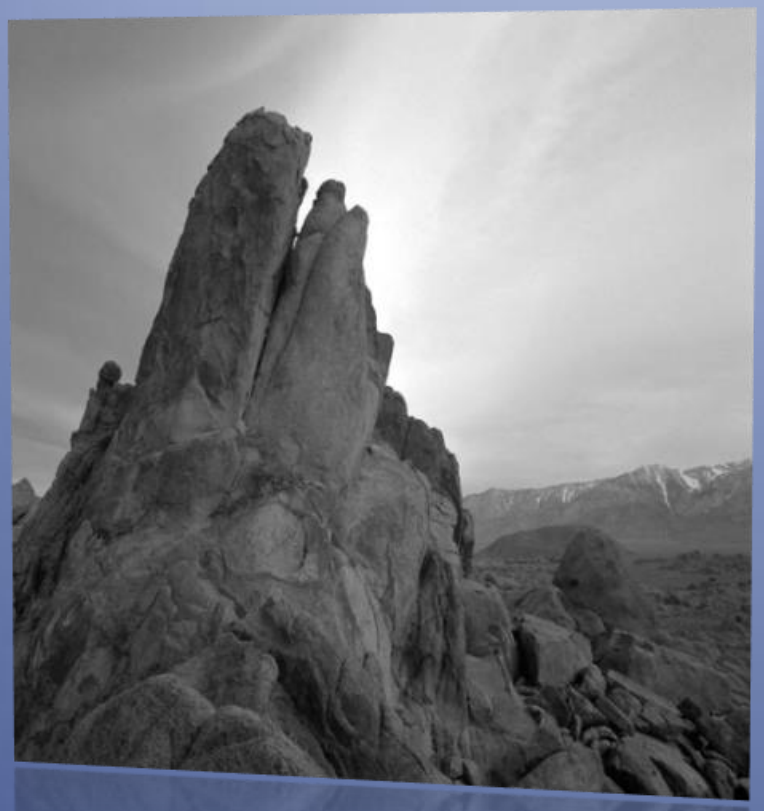


IASA GLOBAL - IT ARCHITECTURE CAPABILITIES

ITABOK – IT ARCHITECTURE BODY OF KNOWLEDGE

CONTENTS

Who Is Iasa Global?

Iasa is the worldwide community of IT architects. A premier association focused on the architecture profession through the advancement of best practices and education while delivering programs and services to IT architects of all levels around the world.

Established in 2002, Iasa is an international non-profit business association dedicated to the advancement and sharing of issues related to all IT architecture specializations in the enterprise, product, education, healthcare and government sectors.

The association is committed to improving the quality of the IT architecture industry by developing and delivering standards, education programs and developing accreditation programs and services that optimize the development of architecture profession. Iasa has the reach to approximately 80,000 architects located in over 50 countries.

Iasa will rely on the following objectives in order to obtain our goals:

Education

Knowledge

Advocacy

Ethics

Values

ABOUT THE BOOK

This book provides an awareness-level review of the broad set of skills every IT architect needs in order to be successful as a practicing architect. These skill shave been identified by 5 main pillars of architecture; Business Technology Strategy, Design, IT Environment, Quality Attributes and Human Dynamics.

Regardless of what title you hold, whether you work in a consulting organization or are a staff architect inside of an organization, these are skills critical to the architect in support the business and the IT organization, and making sure the use of IT provides the highest value to the organization.

In your current position you may not leverage all of the skills we describe, but during the yearly cycle of business prioritization, funding of projects, delivery of projects, and operation of the solutions over its life, someone that is filling the role of architect is leveraging these skills. As you progress your career you will need these skills, and you will find if you start using them you will accelerate your career trajectory.

The process used to identify the skills is an industry standard process for job task analysis. First, we invited successful practicing architects from around the world to participate in several focus groups to create a description of the skill set. Once we had consensus on a well-defined set of skills, we sent a survey to the broader community of architects to gather statistically relevant data on skills ranking and relevance.

The analysis took into account existing definitions available in the space, data from root cause analysis on failed projects, and set a future state of predictable, repeatable success in the practice of architecture, and created a roadmap to move the profession forward.

This is not focused on specific processes, frameworks, technology stacks, and tools that architects use, rather on the broader knowledge and skills the architect needs.

Our goal in all of this is to take our current state which is an environment where were all find path and approach to learning about and practicing architecture, and where most projects are failures because they don't come in on time or on budget. Even worse is when they don't solve the business problems they are started for. We believe we can correct many of these challenges through common skills, tools, and training and that is why we are creating the body of knowledge.

CHAPTER 1: WHAT IS ARCHITECTURE?

1.1 - OVERVIEW

We've all heard that with three architects in a room, there will be seven opinions offered. The same can be said for describing what an architect and architecture are.

Is there confusion over what architecture is? No, not really. There are just different aspects of what architects do based on the role they play in their organization. The good news is these differing descriptions align well.

If you look at the definition of architect in any dictionary, you'll see the definition of a building architect highlighting a blend of art, science, and practice. There are now parallel descriptions for IT architecture that describes the art, science, and practice of organizing and integrating computer systems. From MIT, the description offered is more knowing the business process and reflecting a solution with IT. From SEI, the description offered sounds more like the definition found in dictionaries, focusing on the technical aspects of designing or engineering a complex solution.

Various definitions Include:

1 : the art or science of building; *specifically* : the art or practice of designing and building structures and especially habitable ones
2 a : formation or construction resulting from or as if from a conscious act <the architecture of the garden> **b** : a unifying or coherent form or structure <the novel lacks architecture>
3 : architectural product or work
4 : a method or style of building
5 : the manner in which the components of a computer or computer system are organized and integrated

* http://www.merriam-webster.com/dictionary/Architecture

Enterprise architecture is the organizing logic for business processes and IT infrastructure reflecting the integration and standardization requirements of the firm's operating model. *MIT Center for Information Systems Research, Peter Weill, Director, as presented at the Sixth e-Business Conference, Barcelona Spain, 27 March 2007*

The **software architecture** of a program or computing system is the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them. *Bass, Clements, and Kazman. Software Architecture in Practice 2nd edition, Addison-Wesley 2003*

Building Architecture is known as the art and science of designing and erecting buildings.

Medicine is known as the art or science of treating disease with drugs or curative substances, as distinguished from surgery and obstetrics.

IT Architecture is known as the art or science of designing and delivering valuable technology strategies.

The IASA community feels that both sides of those definitions, when blended are a good definition. We feel that *IT architecture is the art and science of designing and delivering valuable technology strategy.*

We must be able to describe a strategy for solving business problem with IT. By having business skills, we speak the language of business to executive sponsors to understand what drives the organization, what challenges they face, and what others in their domain are doing and what challenges they face.

We must use our broad technical skills to evaluate the existing IT environment, the staff and skills they have, how they operate and at what operational maturity level. We must use the language of architecture to speak with our peers, understand and communicate effectively so we can make sure various teams, projects, and solutions all integrate successfully into an existing operational environment.

We also must use past deep technical expertise to quickly explore and learn new technologies we may not have experience with. We must be able to speak the language of technology to subject matter experts in order to have relevant conversations with them and have any credibility when speaking with them. If we don't we risk them taking a path other than the one we suggest, and that could have a negative impact on the value the solution provides.

Iasa's vision is to turn IT Architecture into a profession, which is a group of people pursuing a learned art with a common calling. There is some controversy around what constitutes a profession, but most agree it is educable, essential, has a large common body of knowledge,

has specializations, can be practiced for an entire career, and often has certifications and a code of ethics.

Twenty years ago, you were a project manager because you could manage projects, and you could manage projects because you had managed projects. Does that sound like your experience as an architect today?

There was no formal training or common set of tools, languages, or process for project management. Now, there are tools, training, common language, common tools, and various but related approaches to managing projects.

1.2 - WHAT IS AN IT ARCHITECT?

Put three architects in a room and they can discuss why they are an architect and the other two are not until their throats are sore. Typically, they possess the same core set of skills and values, but each, based on their organizational structure and the type of company the work for, will have a different archetype that resonates with them.

IASA research suggests that most architect fall into one or two of these architect archetypes. All have similar foundational skills, but each has a different perspective based on their architectural practice. Do you see one or two that resonate with you?

From Iasa's perspective, an architect is the technology strategist for the business.

This requires they have a broad understanding of technology that spans programming and development of custom applications, the infrastructure environment that the solution must reside in, and the operational environment that provides support.

1.3 - POSITIONING THE ARCHITECT SPECIALIZATION INSIDE THE ORGANIZATION

Just as there are different archetypes for architects, each of us has a unique background. Some start as a developer and grow business skills, and possibly infrastructure skills. Some come from an infrastructure or operations background and grow business or developer skills. While others come from a business background and grow technical competence out of passion or need.

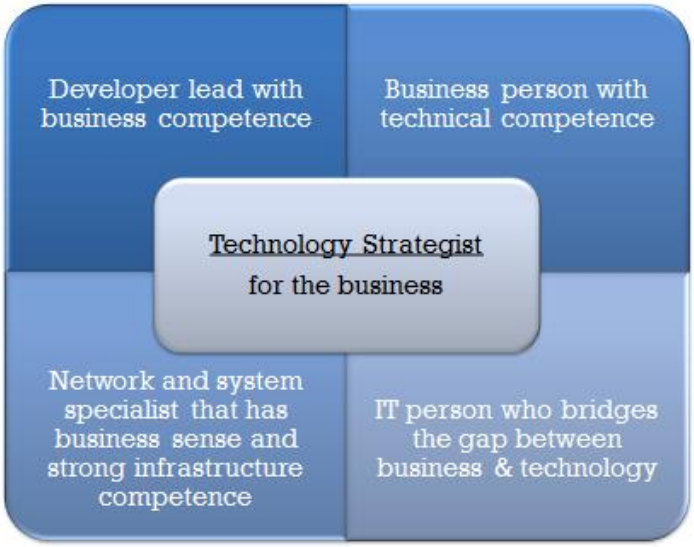


Figure 1.3.1

As architects we face many challenges that go beyond the current projects we may be engaged in, from a lack of understanding of what we do, what value IT does or can bring to an organization, ill-defined structure that has grown over time, and not having true authority to use as a lever.

1.4 - IT ARCHITECTS CHALLENGES:

- Lack of understanding of the architect specialization and no direct authority as an architect
- Existing misalignments between technology and "the business" seen as insurmountable
- Overcoming ad-hoc infrastructures

As an architect, your value and role may not be well defined. Additionally, architects tend to be individual contributors, have no direct authority, and are many times seen as interlopers - dropping into projects for a brief time to provide no value and put up roadblocks to delivering a solution.

Based on your role as the organization's technology strategist, you can work to socialize these value statements. Whether you are a staff architect or a consulting architect, you are part owner of the perception of architecture inside of an organization and on

the projects you are involved in. As you practice and grow your skills, strive to create a perception of architecture that highlights the value an architects and strong architectural practice can bring to the organization.

The architect community doesn't agree on what architecture is and most architects could not describe the value they provide during an elevator ride with an executive sponsor. There is not a widely-held understanding of what value architects provide and when they are needed.

Many organizations feel there is a chasm between the goals and needs of the business and the goals and needs of the IT department. In many cases business leaders feel IT is a cost center or a necessary evil required to operate as a business.

In some organizations budgets and authority over IT are centralized, while in others the budgets and authority are splintered across business units. This can drive confusion and internal "turf wars".

The architect community doesn't agree on what architecture is and most architects could not describe the value they provide during an elevator ride with an executive sponsor. There is not a widely-held understanding of what value architects provide and when they are needed.

Many organizations feel there is a chasm between the goals and needs of the business and the goals and needs of the IT department. In many cases business leaders feel IT is a cost center or a necessary evil required to do operate.

We work in environments that have grown over time and are poorly documented and not well conceived. There are many isolated technical projects going on and delivered, but do not reflect the existing environment and are created in silos. These are operated together and become the infrastructure.

Additionally, architects are typically individual contributors, have no direct authority, and are many times seen as interlopers, dropping into projects for a brief time to provide no value and put up roadblocks to delivering a solution.

In summary, the IT architect profession is in its infancy. Various groups have various descriptions for architects and architecture, and use different terms and titles interchangeably. Based on your own work environment and background, you will have your own perspective on what skills are essential and what the role of an architect and value of architecture are.

Through collaborative work with architect across the globe, the IASA community defines IT architecture as the art or science of designing and delivering valuable technology strategies. We also define the IT architect as the technology strategist for the business. A goal we have is to formalize the IT architect profession, creating a common body of knowledge and common language.

1.5 - HOW DO YOU DESCRIBE VALUE?

How do you describe value, and why do you describe value? As an architect, at times we feel as those our job is to ensure there is alignment between business need and an IT solution, and then to deliver a solution. The value of a solution should be obvious to everyone and really doesn't matter because the project was funded, delivered successfully, and is operational.

A few fellows at Gartner recently published a book suggesting that most CIOs are not successful in their jobs. When they speak about IT to the other members of the C suite, they talk about uptime, data processed, and uptime, while the other members talk about their progress in terms of delivery against the year's key objectives. The analogy used describes having a piece of exercise equipment. The CIO would describe the equipment based on availability and usage while the other members would talk about the impact on health, energy, and looks. The key point is that CIOs need to represent IT in business metric terms and business value delivered.

