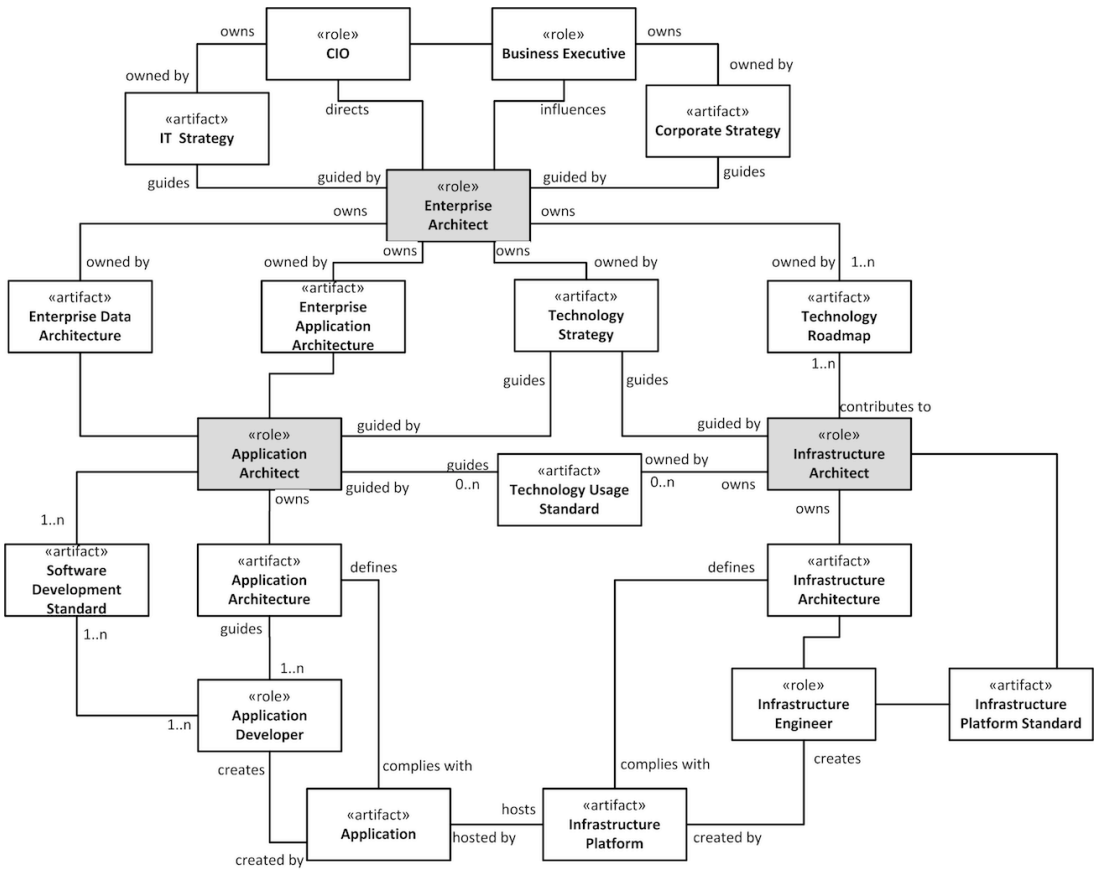

Figure 3 - Inter-Role Relationships

Before describing the content of this model, it is worth briefly clarifying its utility and its limitations. Formalizing the relationships between the roles in this way is useful because it helps to add a degree of precision to what are quite loose relationships, and so helps to clarify the complicated relationships between these three architectural roles. However, it is important to bear in mind that a model of this sort is inevitably a simplification of the complicated interactions between people in real organizations and shows an idealized view of the world, which is unlikely to exist in such a neat and tidy form anywhere in a real organization.

The roles that the actors in the model play are briefly described below:

- *Application Architect* - an architect concerned with the design of one or a small number of application systems.
- *Application Developer* - a software developer who is engaged with the development of one or more application systems.
- *Business Executive* - a role played by any senior non-IT manager in the organization who is interested in shaping its IT environment (typically includes the CEO, COO and CFO).
- *CIO* - the Chief Information Officer, heading the IT organization
- *Enterprise Architect* - an architect concerned with the design of organization-wide (cross-system) IT structures.
- *Infrastructure Architect* - an architect concerned with the design of the IT infrastructure platform within an organization.

- *Infrastructure Engineer* - someone involved in the detailed design and construction of the IT infrastructure platform.