We, as architects are the ground troops for the CIO and are the ones that can capture data on the business value provided through IT. However, to do that we must learn how to describe the value provided. That sounds easy, but can be quite challenging.

Setting a strategy for IT that describes shared concerns, such as operation and infrastructure, and describing an approach for structuring stacks doesn't directly generate additional revenue or reduce costs, but can be shown to provide business value in other ways. What ways could you describe the value of putting a structure like this in place in your organization?

By providing strong alignment with an organization's mission and critical business goals we can have an impact which can be described in a number of ways. For a consolidation project it may be in retiring systems and reducing operating costs. If the solution is introducing a system that provide greater agility, the business impact of delivering solutions 30% quicker provides competitive advantage. If the solution allows help desk people to close trouble call tickets 40% quicker, there is a direct impact to the number of additional personnel needed to support a new product line and that translates to lower overall cost of doing business.

Business could be a for-profit entity, or it could be a public sector organization, such as a government body. Whether you are a staff architect or working for a consulting organization, the executive sponsors enjoys hearing how you are helping them achieve their goals, and their goals are not technology-focused goals. To become a trusted advisor you must show a track record of success on projects, but you must also be able to translate that success to the overall success your projects afford the organization.

As architects we have visibility across the entire IT landscape and are best positioned to identify opportunities that will provide the greatest value to the organization. We can look at projects that could reduce costs and shift budget required to operate existing systems to delivering new capability. We are the innovators for an organization.

Future-proofing solutions and providing a more agile environment helps companies scale beyond what they plan for, increase competitive advantage, or quickly catch up with their competitors. As an architect you should only build what is needed, but you should also design for ease of update as capabilities are needed.

As we design a solution we calculate that we will make the end user 20% more efficient by automating the process they use to do their job. We use that to help sell the solution, but if we provision the solution so those metrics can be captured over time, we can provide evidence of the positive impact we have had for the organization. If we provide that data back to the CIO, we provide the tools needed to justify new projects in the future and increase the perceived value of architecture and the architect team.

In summary, if we can identify key metrics of business success we have enabled, we are successful and the organization reaps reward. If we do that and consolidate those success stories to provide that evidence to the CIO, then we too gain benefit in our relationships with stakeholders and funding for projects we initiate.

To be successful in describing business value, we must understand the business drivers and be able to think outside of just new capability and reduced cost to describe value. Consider taking a project you have or are delivering and looking at the ways of describing value in this lesson. Can you find additional metrics for describing success and value?

CHAPTER 2: BUSINESS TECHNOILOGY STRATEGY PILLAR

2.1 -OVERVIEW

As most people experienced with technology enter the role of architect, they find that technical prowess is not enough, and one of the most surprising skills they learn is needed is business acumen. Many technologist say “tell me what you want and I’ll build it”.

In the role of an architect you must understand the business drivers for the organization, understand the existing IT environment, and provide a strategy for the use of IT that best maps to the environment. Your value statement is being able to speak the language of business, the language of architecture, and the language of technology. Your role is the only role in the list of stakeholders that has visibility across the organization and the skills in business and technology to provide cohesion between the business goals and IT use. It is your value statement!

As representatives of the CIO and the IT department, you must show business managers and executive sponsors that you understand the business drivers and business challenges they are faced with, and then describe how IT can help them achieve their goals. Remember, most clichés have some truth to them so consider what biases and challenges they may drive.

Go into an organization and most people know there is a go-to-market strategy, sales strategy, partner strategy, or financial strategy. If the organization is a government entity, there’s a strategy for providing constituent value, and managing cash flow and budgets. Obvious that marketing owns the marketing strategy and sales owns the sales strategy. Additionally, there is a strategy for managing and reporting budgets and cash flow. But do you ever hear an IT strategy mentioned?

The skills represented in Business Technology Strategy are about create a strategy for the use of IT that maps to and support the goals of the business.

What Strategies Does Your Business Already Use?

- Marketing strategy
- Sales strategy
- Product strategy
- Finance strategy
- Customer strategy
- Partner strategy
- Who owns the Technology Strategy?

The strategy we create should not be something we do for management because we have to, nor should it be seen as something that comes from an "Ivory Tower" group of people in the IT department. The architecture we create is the strategy and we should be able to describe it in a way that is relevant to all of our stakeholders. We should create it without being asked.

2.2 - BUSINESS FUNDAMENTALS

Business fundamentals is having a broad understanding of generic business structures and functions, including legal and operating structures, operating models, and basic financial statements. Architects should be able to demonstrate competence in the organizational and financial aspects of a business including; basic financial statement analysis and various ways to fund and finance project work.

If you don't understand business how can you participate? How could you better understand your business? For example, was McDonalds in the food business?

No. They were in the real-estate business.

Cost of Not Knowing

- Lack of credibility
- Misplaced strategy

Skills Analysis

- Can you describe the makeup and structure of a business?
- Would you know how to work with non-IT stakeholders?
- Have you created strategies that influence a line of business or the entire business?
- Do you coach an IT staff in business trends and changes?

Concepts

- Business fundamentals are the basic elements of business studied in business schools and MBA programs.
- Architects are required to understand and study how businesses work:
    - To ensure that their technology strategy actually supports their current business environment
    - To effectively engage with other business owners
- The level that an architect studies and grows their business understanding is related to their specialization

Specialization Knowledge Areas within business fundamentals are:

Software Architect:	Define/manage business impact of software.
Infrastructure Architect:	Define/manage business impact of data center.
Business Architect:	Provide technology innovation for business objectives.
Information Architect:	Provide information innovation for business objectives.

REFLECTION POINTS:

Read through this list and consider how your understanding of business fundamentals can be used to enhance your architecture practice.

- How much should an architect know about marketing, finance, operations, sales, or other? Use examples.
- Discuss the pros/cons of this statement, "Business architects do not architect the business, they provide the technology strategy underpinnings to any successful business capability."
- When is the last time you used business fundamentals in your specialization?
- How "up to date" should architects be in the business environment?

2.3 - WHAT STRATEGY IS

The official definition of strategy is:

strat·e·gy

- skillful use of a stratagem
- a plan, method, or series of maneuvers or stratagems for obtaining a specific goal or result

Strategy is simply; any plan that makes you money, beats the competition, makes life better, and is architecture.

A good strategy should be written like a good review goal. Ever heard of SMART goals? They are specific, measurable, attainable, realistic, and timely or time constrained. As you create your architecture strategy, consider using SMART goals as your template. As you create the strategy you must be able to describe the value it provides to your constituents.

What if we want to offer constituent value? For example, IASA might define a business goal as" improve member value by 10% in the fiscal year". An executive team at the Association may suggest a solution offering would help to accomplish such a goal. They may estimate that success of that goal through quarterly surveys and other techniques but would the rest of the organization gage that success?

- Marketing may focus on making the solution prominent.
- Sales may train their sales staff to better answer member questions.
- The community may focus on developing a knowledge base for member interviews.
- Finance may set the limit of the cost to $50 K.

What again is a technology strategy for good architecture? Some solutions are not too challenging to create and are easy to create architectural description for, but we still need to describe constituent value for the environment the solution will enter. The following diagram describes the potential solution the architect is working with.

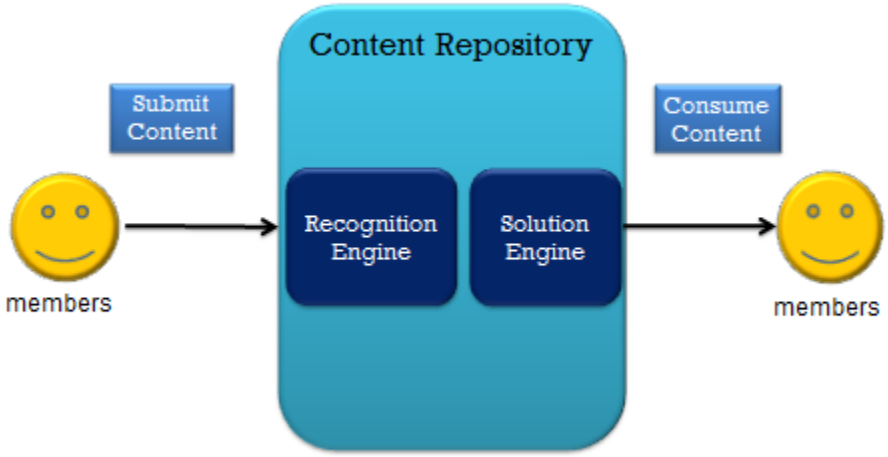


Figure 2

The architect describes a scenario where members submit content to the content repository and other members are able to consume that content to help solve problems. Inside of the repository are solution engine sand a recognition engines that have to be built to match potential problems to submitted solutions.

The technical solution for this is somewhat simple, but how do you measure value against cost? That is the architecture?

This is where most architects stop in the design and delivery of solution architecture. The problem with this approach, of course is that it does not describe how we are going to measure the success of this architecture. How would you define a way of evaluating the success of this content repository to our members? How much is the cost to deliver and the cost to maintain, and is the program worth funding?

Here is a simple example of how you might calculate value.

CBA Element	Cost	Benefit
Program cost	500,000	0
Benefit	**Current**	**Target**
Member traffic growth	50,000/day	75,000/day
Time on Web site	15 min	19 min
Satisfaction overall (survey)	Good	Superior
Solution identification (survey)	Estimated 20 min if at all	7 min
Problem resolution timeframe (survey and site – see wireframes)	Estimated 3 trips (45 min if at all)	25 min

Figure 3

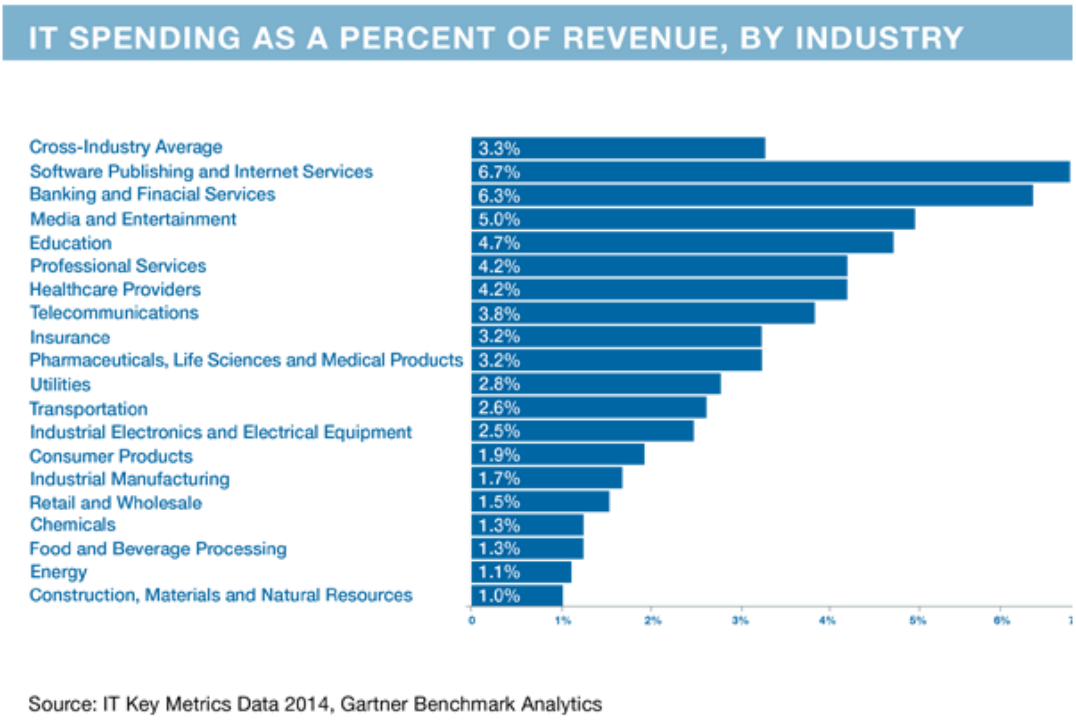


Figure 4

You have all of the numbers needed to complete a cost benefit analysis, but how do you evaluate non-financial value?

In this case, how long does it take a member to find a problem resolution when visiting our site?

The values the constituents may be interested in are increasing traffic, increasing time on a web site, or cutting the time to provide resolution to an issue. While you may not

provide a hard financial number to delivering these capabilities, they will have value to your constituency. Do you know enough about your stakeholders to tell them why they should care about your architecture?

Remember! The architecture you create IS the strategy. As you create the architecture, you must be able to describe the value it provides to your constituents.

2.4 - BUSINESS VALUATION

Business valuation is; identifying how and when to invest in a particular technology and how to manage the overall portfolio of technology investments. This includes common techniques for evaluating and proving the financial benefit of architecture.