Brief definitions of the artifacts through which the actors in the model communicate are as follows:

- *Application* – a computer system, performing one or more functions for the organization.
- *Application Architecture* - the architecture of a specific application system.
- *Corporate Strategy* - a definition of the strategy for the organization, which defines the businesses, it should be in and why, and how it plans to achieve this.
- *Enterprise Application Architecture* - an architectural definition of the organization-wide application landscape (applications, responsibilities and integration).
- *Enterprise Data Architecture* - an architectural definition of organization-wide data structure, storage and ownership.
- *Infrastructure Architecture* - an architectural definition of the organization's infrastructure platform.
- *Infrastructure Platform* - a collection of hardware and software that provides a runtime environment for the organization's application systems.
- *Infrastructure Platform Standard* - the definition of a standard approach to using one or more specific technologies within the organization's infrastructure platform.
- *IT Strategy* - the definition of the organization's objectives, principles and tactics that relate to how information technology will support the organization's business strategy.
- *Software Development Standard* - the definition of a standard approach to some aspect of software development.
- *Technology Roadmap* - the definition of the sequence of steps and their planned timing to evolve the technology in use in the organization towards a planned future state.
- *Technology Strategy* - the definition of the organization's objectives, principles and tactics relating to the specific technologies that the organization uses
- *Technology Usage Standard* - the definition of a standard approach to using one or more specific pieces of technology within the organization.

Having defined the artifacts and roles within the model, we can now consider how they relate to each other and what this tells us about the relationships between enterprise, application and infrastructure architects.

- The Chief Information Officer, who runs the IT organization, directs the work of the *enterprise architects* and their work is also influenced by a number of the organization's executive level business managers. The formal route for the CIO to direct the work of enterprise architects is through the *IT Strategy* that the enterprise architecture must support. Similarly the formal communication path between the business managers and the work of enterprise architects is through the definition of the *Business Strategy* that informs the enterprise architecture work.
- An *enterprise architect* is the owner of a number of important strategic IT artifacts, namely the *Technology Strategy*, the enterprise's *Data Architecture* and *Application Architecture* and the

Technology Roadmap that explains how the IT environment (in particular the infrastructure platform) is going to evolve in the future.

- An *application architect* uses a number of the enterprise architect's outputs in order to inform their work, in particular the enterprise's *Application Architecture* and *Data Architecture* and the *Technology Strategy* that informs their choice and use of technology when designing systems.
- The *application architect*'s outputs are the *Application Architecture* for one or more application systems, and the *Software Development Standards* which define the standard approaches for software development that should be employed when developing or maintaining certain applications. These artifacts are used by the *Application Developers* who are responsible for the relevant application systems.
- Similarly, the *infrastructure architects* use a number of the enterprise architect's outputs when designing the organization's infrastructure platform, in particular the *Technology Strategy* and the *Technology Roadmap* (which they will also need to contribute to in their own area of specialty, to ensure that the plans that the roadmap implies are achievable).
- The *infrastructure architect's* outputs are the *Infrastructure Architecture* (or their part of it) and a set of *Infrastructure Platform Standards* that define how particular technologies should be used within the organization's infrastructure platform. These artifacts are used by *Infrastructure Engineers* who are responsible for the detailed design and construction of the organization's infrastructure platform.
- The *infrastructure architect* also influences the work of the *application architect* through the use of *Technology Usage Standards* that define how particular technologies should be used within the organization, so providing the application architect with guidance and constraints on the technologies they should use within their applications and how they should be used.

CASE STUDY

A retailer of consumer electronics goods and accessories sells its products in stores and over the Internet, and now wants to move into the mobile consumer market. However, the retailer's strategy of encouraging customers to visit their shopping website from their mobile devices is not working well, since the website is poorly suited to the cellphone form factor, and offers a painfully slow user experience over a mobile connection.

The retailer has therefore decided to develop an "app" which will run on a range of Internet-enabled smartphones and portable devices and provide similar functionality to that available on the website. The app will seamlessly align with the website from the customer's perspective – so, for example, the customer will log in to the app using the same credentials, and will be able to see their favorite items, shopping history and account details. The first version of the app will allow customers to browse part of the retailer's catalog, order goods and track their delivery status. Subsequent versions will extend the catalog and offer more features.

The app will rely on a number of supporting servers that process requests on its behalf and the overall application will be required to integrate with the retailer's existing systems for supply chain management, customer management and finance. It will also be required to integrate with the retailer's web infrastructure in a secure, scalable way.

The application architect plays a key role in defining the architecture of the new application and overseeing its implementation. However, the other types of architect also make important contributions to the success of the project, as the case study illustrates.