Cost of Not Knowing

Gartner reports of 19 industries, only five saw an increase in IT spend as a percentage of revenue. 85 - 90% of IT budgets are towards maintenance spending. Demand-side drivers of higher IT spending as a percentage of revenue includes:

- Highly integrated IT components in the product suite.
- Higher service levels for mission-critical systems.
- Higher usage of knowledge workers.
- Decentralized IT environment

Skills Analysis

- Do you know how to define (in numbers) a technology decision?
- Have you evaluated the cost of two competing technologies stacks?
- How much does your company spend on maintenance?
- Have you taught valuation to other architects?

Architecture value refers to the overall value of a corporation's technology in relation to their shareholder's interests.

Architects create value in two ways:

- Ensuring that technology investment provides return beyond investment levels.
- Creating a technology investment portfolio that ensures the long-term viability of the organization.

Your organization must determine at what scope and level they will evaluate technology value:

Specialization Knowledge Areas in regards to determining return on investment (ROI).

Software Architect:	Analyze ROI of project technology
Infrastructure Architect:	Analyze ROI of data center technology
Business Architect:	Analyze ROI of domain/requirement technology
Information Architect:	Analyze ROI of information related technology

2.5 - DETERMINING FINANCIAL VALUE

Without exception, calculating the value of a complex solution is challenging. For this example, imagine you have to calculate the value of adding a smartphone client.

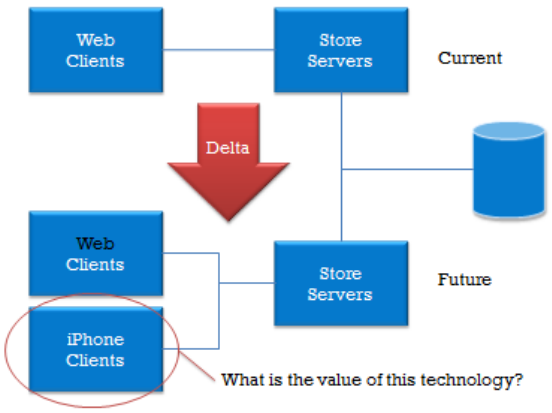


Figure 5

Some factors you will need to consider:

- Where are the clients or customers that use the new smartphone app going to come from? Will they be new clients or existing web clients?
- What is the cost of goods sold through the web client compared to smartphone clients?
- Which and how many smartphones clients will be supported?
- Is there good will or perception value in providing smartphone clients?
- Can the Helpdesk and other Operations staff support the new clients or will additional spend be needed to prepare them?

These are just a small sampling of the questions that will arise in this simple example. What about the projects you currently work on?

Here is an example of capturing some of the more tangible costs and comparing to the benefit provided.

CBA Element	Cost	Benefit
Development	200,000	0
Maintenance (yearly)	20,000	0
Complexity	15,000	0
Training	5,000	0
Documentation	2,000	0
Hardware/Software	100,000	0
Consumer sales/day		25,000

Figure 6

The development and maintenance costs will be pretty straightforward, but what about complexity? What if the current staff is not familiar with developing smartphone apps for the platforms needed? Will you train them or contract external staff with that expertise? Once you've detailed some of the costs, you should be able to estimate the projected revenue stream based on information from the business unit and calculate when the project will break even and become a profit center for the organization.

Here are examples of the business calculations you will need to be familiar with in order to complete business valuation.

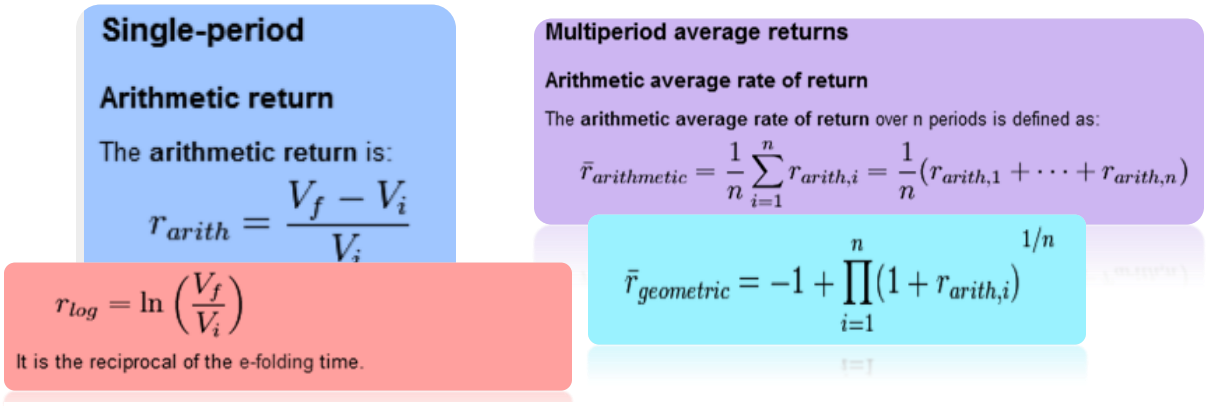


Figure 7

One interesting note, technology moves at an incredible pace and as someone familiar with technology we realize that. We may not realize that business process and how business value is calculated changes just as quickly. Consider seeking out the business manager responsible for the projects you are engaged in and find out how they will measure and monitor spend and success.

One calculation we are more comfortable with and likely have experience with is value through reduced redundancy. You have likely been involved in a consolidation project at some point in your career. Calculating costs savings and selling the idea to an executive sponsor can be less challenging that other ideas, and as the innovators for the organization we are more familiar with where efficiency can be gained that any other stakeholder, save the Operations team.

Here is another approach to showing value and will help in selecting the products that will be used.

Cost Component	Product ABC	Product XYZ
Software License	0	40,000
Hardware	120,000	60,000
IT Environment	20,000	10,000
Training	30,000	10,000
Consulting	35,000	0
FTEs	300,000	100,000
Contractors	40,000	0
Total Cost of Ownership	510,000	220,000
Difference		+310,000

Figure 8

Consulting architects are typically quite familiar with this type of comparison as this is a tool they used to win bids. Staff architects may or may not be familiar with these, or have seen and validated proposals that have similar comparative numbers.

By exposing costs beyond software licensing costs you are able to provide a better overall estimation of the true cost of a solution based of product selection. This is also a great tool for you to use in showing the value of architecture to executive sponsors and other stakeholders. It can be used with the business owners to justify selections made and can be used with technical staff to show why a specific product stack was selected.

REFLECTION POINTS:

Read through this list and consider some of the questions asked and how they impact overall cost. By the way, many people have never heard the term “persistence framework”. This is just another way of describing a data storage solution. We encourage you to research any term you are unfamiliar with as this is the language of architecture.

- Does a code loop impact value? Does a persistence framework? Does a product suite?
- Do you have valuation metrics/documents for all of the technical frameworks or platforms in your current environment?
- How much money does your company spend on persistence each year?
- How does that impact specializations?

- How would you change your current environment to make valuable technology decisions?
- How many architects would you need to perform valuation on critical projects?
- What is the cost to keep an existing customer versus getting a new one?

2.6 - ARCHITECTURE METHODS AND TOOLS

Architecture methods and tools is defined as; the strategic and tactical use of business architecture methods and tools, including but not limited to business process engineering, business process management, business process modeling, workflow, and similar technology in relation to business capabilities and design. Best practices for integrating business processes that span multiple internal organizations.

Additional definitions and/or information within this area are:

Architecture Methods

- Is familiar with various Business Architecture Methodologies
- Understands various approaches to Business Process Management
- Can compare several business architecture methodologies
- Champions the use of an architecture framework or methodology

Architecture Tools

- Strategy, objective tracking and visualization
- Visualization techniques
- Methodology tracking throughout the life cycle

Cost of Not Knowing

- Limited relevance with the business-oriented team members
- Inability to provide relevant input

Skills Analysis

- Name 3 Industry Architecture Methodologies?
- Compare a methodology like TOGAF® to the Zachman framework (EAF)?
- Championed the use of an architecture framework or methodology?
- Delivered or customized one?

Concepts

- An Architecture method or methodology is the process by which architecture is created, including:
    - The related tasks and steps
    - Inputs and outputs of each step
    - The specializations involved with each task or step

Architecture methods must be evaluated for:

- Scope of effectiveness (project, domain, enterprise)
- Connectivity to other activities, skills and concepts
- Appropriateness for target engagement model (later module)

Architecture tools are any and all applications and documents that are required or support the delivery of effective technology strategy. Examples:

- Tracing requirements to value and deployment
- Reviewing quality attributes across projects
- Describing technology impact on business processes
- Calculating ROI/Investment/etc.

Specialization Knowledge Areas in regards to Methods and Tools.

Software Architect:	Integrate Systems Development Life Cycle (SDLC) in methodology and effectively use shared processes.
Infrastructure Architect:	Integrate technology stack in methodology.
Business Architect:	Manage process and technology integration.
Information Architect:	Integrate information flow into methodology and processes.

REFLECTION POINTS:

Read through this list and consider some of the questions asked and how they impact your understanding of Business Architecture Methods and Tools.

- Exactly what elements of your job do the existing 'frameworks' help with? Think in terms of day to day effort.
- What tools would actually help you be a better business technology strategist?
- What is the difference between a development methodology and an architecture methodology and framework? What does each give you?
- When do methods and tools begin to truly impact your day to day activities?
- How much did it cost to create or modify your existing methodology?

2.7 - REQUIREMENTS DISCOVERY AND CONSTRAINTS ANALYSIS

As an architect, you are expected to know what requirements and constraints are, are able to leverage architecture frameworks and can deliver against one, can create a successful strategy from poor requirements, and can assess requirements and constraints through business risk analysis. In a perfect world you would get fantastic requirements, but typically you have to work with what you are given.

Requirements Discovery and Constraints Analysis is the tools and techniques for understanding business requirements with multiple strategic impacts, including how such requirements and constraints are formed internally and externally.

The IT architect is expected to apply those constraints and requirements to their technology and business decisions, using constraints analysis to understand and plan for technology capabilities of the current resources/environment.

Wish all you want…architects don't get good requirements. Good architects learn to ask questions and what get defined limits around; Quality attributes strategy, goals, governance and compliance.

Cost of Not Knowing

- Waste, scope = $$
- Fragile Architecture

Skills Analysis

- Can you describe what a requirement and constraint are?
- Would you be able to deliver against an architecture framework?
- Have you ever created a successful strategy from poor requirements?
- Have you taught these things to others?

Concepts

- An architectural requirement is a representation of an architectural concern that impacts the technology strategy.
- An architectural constraint is any process, tool, decision or framework that limits choices within a technology strategy.
- Develop requirements that impact the technology strategy by ensuring that:
    - They are of the appropriate scope (not too low or too high)
    - They extract constraints and opportunities to impact actual value

Specialization Knowledge Areas in regards to Requirements Discovery and Constraints Analysis.

Software Architect:	Manage requirement and constraint impact on software.
Infrastructure Architect:	Manage requirement and constraint impact on data center.
Business Architect:	Track and manage requirements through business value.
Information Architect:	Manage information requirement value.

REFLECTIONS POINTS:

Read through this list and consider some of the questions asked and how your knowledge of Requirements Discovery and Constraints Analysis can help to answer them.

- What is a non-architectural requirement? How do you know?
- How does this skill relate to the design skill Requirements Modeling? Why are they in two sections?
- How does that impact specializations?
- How does this relate to traceability in the lifecycle?
-

2.8 - STRATEGY RATIONALIZATION AND DEVELOPMENT

As an IT architect you are expected to be able to define a business objective, should be able to match technology to the business objectives of a project, identify a technology strategy that positively impacts the entire organization, and lead the executive sponsors and business managers in defining IT strategy.

By definition is the partnership between the business architecture and the technical architecture relative to the creation of a business model that defines the principles, standards, structure, and dynamics of the integrated business and technical strategic direction.

The IT architect should have an understanding of the components of a sound architecture strategy and be able to evaluate a business problem and identify an appropriate technology direction.

Do you understand how business strategies are formed? Ever wonder why people say “the business” and then “IT” (and not just “the business”)? Creating a technology strategy from a business strategy (or better the other way around), now that is how you get promoted!

Cost of Not Knowing

- Useless spending
- Maintenance spend

Skills Analysis

- Can you define and create a business objective?
- Could you match technology to the business objectives of a project?
- Have you identified a technology strategy that impacts your entire enterprise?
- Are you part of the team leading business and IT in defining strategy?

Concepts

- Strategy rationalization is the process of coming up with a technology solution which delivers or supports a business goal.
- Business goals are created in exactly the same way that technology goals are
- Architects should always attempt to lead the business
- Architects should equally participate in business strategy which is led from other business units
- In either scenario an architect should be responsible for defining the technology strategy
- A technology strategy is any plan that uses technology as its central component in accomplishing a goal
    - Growth in market size or share
- It is still a technology strategy even if other business units are involved

Specialization Knowledge Areas in relation to Strategy Rationalization and Development.

All:	Participate in the development of project strategy development; ensure delivery against business goal.
Business Architect:	Lead LOB/Capability strategy development across all related projects.

REFLECTION POINTS:

Read through this list and consider some of the questions asked and how they impact your understanding and use of Strategy Rationalization and

Development:

- How much should an architect know about marketing, finance, operations, sales, or other?
- Can you think of any pros or cons for this statement, “Business architects do not architect the business; they provide the technology strategy underpinnings to any successful business capability.”?
- When is the last time you helped in developing an IT strategy and provided rational for an innovative approach to delivering value?

- How much should architects know about the business environment?

2.9 - INVESTMENT PRIORITIZATION AND PLANNING

Are you able to leverage multiple resources for technology trending and planning? Manage suppliers through the bidding, selection and oversight process, or provide a bid if you are a consulting architect? Map current IT spend and create a capability map to the trending and investments? Can you lead an initiative to increase capability mapping, portfolio investments and optimized procurement? As an architect these are some of the skills you are expected to possess.

By definition it is the management of asset and project portfolio lifecycles, including planning and managing, and understanding of the very different investment approaches required for each.

- Do you know where IT spends its money?
- Do you know what trends it could be investing in?
- How do you research new technology?

Cost of Not Knowing

- Lack of competitive advantage
- Poor spend choices

Skills Analysis

- Can you name 3 resources for technology trending?
- How does your current IT spend and capability map to the trending?
- Have you been part of an initiative to increase the capability mapping?
- Have you mentored others in this area?

Concept

Investment prioritization and planning for architects is the process of describing:

- What technology the company will spend money on and why
- What priority technology investments have and why
- How technology strategy relates to the project portfolio including project priority
    - This is a major area of contention between PMO and architecture
- The ownership of project priority and planning often requires significant political influence
- There is no accountability in technology strategy without ownership of technical investment

Specialization Knowledge Areas in regards to Investment prioritization and planning are:

Software Architect:	*P*rioritization details for upcoming projects.
Infrastructure Architect:	Develop prioritization details for upcoming projects.
Business Architect:	Map technology prioritization to business impact.
Information Architect:	Develop prioritization details for upcoming projects.

2.10 - PORTFOLIO PLANNING

In IT environments with little or no architectural practice portfolio planning can be left to people that do not have a broad understanding of IT or current and future capabilities. Additionally, they may have some opinion of technology that is formed through the advertising in the sources they use to gain insight, and that may lead to portfolio planning based on the latest hyped approaches and products. Have you seen organization with symptoms like the ones listed here?

- Business units decide strategy
- Stakeholders prioritize projects
- Little or no understanding of existing application portfolio
- No metrics for complexity or priority
- Strategic justification is ad hoc
- Technology investment decided by project team
- No governance

A good starting point for portfolio planning is to list all of the proposed projects together, put together rough timelines, resource requirements, and dependencies. In general, moving from timeline management to strategic return allows architects to move from cost center to profit center.

The results of such an effort may look something like this and any capable program or project manager can deliver this.

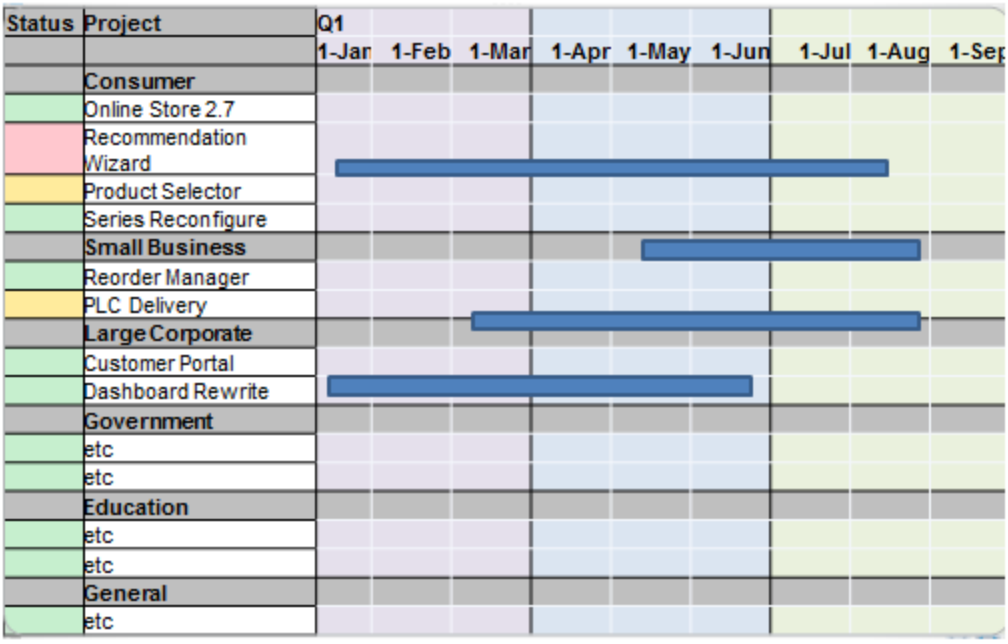


Figure 9

As an architect you can use a weighted scale to further prioritize projects. Consider adding a column for importance to the organization for each of the projects. For instance, will the project provide competitive advantage, or catch your organization up with the competition? Is it needed to protect an existing revenue stream that the company is reliant of to stay afloat? Add a column for ROI and calculate each, along with competitive ranking, project complexity, and any compliance or governance flags.

Status	Project	Quarter Start End	Business Priority	Estimated ROI	Competitive Rating	Complexity Rating	Compliance Flags	Governance Flags
	Scales	Dates	1 to 10	$Amount/ Percentage	1 to 10	1 to 5	-	-
	Consumer							
	Online Store							
	Recommendation							
	Product Selector							

Figure 10

Including this information in portfolio planning and project prioritization will provide much stronger alignment with the business goals and objective of the organization. As a consulting architect, you may not be included in this process, but if you are aware of the planning cycles in the organizations you engage with, you can provide thins data to help solidify your role as a trusted advisor.

REFLECTION POINTS:

Read through this list and consider some of the questions asked and how you would leverage them during project prioritization and planning.

- Who should own a technology investment budget?
- What issues would arise if architects were responsible for technology procurement?
- Describe a project that would have been changed if you had investment authority.
- Where do technology investment authority and project priorities diverge?

2.11 - KNOWLEDGE MANAGEMENT

As an architect you are expected to be familiar with different levels and tools for managing knowledge and intellectual assets, should be able to implement or effectively utilize corporate knowledge management systems, consistently increases the knowledge management and learning capabilities of the organization, and understand and evangelize the value and relevance of intellectual property.

Knowledge management is the way in which business information is built, stored, managed, and made easily accessible, including current issues around the organizational challenge of maintaining long-term organizational memory.

Knowledge Management is the beginning of information architecture, database management, knowledge management and usability all combined.

It's not just how you STORE information, it is how you USE information and integrate IT knowledge into common practice. You don't always need massive products to do it better. Process is not enough; it takes people.

Cost of Now Knowing

- Re-inventing the wheel
- Poor communication

Skills Analysis

- Can you name 3 different levels and tools for managing knowledge bases?
- Could you implement or effectively utilize the corporate knowledge base?
- Have you helped increase the knowledge management capabilities of the organization?
- Have you mentored others in this area?

Concepts

- Knowledge management should include:
- Information storage, retrieval and delivery
- Information structure, rules and format
- Information usage

Much like application development is the kernel for specialization in software architecture; knowledge management is the kernel for information architects.

Specialization knowledge areas within Knowledge Management are:

Software Architect:	Deliver and update architecture descriptions.
Infrastructure Architect:	Deliver and update architecture descriptions.
Business Architect:	Develop strategic technology plans with integrated artifacts.
Information Architect:	Deliver and update architecture descriptions.

REFLECTION POINTS:

Read through this list and consider how your prowess in Knowledge Management would help with answering these questions:

- What elements of data storage and retrieval impact the technology strategy?
- What is the difference between data oriented, structure oriented or usage oriented components of architecture?
- When would a software architect bring in an information architect?

2.12 - DECISION SUPPORT

As an architect you must be able to describe the technical basis for decision management, identify and deliver a technology which helps the company to decide faster, and have insight into key business metrics that are relevant for timely and accurate business decisions.

Decision support is the understanding and application of decision support and "smart" systems, including basic concepts and components in decision and business intelligence systems and demonstration of effective architectures using these components.

Cost of Not Knowing

- Missed opportunities
- Lack of reuse
- Skills Analysis

Do you know what the technical basis for decision management is?

Could you identify and deliver a technology which saves or makes the company money?

Do you know how much your company spends on maintenance?

Have you taught valuation to other architects?

Concepts

- Decision support includes all aspects that influence the way that significant decisions are made in an organization.
- Understanding how decisions are made is key to influencing strategy:
- Decisions are based on understanding of business goals and context
- Decisions are very risky if they are not backed
- Decisions that can be automated represent a significant opportunity

Specialization Knowledge Areas in decision support are:

Software Architect:	Utilize decision support and identify key business decisions.
Infrastructure Architect:	Manage decision support environments and connections.
Business Architect:	Identify key decision support opportunities.
Information Architect:	Business decision management and documentation.

REFLECTION POINTS:

Read through this list of questions and consider how you would leverage the skills represented in Decision Support to help answer them.

What constitutes an architecturally relevant decision?

Which specializations should be involved with decision support? How?

Is there a need for skills in business intelligence needed to enable decision support?

2.13 - PROJECT PRIORITIZATION – BUSINESS CASES

Activities at Scoped Levels in the Organization

Now that you have reviewed the key concepts for the skills and capabilities in the Business Technology Strategy pillar, consider how they are applied in a simple yearlong lifecycle structure. In every organization there is a cycle where projects are presented and prioritized, architecture is created, and projects are developed and delivered. Additionally, in organizations with mature architectural practices, post mortems are done not only on projects, but also on your architecture practice.

Regardless of whether you are a staff architect in an organization or a consulting architect that only has exposure to a portion of this lifecycle, you should have an understanding of what an architect is expected to provide during these phases and what artifacts they help create.

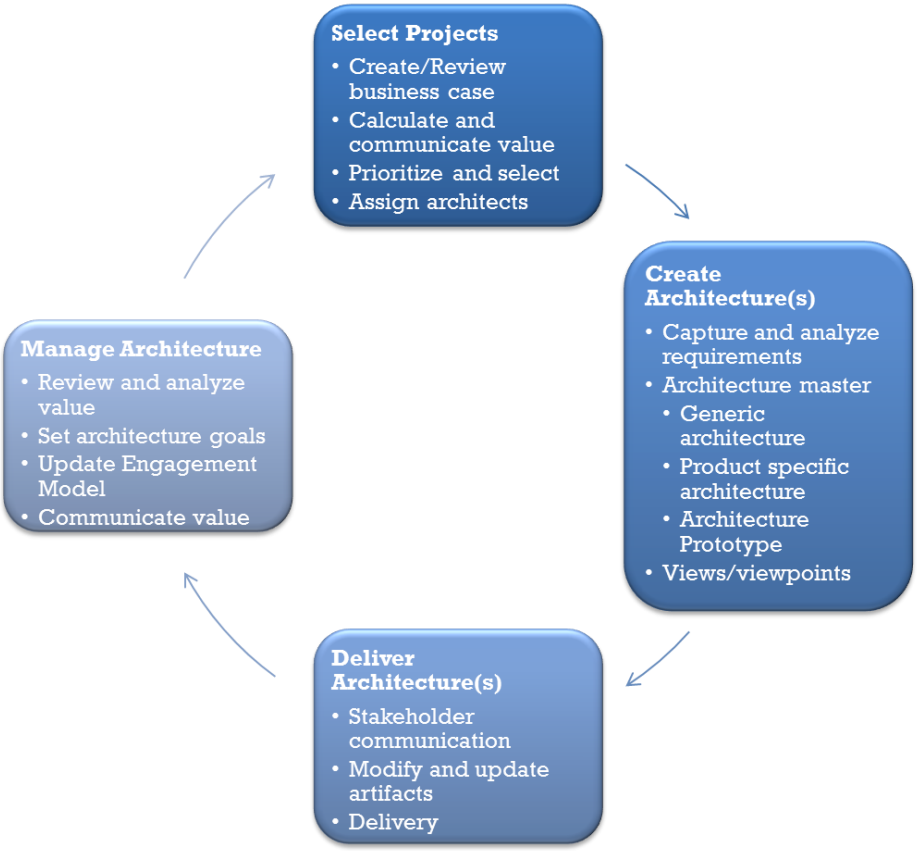


Figure 11

Engagement Scope/Business Case

Ideally, architects are involved in helping with project prioritization. During this phase, details are limited and will require more experience as an architect. Additionally more industry and domain knowledge will help you be more successful. Understanding business fundamentals, strategy development, and industry analysis will help during this phase.

Different organizations approach this phase in different ways, but ideally, we want to be involved during this phase as we can provide the more value and positive business impact during this phase.

A business case captures the reasoning for initiating a project and what value the project should provide. Architects should verify the value of the technology, add strategy, and make sure the solution fits the governance model, existing portfolio, and can be supported by the Operations group. They can also add value by initiating business cases for projects they feel will add business value. Business valuation, compliance, and decision support skills will help during this phase.

- Architects interact with business cases:
    - By verifying accuracy of value from technology
    - By adding technology strategy
    - By ensuring they align with existing portfolio
    - By creating technology-led business cases

If you are a consulting architect, you typically do not have visibility into this phase of projects, but you still should know what is expected and what the output of this phase is. By gaining status as a trusted advisor and having business skills, you will be sought out to provide input earlier in the lifecycle of a project thereby allowing you to have greater business impact.