The Enterprise Architect

As we have seen the enterprise architect focuses on the problem domain and owns the "big picture." For the shopping app, her most important contribution is to provide an enterprise functional model. This model defines the retailer's important enterprise functions, such as manufacturing, distribution and warehouse management, and the major information and control flows between them. She discusses this model with the application architect, so that they can agree where the new application "fits" into the model, and which enterprise functions it will have to interact with. The application architect uses this information to produce a first-cut definition of the scope and requirements of the app.

The enterprise architect also maintains a system landscape model which maps the enterprise functional model on to physical systems. This mapping is sometimes one-to-one (for example, the retailer has a single financial accounting system), but in many cases it is more complex – the biggest challenge being that the distribution function is implemented in a number of different systems according to region and product. The enterprise architect helps the application architect understand these systems, what functions they provide, and how the app and its servers should interact with them. The application architect uses this information to produce a more detailed specification of the app's interfaces and control flows.

Another important contribution of enterprise architecture is the definition and application of enterprise messaging standards. The enterprise architecture team has defined a standard set of messages and data items, such as order, delivery note and invoice, and all systems are expected to conform to these definitions in their interfaces. The enterprise architect works with the application architect to understand which messages should be used for which interfaces, and how these messages should be used.

Infrastructure Architect

It is always very difficult to predict the take-up of an Internet application, and for this reason the application architect is keen that the back-end infrastructure is easily scalable to deal with unexpected peaks of customer demand. The app must also comply with the retailer's technology standards for servers, storage, networking, security, and middleware.

The retailer runs several data centers each with dedicated web and application server farms which support their main website. Infrastructure components such as servers, storage, or network connectivity are usually provisioned in the form of standard services with defined characteristics (such as a server's CPU and memory configuration) and standard SLAs for deployment, management and performance.

The infrastructure architect works with the application architect to understand how this infrastructure can be used to host the app's servers and the existing enterprise services that it relies upon. He also helps the application architect define the required storage management architecture, and how interactions between the app and the data center will traverse the retailer's network and security infrastructure (firewalls, load balancers, session endpoints and so forth).

The infrastructure architect then helps the application architect define the infrastructure for the app's middle-tier components: the application servers, relational database, messaging-oriented middleware, and file transfer services. Finally, the infrastructure architect helps the application architect define the business continuity architecture which will enable the service to come back on-line quickly in the event of a severe failure.

The biggest challenge to the two architects is to understand how the app will be deployed to customers' smartphones. This is an area of which the retailer has no previous experience, so the infrastructure architect brings in an outside expert to advise on the best way forward. The solution requires some changes to the retailer's data networking and security capabilities, and the infrastructure architect oversees the implementation of these by the retailer's production engineering teams.

Application Architect

Involving the enterprise and infrastructure architect has made the application architect's life much easier. He is able to draw on their expertise and knowledge to help design aspects of the architecture – its interactions with other systems, and the infrastructure it will run on – of which he has little knowledge himself.

The communication is not just one-way, however: the enterprise architect has made a number of clarifications and improvements to the enterprise model, and has disseminated to other teams the insights she has gained. The infrastructure architect has been able to add a service for the deployment of software to customer devices to the retailer's portfolio of standard infrastructure services. This will benefit the next project that needs to write a smartphone app (there are already a couple in the pipeline).

Conversely, the application architect has been able to dedicate much more of his time to the architecture of the application itself. He is able to represent external systems and underlying infrastructure as "black boxes" in most of his models, confident that the nature and characteristics of these are correct and well understood.

CONCLUSION

In conclusion, the role of the IT architect has become well established and can be seen in action in many organizations, although we find a great diversity in job titles and descriptions. This leads to confusion about the role amongst stakeholders and the architects themselves.

By classifying architects along two axes, namely the scope and focus of their jobs, the classification problem is greatly simplified. This analysis led us to define a simple, widely-applicable taxonomy of architects, namely *enterprise architects*, *application architects* and *infrastructure architects*. With this insight we have been able to clearly define the roles of each and better understand the interactions between them.

The enterprise architect sets the context for system and infrastructure architects. She provides the application architect with a context for her work, and the infrastructure architect with strategies, priorities and requirements for enterprise wide infrastructure. Application architects are responsible for the architecture and design of individual systems, within an overall enterprise application architecture, using technology services designed by infrastructure architects. Infrastructure architects provide cross system infrastructure services (storage, compute, network), within a set of priorities and direction set by enterprise architects.

REFERENCES

Akenine, D. (2008). A Study of Architect Roles by IASA Sweden. *Architecture Journal*, 15, April 2008. Retrieved September 1, 2011, from http://msdn.microsoft.com/en-us/architecture/cc505968.

Clements, P., Kazman R. & Klein, M. (2001). *Evaluating Software Architectures: Methods and Case Studies*. Boston, MA: Addison-Wesley.

Ellison, R.J. & Moore, A.P. (2003). *Trustworthy Refinement Through Intrusion-Aware Design (TRIAD)*. Technical Report, CMU/SEI-2003-TR-002, March 2003.

Evans, E. (2003). *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Boston, MA: Addison-Wesley.

Fowler, M. (2003), Who Needs an Architect? *IEEE Software*, 20(5), 11-13.

IASA (2011). *Architecture Roles.* Retrieved 28 September 2011 from http://www.iasaglobal.org/iasa/Roles.asp

IEEE (1995). *Software* 12 (6).

ISO (2007). *ISO/IEC 42010:2007: Systems and software engineering -- Recommended practice for architectural description of software-intensive systems*.

Kruchten P. (2008). *What do software architects really do?* Journal of Systems & Software, vol. 81, pp. 2413-2416.

Taylor, R., Medvidovic, N. & Dashofy, E. (2010). *Software Architecture: Foundations, Theory and Practice*. Hoboken, NJ: John Wiley and Sons.

Vickers, B. (2004). Architecting a software architect. In *Aerospace Conference 2004*, Big Sky, MT, USA, 6-13 March 2004, Proceedings. Piscataway, NJ: IEEE.

Weill, P. (2007). Innovating with Information Systems. Presented at *6th e-Business Conference*, Barcelona, Spain, 27th March 2007. Retrieved January 5, 2012 from http://www.iese.edu/en/files/6_29338.pdf

Woicik R. et al. (2006). *Attribute Driven Design*, Technical Report, CMU/SEI-2006-TR-023, November 2006.

Woods, E. (2011, June). Industrial Architectural Assessment Using TARA. In *9th Working IEEE Conference on Software Architecture, WICSA 2011*, Colorado, USA, 20–24 June 2011, Proceedings. Piscataway, NJ: IEEE Computer Society Press, 2010.

van't Wout, J., Waage, M., Hartman, H., Stahlecker & M., Hofman, A. (2010). *The Integrated Architecture Framework Explained: Why, What, How*. Springer,

Yi Qun, P. (2009). *IBM's ITA & ITS Profession*. Retrieved September 1, 2011, from http://www.kingdee.com/news/subject/09togaf/pdf/wuzhifan.pdf

KEY TERMS & DEFINITIONS

IT Architect: Someone who is responsible, in whole or in part, for the conception and definition of the architecture of some part of an organization's information system environment and whose work can be characterized by performing design work, a focus on stakeholders and their needs, the responsibility for system-wide concerns, the need to balance conflicting concerns, leadership and the need to manage uncertainty.

Architecture: The set of design decision which define the essence and core characteristics of the system, or more formally, following ISO 4210, "the fundamental conception of a system in its environment embodied in elements, their relationships to each other and to the environment, and principles guiding system design and evolution".

System: In the information systems domain, a system is the combination of hardware, software and human activity that supports one or more functions within an organization, such as management, operations or decision-making. This concept of a system can span an organization-wide set of applications that automate an entire-business process through a generic service that provides a facility used by other systems, to an individual small information system that automates one part of a department's work.

Enterprise Architecture: The set of principles, standards and blueprints for the organization's business processes and IT infrastructure that reflect the integration and standardization requirements of the company's operating model.

Enterprise Architect: Someone who is responsible for defining the capabilities, functional components, information assets and systems required to support the activities of a business unit or of the whole organization, requiring a blend of domain and technology knowledge, and is often biased towards broad domain knowledge. Common organizational synonyms for enterprise architect include functional architect, business architect, strategic or strategy architect, domain architect, data architect, information architect and (business) stream architect

Application Architecture: The set of key design decisions, structures and organizing principles for a software application system that define the characteristics and capabilities of the application.

Application Architect: Someone who is responsible for the design, capabilities and internal structures (i.e. the architecture) of a specific set of systems, requiring a balanced blend of technology and domain knowledge, combining deep practical knowledge of the technologies needed to build and run systems, with enough detailed domain knowledge to ensure that the systems effectively meet business needs.
Common synonyms for application architect include software architect, solutions architect, application architect, systems architect and technical architect

Infrastructure Architecture: The definition of the key design characteristics of part or all of a collection of compute, storage and networking components that combine to form a unified platform on which applications can be hosted.

Infrastructure Architect: Someone who has responsibility for the architecture of an infrastructure platform used to host many systems, probably all of those in an organization, requiring primarily specific technology knowledge, with a limited amount of domain knowledge being required. Common organizational synonyms for infrastructure architect include technical architect, technology architect, and specialist job titles such as database architect, middleware architect, network architect or storage architect.