Most organizations have templates and processes in place for justifying a project. As an architect, you are interested in the completed business case providing the information needed to help with portfolio management and project prioritization. As projects are funded, architects must be able to rationalize and optimize projects across the enterprise. This requires all of the skills represented in the Business Technology Strategy pillar.

Business cases have some key components, including an executive summary which is normally only one or two paragraphs long and provides high level details of the need and how it aligns with the organization's business objectives. There will not be many details available so project overview, evaluation, and project selection justification will be high level.

In your current role as an architect you may or may not be part of the process of reviewing business cases, but in organizations with moderately mature practices, there are architects the help write and review business cases. As you grow in the role of trusted advisor you will have to be able to provide relevant input into business cases and increase your personal value to the organization. If you are in a consulting role, this reflects well on you and the company you represent.

As you review business cases, make sure the intended value of the project is clear, including how that value is calculated and will be measured and tracked.

Remember, you want to capture as much data as you can to help you provide valuable input but do not want to be seen as a roadblock.

Speak to some architects and they will tell you all business cases overstate value and understate cost and risk. As you review business cases, ensure the data is as accurate as possible given the limited information available. Leverage industry trends in order to help with projections. For instance, Tinkleman wants to launch an e-book product and content.

What other companies have readers or content they sell and what did their initial ramp and penetration look like? Will that data help you to project best case and worst case scenarios?

High Level Questions

Question-based patterns are starting to gain traction in the IT architecture and business architecture spaces. Having a standard set of questions as a template for reuse will help you to capture the information you need in a predictable manner.

Use a standard set of questions to get as much detail as possible:

- How many systems will be changed?
- How many users? How often will they use the system?
- Which workflows/processes does this modify? Are they documented?

Questions at this level are meant to get a profile of the vision not derail the business case. If architects helped create the business case you can probably skip this step (they will already exist).

To help ensure the accuracy of the business case, create internal (to the architect team) models:

- Formal modeling language not required
- Extremely high level
- Only dig into areas to ensure your understanding
- Keep the models for later
- Do not publicize the existence of the models

As an architect you have broad technical knowledge of multiple technology domains and products stacks. You also have better visibility into the existing environment and other solutions being delivered than other stakeholders. Additionally, you keep up with current and future trends in technology and innovation in the general use of IT as well as in the industry you are working in.

Your thoughtful and considered input on business cases can significantly improve the value of any proposed project.

Keep Them Honest

Executive sponsors may have a different perspective on how value is calculated and whether the value should focus on their team or overall value to the organization. As an architect you will be placed in politically charged environments where you may be pushed to massage your estimates.

If you need more time for research, consider asking for the time and describing the value the additional time will provide.

As an architect you will be placed in politically charged environments

- Consider asking for the time and describing the value the additional time will provide.

- Provide a value statement if more review or exploration is needed. “We believe we can reduce these costs by 20% if you give us a week to do a little research”.

As stated earlier, you have broad technical knowledge of multiple technology domains and products stacks, visibility into the existing environment, and other solutions being delivered. By keeping up with current and future trends in technology and innovation you can preset business cases for projects that will provide value to the organization. Strive to understand the funding cycles in your organization, or, if you are a consulting architect the funding cycles of your customers to present potential projects for consideration alongside other, sponsor-initiated projects.

- By keeping up with current and future trends in technology and innovation you can preset business cases for projects that will provide value to the organization.
- Strive to understand the funding cycles in your organization

2.14 - DESCRIBING VALUE

Have you ever seen a proposed project that would have provided great value not get prioritized or funded? Have you ever seen projects successfully delivered that provided great business value and the only ones that knew about them were you and your manager? Ever deliver a project that opened a new revenue stream and boosted the organizations business goals for the year and other teams seem to get all of the credit?

As an architect, you need to calculate business value, and present and collect information on the value of IT and the value of architecture. For some, this feels like bragging, but it is not. It is simply showing the value you and your teams bring to the organization.

Here are some of the tools used to calculate value.

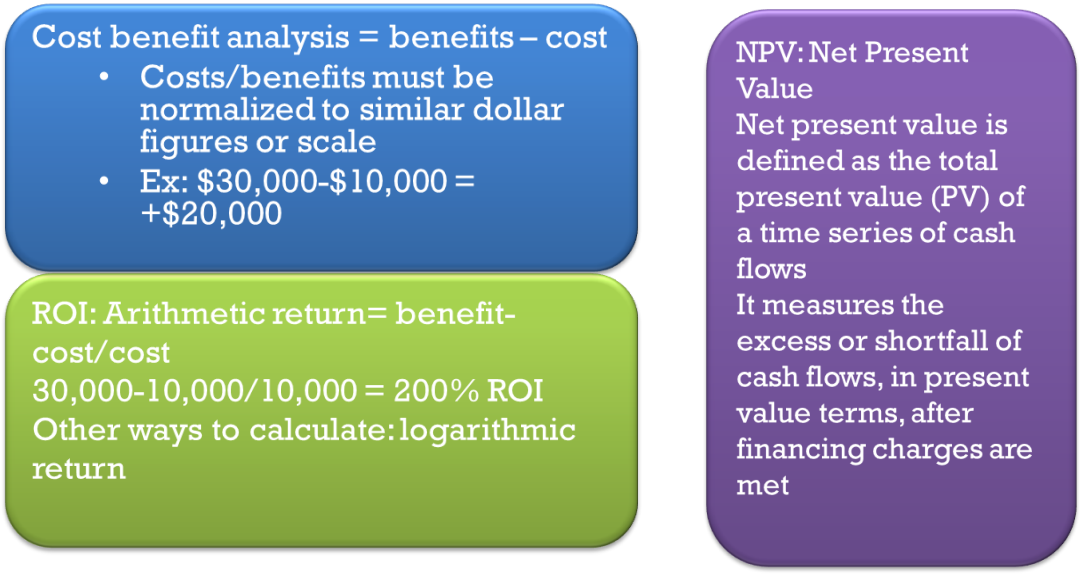


Figure 12

As you can see, the calculations are not too challenging. The challenge comes in identifying the correct data to complete the calculations. Both staff architects and consulting architects may or may not have access to the financial data need to complete these projections, and

more often are not required to provide these details. However, we encourage you to grow you skill in this areas as it will be needed to advance your career, and trending suggests it will be required over the next several years.

The challenge with trying to describe value is in determining how value will be measured and where the value will come from. In the example, customers can complete a sales in person, on the phone, and online. The cost of sale online is least expensive and cost in store is most expensive.

If you provide a solution that will increase online penetration, you have to determine if the sales are new sales or will pull from existing sales and calculate the value to the company based on that data, not just the revenue stream generated through online sales.

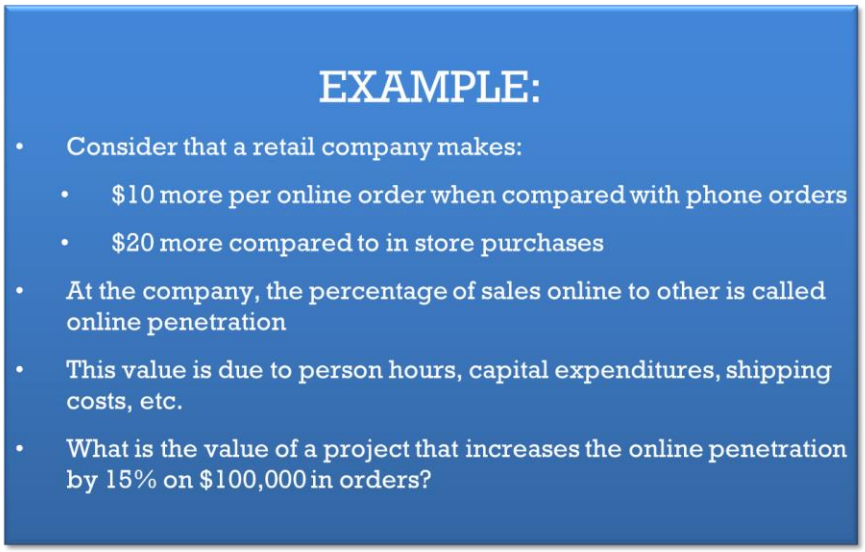


Figure 13

Given the previous example, consider the following questions and their impact on overall value. How would you incorporate these into you calculations?

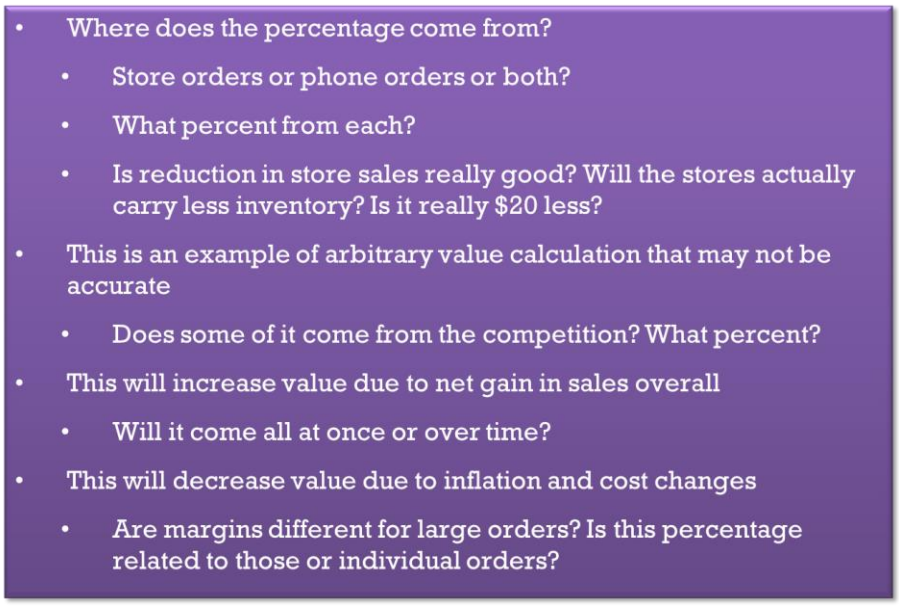


Figure 14

Your intent should be to provide as realistic information as you can. Many times we are provided numbers or nebulous percentages to use in determining value. As you approach validating this information through discussion with the people that initially provided the data, consider researching similar or competitive groups to facilitate discussions. Since you likely do not have authority and do not want to opening challenge others, you must use your interpersonal or human dynamics skills to question without confronting given the opportunity.

Your job is to accurately calculate the impact of the technology on real value:

- How do you know it will increase online orders by 15%?
- Data from similar project?
- Data from competitor?
- Estimated increase in Web site traffic related to increase in historical increases in online orders?
- What are competitors doing? Is there data available to analyze?

2.15 - CONSTITUENT VALUE

Technology-focused people tend to feel technology has an innate value in and of itself. As you determine business value, you must think about the entire package, including people, process, usage, and technology. Reflect on the examples here. How would you have sold a project to create Google Maps?

Example:
Large corporate sales portal is improved to increase customer satisfaction

Example:
E-mail system is improved due to complaints

Example:
Google doesn't generate revenue but increases traffic and awareness

Leveraging historical information and providing tools to track results over time is valuable. For instance, if you created a portal solution for an existing sales staff, would you suggest the solution be monitored by tracking past and futures sales and comparing the change for people that started using the new portal versus those that did not? Would you try to determine the impact of end user training based on traffic using new capabilities?

Here is one simple example of how you could approach determining the value of your solution.

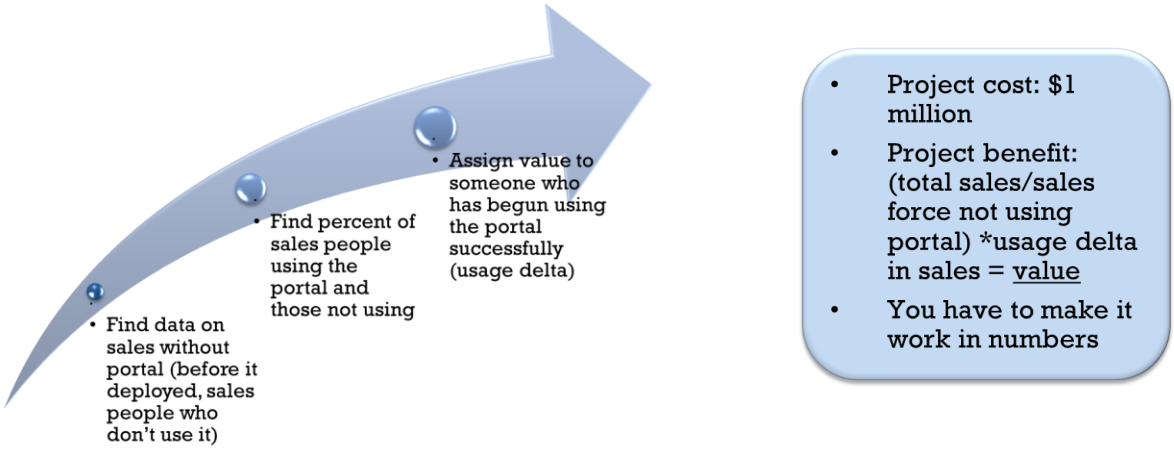


Figure 15

REFLECTION POINTS

The architecture you create IS the strategy. As you create the architecture, you must be able to describe the value it provides to your constituents.

CHAPTER 3: HUMAN DYNAMICS PILLAR

3.1 -OVERVIEW

Human dynamics is the set of interpersonal skills that allow you to work with stakeholders from various backgrounds and with various interests effectively. You may or may not have any authority or may be a virtual team member and seen as an interloper or roadblock to stakeholders goals.

To be successful, you will need to combine strong written and verbal communications skills with an understanding of the environment and personalities involved, and how you can modify your communications style to effectively lead your projects to successful completion. Below is a visual image of the Human Dynamics taxonomy.

Figure 16

Taxonomy is another term for grouping or arranging skills. This taxonomy is not hierarchical, rather it is just a way to organize and display the skills represented in the Human Dynamics pillar. Each is critical and all are inter-related, but there are not strong dependencies between them, with the exception of written and verbal skills.

Written documents or presentations you give many times offer the first impression of you to others. Concise, well-formed documents whether email or a whitepaper should be the trademark of an architect. Additionally, being able to communicate complex ideas to people from diverse backgrounds can be challenging. Once you have evolved these two bedrock skills, you have a platform to grow the other skills.

Human dynamics are the *communication, political and leadership skills* necessary for any IT Architect. They are a necessary and a must-have survival skill for IT architects.

Human dynamics are critical and involve up to **80%** of IT architect routines - Ignoring this can cost you a job. The good news is: If these skills do not come naturally to you, they can be learned!

As we have stated, BTS skills and being able to create an IT strategy that is aligned with the business goals and direction are what set us apart from the other stakeholders on a project. However, human dynamics skills are critical to enable us to successfully navigate politics and deliver a solution that provides innovation and value.

3.2 - PRESENTATION SKILLS

As an architect, you will likely do a lot of presentations, in meetings and speaking at conferences. As you create content, you must use creativity and strong visualization to hold the audience's attention. Additionally, if the audience is focusing on typos and punctuality, or if they have to try and decipher the message you are trying to convey, you will lose their interest.

Consider researching current techniques for more interesting presentations and practice various techniques. Having strong public speaking skills to deliver a great presentation or get executive sponsors and stakeholders to understand and buy-in to your vision is critical. Time spent honing your presentation skills is time well invested.

Presentation skills is defined as the ability to manage presentations made to key stakeholders, clearly identifying technology strategy decisions and opportunities, and tying those to the concerns of the stakeholders in that audience.

Presentation Skills

- Includes both writing and speaking skills
- You do NOT have to be the best speaker to be able to present but you have to be articulate
- No one listens to you if you can't communicate well
- Don't imitate others, use your own style
- Presentations can be in writing too
- Putting design thoughts onto paper/blogs/news/article, etc.
- Tools include Editor, Evernote, Prezi, PowerPoint, etc.

The Cost of Not Knowing

The cost of not knowing includes no communication which ultimately leads to no IT architecture. You will have difficulty growing your career and there will be a lack of perception value.

Skills Analysis

- I am aware of basic presentation skills and techniques
- I have studied presentation skills in detail
- I regularly use my presentation skills in my specialization
- I mentor others in presentation skills

Specialization Knowledge Areas in Presentation Skills are:

Software Architects: present regularly to software project stakeholders; provide deeply technical presentations; prepare presentation materials to be used by other architect team members.

Infrastructure Architects: present regularly to infrastructure project stakeholders; provide deeply technical presentations; prepare presentation materials to be used by other architect team members.

Business Architects: present major presentations to executive and managerial staff and are able to present effectively at any level in the organization, and use technology strategy presentations to impact change in the organization.

Information Architects: are able to present effectively at any level in the organization, provide deeply technical presentations, and prepare presentation materials to be used by other architect team members.

3.3 - PRINCIPAL OF PRESENTATION

ICEPAC is an acronym we can use to develop interesting presentations for conference delivery and stakeholder meetings:

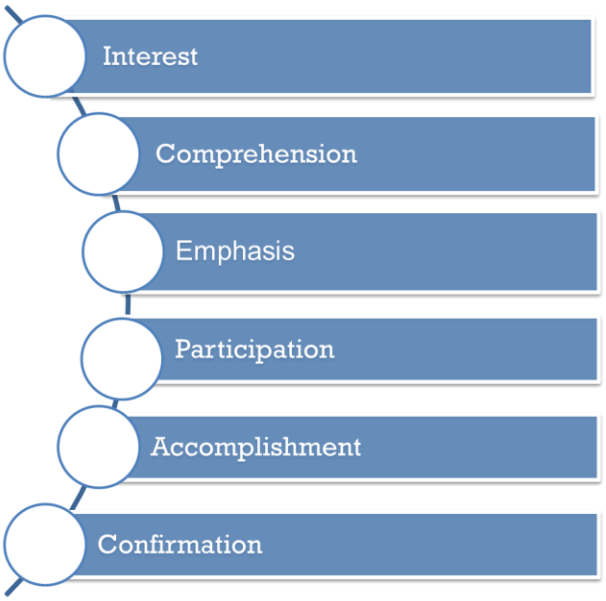
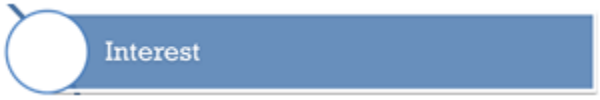


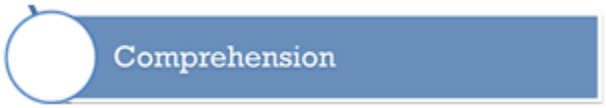
Figure 17

Architects should be masters of reuse. From technical writing and training creation we can borrow a principle of instructional design.



Spark interest. It seems obvious, but people won't pay attention if they are bored. Help them understand the value, to them, of knowing the information and explain the consequences of not knowing the information. Relate the information to the audience by finding new and innovative ways to present the material, and avoid death-by-PowerPoint or bullet list oblivion.

- People don't pay attention when they are bored
- Help them understand the value
- Explain the consequences of not knowing
- Relate the information to the audience
- Find a new and innovative way to present



Check for comprehension, though the softer skill of reading body language and checking for eye contact. Ask questions of people attending meetings and make sure you always spell out the three letter acronyms you use. After all, how many ways is SOA used right now? Would a lead developer and network engineer interpret SOA the same way?

- Ensure the audience understands
- Ask questions before you start
- Adjust your presentation to the audience
- Realize sometimes the pre-requisite knowledge is not there. If not:
    - Build/create it.
    - Adjust your material to suit the level of knowledge.

Make sure and emphasize key points so the audience remembers them. A good approach to take is the same one used by trainers, tell them what you will tell them, tell, them, and then tell them what you just told them. To move information from short-term memory to long-term memory most humans need to hear something 2-3 times in 15 minutes. Additionally, stating "This is important" or pausing for a moment or two are good ways to bring attention to what you are saying.

- Emphasize key points
- Use visual supports
- Use verbal supports
- Do something out of the ordinary
- Make use of repetition
- Make use of repetition
- Say "This is important."
- Make use of repetition. This is important!

Participation

Participation helps. Most people naturally want to help so if you pose a question to the audience or members of a meeting, you get them engaged and they will then have a vested interest in the outcome.

Also, combine listening with doing. People remember doing things better than they remember hearing about things.

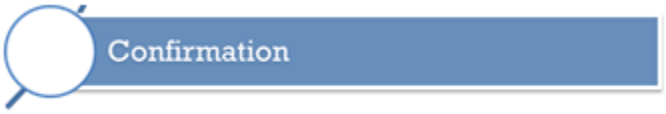
- Combine listening with doing
- Ask questions of the audience:
    - Pose – Ask the question out loud
    - Pause – Let the audience think about it
    - Pounce – Pick the person to answer
    - Ask for anecdotes and examples
- Use leading and thought-provoking questions

Accomplishment

Having you ever attended a meeting with no agenda, no clear idea of what needed to be decided or agreed upon? Feeling a sense of accomplishment is important, especially in a meeting. Make sure and set clear agendas, with goals for the meeting and outcomes you desire.

- Helps the audience feel a sense of accomplishment
- Review what you have presented:
    - Help them frame the knowledge as valuable to them
    - Understand where/how it benefits them
    - Prior to the presentation/lesson, ask for "objectives" from the audience

Before closing the meeting, asked if there are any other topics that need to be covered in that meeting or the next meeting, and recap the decisions that were made, sending meeting minutes highlighting any action items and setting completion dates for each.



Some points about confirmation have already been made, possibly 3 times in this lesson. When closing a meeting with stakeholders, confirm that they understand what the meeting objectives and everyone is clear on decisions made and follow-up in email.

- Confirm that the audience understands at each stage of your lesson or presentation
- Do not continue until they understand
- Ask questions of the audience
- Encourage your audience to ask you questions
- Even as the presenter, find a way to ask questions

3.4 - CUSTOMER RELATIONSHIPS

The Architect needs to demonstrate an understanding of the psychological dynamics of customer management and discusses business imperatives, modern techniques and tools for relationship management, industry engagement, contractual agreements, transparency and accountability, and related issues. Managing high-risk scenarios is also a demonstrated competence of the role.

Customer Relationships

- It is best to form a provider-customer team.
- All the communication channels must remain open.
- Know the strengths and weaknesses of each one of the "characters" involved in a technology solution.
- Avoid generating situations where the real "personality' of the individual interferes with the project resolution.

Cost of Not Knowing

The cost of not knowing includes the lack of credibility, the inability to implement ideas, you won't get support from others, and you may suffer from a loss of promotion, or worse, lose your job.

Skills Analysis

- I am aware of basic customer management principles.
- I could apply customer relationship techniques on a project.
- I regularly practice customer management in my specialization.
- I lead others in improving customer relationship management skills.

Specialization Knowledge Areas in Customer Relationships are:

Software Architects lead customers internally and externally to effective software technology strategy; provide mentoring to project staff in customer relationships; develop deep bonds to customers.

Infrastructure Architects lead customers internally and externally to effective infrastructure technology strategy; provide mentoring to project staff in customer relationships; develop deep bonds to customers.

Business Architects lead customers internally and externally to effective technology strategy; provide mentoring to staff in customer relationships; develop deep bonds to customers; lead customers through technology impact to line of business and capability.

Information Architects lead customers internally and externally to effective information technology strategy; provide mentoring to key staff in customer relationships; develop deep bonds to customers.

Customer Goals:

The concepts of customer goals include the following: Acquire, retain, and extend. As an architect, we must become an integral part of any IT project's lifecycle and leveraging the tools used by sales and marketing teams can be useful.

As a staff architect, making adjustments to the governance model can be used to insinuate architects into projects, or, as a consulting architect you can be part of a project team in a role other than architect but provide architect-level input.

Once you are a part of the process, you have to show value to the executive sponsor and other project stakeholders. As you show value you will be retained as a valued member of the team and that leads to extending. Ideally, you want other stakeholders you have worked with to recommend having an architect involved in projects because of the value they provide. This requires not only providing value, but also having a predictable, repeatable approach to creating architecture that is shared by all architects in an organization. A word of caution, there are three important issues that must be addressed regarding customer relationships. We should not treat stakeholders like customers but do leverage the skills of customer relationship to manage interactions with stakeholders.

Setting up a service/customer relationship builds a mental wall of mistrust and an "us" vs "they" attitude. For staff architects, you must avoid this. For consulting architects, you must minimize this.

REFLECTION POINTS

Do Architects need to acquire, retain, and extend customers?

Do you have an "epic fail" related to human dynamics in your past?

3.5 - LEADERSHIP AND MANAGEMENT

Leadership and management are two different things. Most architects are individual contributors and do not have any direct reposts, though in some organizations the architects are managers of the delivery team.

Managers have direct authority and can leverage that authority to protect the intent of the architecture and keep a project on tack. Architects typically use leadership techniques and skills to persuade stakeholders and team members to follow their direction. With technologists that requires having the technical skills to hold a relevant conversation with them and explain why you are correct.

Leadership and management is defined as the ability to manage the unique challenges of leading and managing in a technical environment by leveraging basic management theory, techniques, and tools, and applying within the context of the lifecycle of IT products and services.

As an IT architect, you are expected to demonstrate understanding of how to assess one's personal leadership skills and evaluate the leadership attributes of others, and ability to form a personal leadership growth plan.

Leadership and Management

- The art of influencing human behavior to accomplish a mission in a manner desired by the leader
- May not be an inborn trait
- Most leaders learn to be leaders
- By taking charge of your skills you WILL become a leader
- However, not everyone will get there

Cost of Not Knowing

The cost of not knowing includes a lack of leadership, no results, loss of real opportunity, struggle in dealing with people, and lack of respect.

Skills Analysis

- I am aware of basic people management principles
- I could apply culture and management techniques during projects
- I regularly use practical management techniques in my day to day specialization
- I coach others in management and culture initiatives

Specialization Knowledge Areas in Leadership and Management are:

Software Architects guide software project teams to successful completion of technology strategy; lead in principles of effective software delivery; manage junior architects in software architecture best practices.

Infrastructure Architects guide infrastructure project teams to successful completion of technology strategy; lead in principles of effective infrastructure delivery; manage junior architects in infrastructure architecture best practices.

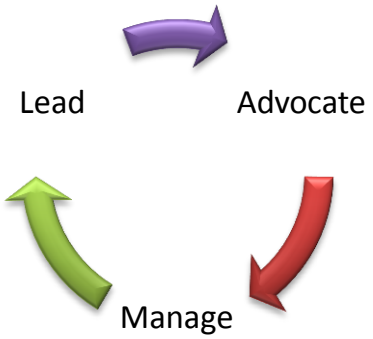
Business Architects guide business project teams to successful completion of technology strategy; lead in principles of effective business delivery; manage junior architects in business architecture best practices.

Information Architects guide information project teams to successful completion of technology strategy; lead in principles of effective information delivery; manage junior architects in information architecture best practices.

As an architect, you must be a champion for the architecture and also be a thought leader. To be a champion for the architecture, you must believe it is the best approach to take and the tools and processes the best fit. You then have to use your influence to influence others to buy into your vision.

As a thought leader, you must get people to see thing in a new and different perspective. That may be introducing an agile method into a waterfall shop, or a different testing approach or technology stack. As you do this, consider asking questions that cause people to sees things differently or research new paradigms. Simply feeding them information is not always the most effective way to look at things differently.

The Architect may be looked to for a decision in terms of choosing one technology over another to deliver a project or implement a corporate system or service. We must be able to answer the question: What are the relative strengths and weaknesses of each solution?



As we drive through projects we need to be the advocate for the end user, and that may require us to influence and, at times, cajole different, competing parts of an organization to achieve the right solution for a client.

Finally, while you are not a manager, you must be able to manage a team. This can include leading the implementation effort and managing the day to day execution of a project. While you do not want to over function, you do want the right solution to reach completion.

What makes a good leader?

There are many opinions, books, lectures, blogs and more on this matter. This is not an all-encompassing list but more of the top characteristics and traits all good leaders should or will have. Staying knowledgeable in this area and applying the principals of these traits will only assist you in your career.

Emotional Maturity – This may seem like a no brainer, but leaders usually believe they have to have the answer to everything. Knowing you will not have the answer to everything; you can set your ego aside and focus on creating great architecture and a good team of people to help you accomplish your goals.

Passion – Showing devotion and passion to some may be considered a weakness. But true good leaders know this is one of the keys to their success. They might not have known it in the beginning but eventually realized that great organizations and people naturally follow or want a good leader that has this most important quality.

Great Communicator - Being a great communicator takes time and practice but it is something that can be learned. It also requires being clear about the purpose of your communication: What outcome do you want? What do you want your employee(s) to do? It also takes knowing how to convey your solution to stakeholders in "their language". And it requires that you listen to your team members and understand the difference between what they say and what they mean. This requires a lot more skill than is apparent, because a lot of factors go into what someone says.

Thinks Positively - Inspiring hope in others, even in a difficult situation, is a key leadership trait. Think about the last time you had a leader that had this trait. Did you want to try harder to find a solution to a difficult problem? Did that leader make even the hardest situation a little easier to cope with? Such a leader empowers followers, making them believe that they can accomplish worthwhile but very difficult things. They create energy and positive thinking in the group by being positive without becoming unrealistic.

Integrity and Authentic - An effective leader is "authentic," internally and externally consistent. A good leader has a purpose, operates out of discernible principles, and "stands for" something worthwhile. Integrity and authenticity require self-reflection, the ability to understand oneself honestly, the capacity to assess one's strengths and weaknesses accurately, and acceptance of one's self. In general terms, "what you see is what you get" and there are no doubts as to the leader's intensions.

Think about the few good leaders you know, what makes them good leaders? Do any of these traits fit them? If so, which ones? Encompassing all these traits is rare; we can always improve on any or all of these. But those leaders who have some or all of these are the leaders you want to follow and/or are drawn to.

3.6 - CRISIS MANAGEMENT

Ask any senior architect, it is not if a crisis will occur, it is when and how often. As an architect you must have a strategy for managing crisis and you must get the team into action resolving problems quickly. This is critical for staying on track, but also for displaying your leadership to calm others and instill confidence.

One approach is to identify the biggest issue and start a portion of the team determining a potential solution. Next, work with the rest of the team to prioritize other problems and consider using the "optometrist" method. Is this problem or that problem first priority, then match the top problem with the next problem in the list until you have looked at all of the areas of concern and have a rough prioritization and the dependencies for each.

Finally, you can use a group ranking or vote to build consensus, but this can be quite challenging and is not suggested if you have not fully evolved your leadership skills.

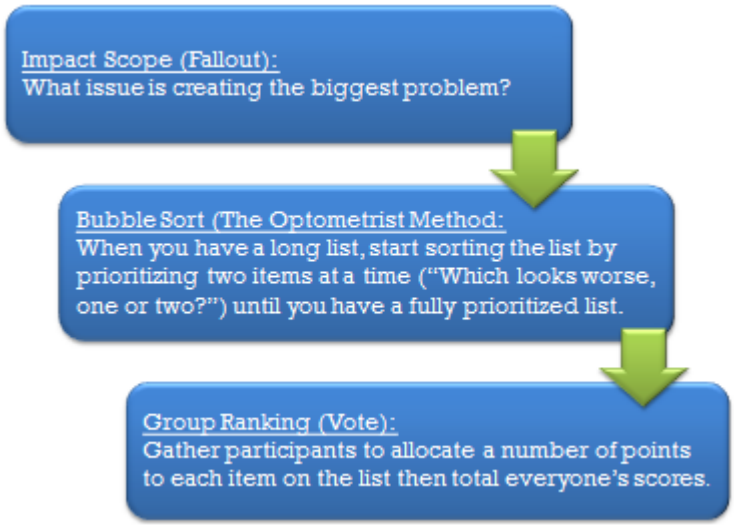


Figure 18

Managing crisis and fixing problems as they occur is important, but making sure you understand why they happened and setting a strategy and approach to keep them from reoccurring on the current projects as well as future projects is critical.

Here is a simple root cause analysis process.

The process is to:

1. Define the problem
2. Gather data/evidence
3. Identify problems that contributed to problem (causal factors)
4. Find root causes for each causal factor
5. Develop solution recommendations

Consider seeking out a six sigma expert to provide additional insights on growing your skills in root case analysis.

3.7 - WRITING SKILLS

Reading about writing skills can help you understand the principles and techniques of strong writing skills, but practice is required in order to become effective with written communications. Fortunately, as an architect you have ample opportunity to grow expertise in writing.

Make sure as you author content, you focus on clarity, brevity, and simplicity, and that spelling and punctuation are correct. Also, with writing you do not have the opportunity to read body language and tailor your message so you must be aware of the tone of your

message and how it might be interpreted. Confidence in the message is diminished when the reader is focusing on grammatical errors or is offended by the tone of the message.

Finally, and most importantly, make sure your pause and reread anything you write prior to sending it.

Writing skills are techniques, methods, and examples for book writing, informal composition, memos and e-mail, note taking and technical documentation.

An architect is expected to understand and demonstrate competence in writing skills necessary to support meaningful communications and artifact creation.

Writing Skills

- Architects must be able to write effectively
- Communication to convey a particular piece of information to a particular audience for a particular purpose
- Concise technical writing can be the difference between success and failure
- Contrary to popular belief working systems are NOT our goal
- Writing can ensure strategy delivery

Cost of Not Knowing

The cost of not knowing includes a lack of leadership, no results, loss of real opportunity, struggle in dealing with people and a lack of respect.

Skills Analysis

- I am aware of ways to improve technical writing skills.
- I could apply my writing skills to create architectural artifacts.
- I regularly use technical writing techniques in my day to day specialization.
- I create templates to help others in my organization create architectural artifacts.

Specialization Knowledge Areas in writing skills are:

Software Architects: map requirements/constraints to component decisions.

Infrastructure Architects: map requirements/constraints to component decisions.

Business Architects: map technology solutions to business objectives.

Information Architects: map information usage to requirements/constraints.

With complex or technical content, make sure you understand the objectives of the document you are creating, and make sure you have the target audience in mind as you

write. The principles described on this slide also come from the instructional design and technical writing toolkit.

Primary examples include:

- Architecture documents
- Executive summaries
- Views/viewpoints
- Explain technology and related ideas to technical and nontechnical audiences
- Gather information from existing documentation and from subject matter experts

As you are creating content, consider a technical writer writing a cake recipe:

Audience: Is the audience composed of people in home kitchens or highly trained chefs in professional kitchens? Source: Is there existing documentation—a rough draft? Who is the subject matter expert (SME)? Deliverable: Is the deliverable simple text for inclusion in a book, or formatted to final form? Is the target a paper, a Web page, or something else?

3.8 - PEER INTERACTION

If you were in a hospital about to have surgery, and the doctor, nurses, and other staff were not all speaking the language of medicine you would lose confidence and they would function at an extremely slow pace. If an ear, nose, and throat specialist and bone and joint specialist spoke to each other and did not have the common language of medicine, imagine the challenges they would face.

As an architect you are expected to be multilingual, speaking the language of business, architecture, and technology. To do this you must grow your acumen in each area while growing your interpersonal and professional skills. Additionally, you must strengthen your skills in coaching and mentoring, being coached and mentored, and work to be sought out for your expertise.

Peer interaction is leveraging the understanding of the psychology of interpersonal human interactions and applying them in the context of IT products and services design and delivery.

Peer Interaction

- You must be able to speak the language of architecture with other architects
- You must be empathetic to the other person
- Listen more than you speak
- Repeat what you hear

Cost of Not Knowing

- The cost of not knowing includes:
- Lack of leadership, no results
- Loss of real opportunity
- Struggle in dealing with people
- Lack of respect

Skills Analysis

- I am aware of the language and communications styles used by all architects.
- I could have a relevant conversation with any architect, regardless of specialization.
- I regularly interact with architects having different specializations.
- I teach others architectural concepts, frameworks, and methodologies.

All Specialization Knowledge Areas

- Language of other architect specializations
- High level understanding of other architect specializations
- Artifacts used to transfer information

3.9 – COLLABORATION AND NEGOTIATION

Definition

Collaboration and negotiations are leveraging knowledge of the psychology of human collaboration, networking, strategies and methods to facilitate working together and reaching agreement.

As an architect you are expected to be able to apply communications theory and the specific collaboration and negotiation skills essential to be effective.

Collaboration and Negotiation

- Improved cooperation among team members is directly proportional to the level of trust among team members.
- Architecture Review Board and the Architecture Governance can reduce conflict.

Cost of Not Knowing

The cost of not knowing includes:

- A lack of leadership and no results
- Loss of real opportunity
- Struggle in dealing with people
- Lack of respect

Skills Analysis

- I am aware of collaboration and negotiation techniques.
- I could apply collaboration and negotiation skills with other team members and stakeholders during a project.
- I have successfully driven projects in politically charged environments.
- I mentor others in this area.

Specialization Knowledge Areas in writing skills are:

All Architects: Executives, business owner, business analyst, enterprise architect

Software Architects: Lead designer, development team

Infrastructure Architecture: Operations manager, network, systems engineers

Business Architects: Business analyst, office of the CFO, legal organization

Concept:

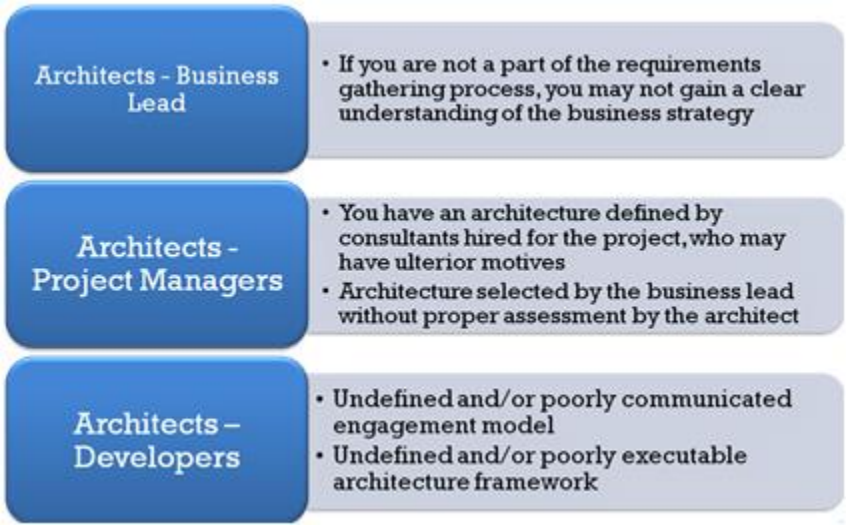


Figure 19

3.10 – MANAGING THE CULTURE

Definition

Managing the culture can be the ethnical culture of a global organization, a technical bias based on the level of comfort and experience with different product stacks or methodologies, or the culture of being risk averse or slow to enact change. As an architect you must balance between matching your approaches to the culture and moving the various biases in play to introduce change.

Managing the culture is affected by many factors:

- Inter and Intra-team friendships
- Physical barriers
- Methodologies
- Tools such as email vs. surface mail and wiki vs. bulletin board
- Top performers have mastered the art of office politics
- There's no such thing as free speech; you will pay for every word you say. Everything you blog, write, Twitter, Facebook etc. The Internet may remember longer than you would like it to!

Cost of Not Knowing

- Inability to connect with teams
- Inability to implement technology strategy or get buy off
- Poor delivery against goals

Skills Analysis

1. I am aware of cultural differences that may be faced on large projects.

2. I could manage an IT project with multiple cultural influences.

3. I regularly work on projects represented by multiple, diverse cultures.

4. I mentor others in managing cultural dynamics.

Concept:

Review a few key concepts around Managing the Culture. Many times finding a mentor that you feel has strong soft skills that fit your personality and asking them to mentor you (either formally or informally) can be the best approach to growing these skills. As always, finding good books on the subject is a good approach. On the Iasa web site and the Forrester web site are lists of books recommended books covering these skills, and setting a goal of reading one leadership or management book per month is recommended.

1. Know that you have a choice
    a. No matter how bad a situation is, you can choose how you let it affect you and how you react

2. Know what your objective is
    a. Don't focus on immediate differences, seek to satisfy the long term business objective
3. Focus on your circle of influence
    a. Don't complain (at work) about things over which you have no influence--it won't get you anywhere
    b. This is different than offering constructive criticism if you feel you can influence a situation
    c. Focus on places where you CAN effect change and have influence
4. Don't take sides
    a. Focus on business objectives and not differences between your colleagues
    b. Open communication prevents an opportunity to say "I didn't say that"
5. Don't get personal
    a. There will come a time where you are disappointed and/or angry with a colleague and will feel the urge to lash out…Don't!
    b. People remember when they are humiliated and/or insulted
    c. You'll pay the price later
    d. Understand others before trying to have them understand you: Seeking to understand others disarms them and lets them know that you value them and their opinion
        i. Sets the stage for open communication with a solution everybody can accept
6. Think Win-Win: It sounds cliché, but it works
    a. This requires you to understand what the other person needs to achieve
    b. If everybody gets something out of a solution, they can really "buy in" instead of paying "lip service"
    c. Builds relationships and allies for the future

Cultural Differences:

The world is flattening and architects must be aware of cultural differences. When you are face-to-face you can quickly ascertain when you have done something that may be considered offensive or in bad taste, and you can recover. When you interact via email, conference calls, and virtual meetings, it can be more difficult to recognize and recover from a faux pas.

Other, common, cultural differences that you will encounter in the business world:

- Different perceptions of what maintaining eye contact means
- Different perceptions of what a firm hand shake means
- Different traditions with respect to deference to/willingness to question authority

The ability of a person to communicate in the local language of business will directly affect how willing they are to communicate:

– For example : A person who is supposed to write detailed test plans in French may not be willing/able to do so if their ability to communicate in French and/or available vocabulary is limited to social conversation

Based on you own experience you may use hand gestures that are perfectly acceptable in the environments you have been exposed to. As you are working with new team members from different geographies, be careful what how you express things.

There are many cultural differences with respect to non-verbal communication.

For example, this sign is considered to be positive ("ok") in some cultures while in others has quite the opposite meaning. In some cultures it is negative ("zero").

The same is true of this sign.

In some cultures (or sub-cultures) it variously means the same as "the finger."

REFLECTION POINTS

As you create architecture, you are working on a specific project that will be introduced into an operating IT environment. There will also be other projects in flight that will be introduced before and after your project is delivered. How would you discover what other projects are in flight and what dependencies and risks they might introduce to your project? What about reuse of capability they are creating?

In smaller IT environments several roles may be consolidated and you may know all of the key stakeholders and members of the architecture team. In larger environments, you may

not have ever met nor have direct access to the stakeholders or other architect teams. How would you introduce yourself to others and gain strive to gain their respect and support?

As you speak with other architects that may specialize is business, security, operations, web portal solutions, SOA solutions, infrastructure solutions, or data center management, you will need to be able to speak a common language and have common tools. You will also have to show respect for their skills and abilities and strive to communicate in their language, while helping them understand the language of your specialization. Do you have an approach for sharing information and building a common language?

Managing the culture will be critical as well. How would you try to introduce an agile methodology into an environment that primarily uses a waterfall approach? If the organization is risk averse and you want to try a new approach to an old problem, how do you socialize your ideas? How do you identify and collaborate with the people that have the political capital to help you?

Human dynamics are the communication, political and leadership skills necessary for any IT Architect. You will likely do a lot of presentations, in meetings and speaking at conferences.

As we drive through projects we need to be the advocate for the end user, and that may require us to influence and, at times, cajole.

You are expected to be multilingual; speaking the language of business, architecture, and technology.

CHAPTER 4: IT ENVIRONMENT

4.1 - OVERVIEW

The IT Environment section of the Iasa skills taxonomy describes the functional and procedural aspects of a technology organization and includes the structures and processes fundamental to an architect's role in an organization. This area is about how the architect works as part of the greater technology organization. Fundamentals include organizational dynamics, roles and responsibilities, elements related to interacting with the engineering and operational elements of the organization, project management concerns, and governance issues.

Architects must know a lot about a lot of different subjects. Al l of us has areas that we enjoy researching and areas we do not. But, just as we have to balance projects that provide new capability with refactoring and consolidation projects, we have to balance our growth between topics we are passionate about and topics we would just as soon leave to someone else.

To have a strong enough understanding of the IT environment to communicate well and make the best decisions you will need to know the technical side of IT as well as the operational side. On the technical front, you must have a broad understanding of the infrastructure, operations and support services, as well as application development. For operations, you must understand and be empathetic to the people that use your solutions as well as the people that support your solutions.

How does an IT shop work? What are the major groups and divisions? Are there better ways to organize, and who created the organization currently used? If you were creating one for a start-up organization, how would you set up an application development or engineering group?

While you may not ever have to make the decisions for setting up or reorganizing an IT shop, you should understand how and why decisions are made. Consider researching the environment you are currently engage with and discover why and how they came to be organized the way they are.

Results from the survey suggest that strong understanding and expertise in the IT Environment pillar are not critical to architects focused in their specialization area, especially software architects. While Platforms and Frameworks rated relatively high, IT Operations and Software Asset Management are abysmal.

Root Cause Analysis on failed projects indicates lack of skills and knowledge about the IT Environment and specifically IT Operations contributes to failure. In the future state of IT architecture practice, all architects will have knowledge of these areas.

4.2 - PLATFORMS AND FRAMEWORKS

Are you aware of the categories of IT frameworks? Do you monitor trends/changes to major frameworks and do you know what the current version of TOGAF is? As an architect you must understand the different classifications of frameworks used in IT. You must also be able to discern the architectural qualities of different platforms, understand some of the major challenges you will have to overcome when re-platforming, refactoring, or redesigning, and be capable when considering systems management and processes integration. Having broad exposure to various platforms and frameworks is critical for some of the reasons shown below.

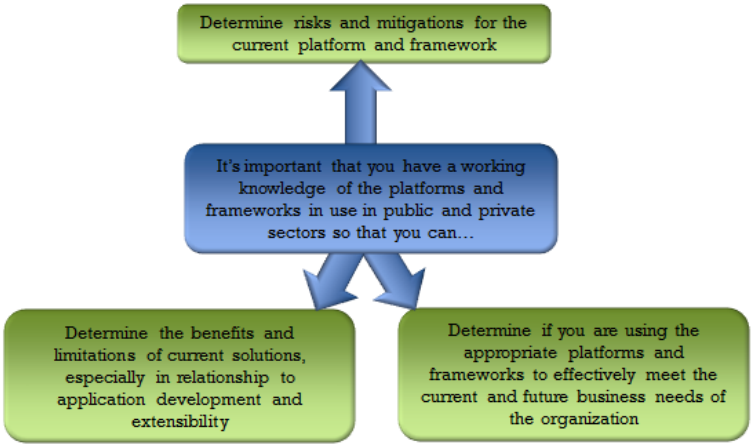


Figure 20

Depending on context, the terms platform and framework have different meanings. In this case, frameworks are the tools employed to provide predictable, repeatable results that include methodologies. For platforms, we are describing the runtime environments that

architects use to develop and deploy solutions. The most abstract frameworks may be considered the enterprise architecture frameworks which include Zachman, TOGAF, MODAF, and FEAF, while the most tactical may be the operational frameworks such as ITIL and COBIT. Examples of platform include mainframe, UNIX, and Wintel, and on the application development side Java and .Net.

While you do not need expertise in all of these, you do need to have pockets of expertise as well as some awareness and understanding of all of them.

Definition:

An understanding of how platforms and frameworks relate to each other, and able to evaluate frameworks against each other in specific and different contexts.

An architect is expected to have a working knowledge of primary platforms and frameworks used throughout the public and private sectors, with emphasis on the architectural qualities rather than the specifics of the platform platforms.

Cost of Not Knowing:

- No tools implies no architecture
- Massive IT budget spend on maintenance

Skills Analysis:

- I am aware of the categories of IT frameworks.
- I keep regular tabs on platform and framework trends.
- I have successfully introduced a framework that MADE US MONEY.
- I lead the adoption process for product and frameworks.

Concepts:

A technical platform or framework is a generic technical base which allows simpler or more abstract elements to be deployed within it. A framework provides an API or interface to which one may attach specific functional elements.

Frameworks are a key element to architectural practice

- They allow us to do more while creating less
- They are equally applicable to hardware, software, information or business elements

Frameworks are essential to delivery of two major value elements:

- Cost reduction,
- new capability

Frameworks often pose significant risk to architecture due to

- Over-generalization, a dedication or preference for a particular technology or direction

Specialization Knowledge Areas in Platforms and Frameworks are:

Software Architects: Map requirement/constraint to component decisions.

Infrastructure Architects: Map requirement/constraint to component decisions.

Business Architects: Map large scale frameworks to business goals.

Information Architects: Map information usage to requirement/constraint.

REFLECTION POINTS

When is a framework too generic?

How much should a business architect know about platforms and frameworks?

How does awareness of frameworks change across specializations?

How often should an architect update their understanding of current platforms and frameworks?

4.3 – APPLICATION DEVELOPMENT

For application development, regardless of your specialization, you should know the basics:

- Industry trends
- Leaders in the specific application space - and why they are leaders
- Benefits and limitations of various methodologies and technologies
- Methodologies and technologies currently in use (and why)
- How to gage supportability, impacts on operations, etc.

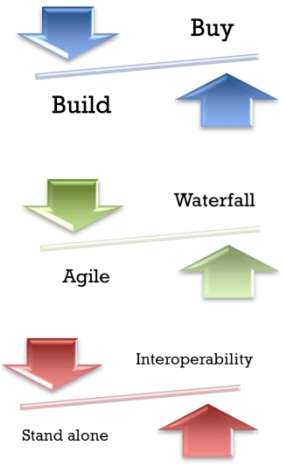


Figure 21

These include tradeoffs between building a custom solution and buying a Commercial off-the-shelf, or COTS solution. Being able to discern the key differences between a waterfall and agile methodology, and when one might be preferable to the other. In addition, know when it makes sense to have a standalone solution versus a fully integrated, interoperable solution.

Do you know current best practices in testing? Do you think load testing is important? Should your organization or the organizations you provide consulting services for start using Test-Driven Development, or TDD best practices?

As an architect you will work with test leads and possibly members of the test team. Being able to have a relevant conversation or answer questions they may have will be critical to gaining their confidence in your decisions.

To drive adoption of appropriate test methodologies, tools, and techniques, you must be familiar with:

- Best practices in testing theory
- Benefits and limitations of common testing practices
- Benefits and limitations of available and in-use testing tools.
- How development methods impact testing
- Organizational attitudes toward and funding of test groups.

Definition:

Application development is defined as the preliminary analysis, software development, testing, deployment, and overall managing of the development environment and process.

Architects are expected to be familiar or experienced with different ways to approach structuring projects including waterfall, iterative and incremental, different levels of formality (from methods with a high degree of ceremony to agile approaches). Some tools in this regard are: IDE, application servers, Web servers, modeling and UML, compiler, etc. As an architect, you are not required to know all of these but having a go to tool that you or your company has defined as the tool of choice is acceptable.

Cost of Not Knowing:

- No developer will ever take you seriously.
- Your strategies will be based on, well, thin air.

Skills Analysis:

- I have studied the major aspects of application development.
- I have studied application development in detail.
- I regularly participate in application development in my organization.
- I lead application development architecture in my company.

Specialization Knowledge Areas in Application development are:

Software Architect: In-depth code design and development alongside development team.

Infrastructure Architect: Interaction with software architect and development team, script development, accurate constraint analysis.

Business Architect: Awareness of current application development practices and limits.

Information Architect: Interaction with software architecture and development team for data, information and usability.

REFLECTION POINTS

Should all architects be required to code?

When is the last time an architect should have written code?

How does that impact specializations?

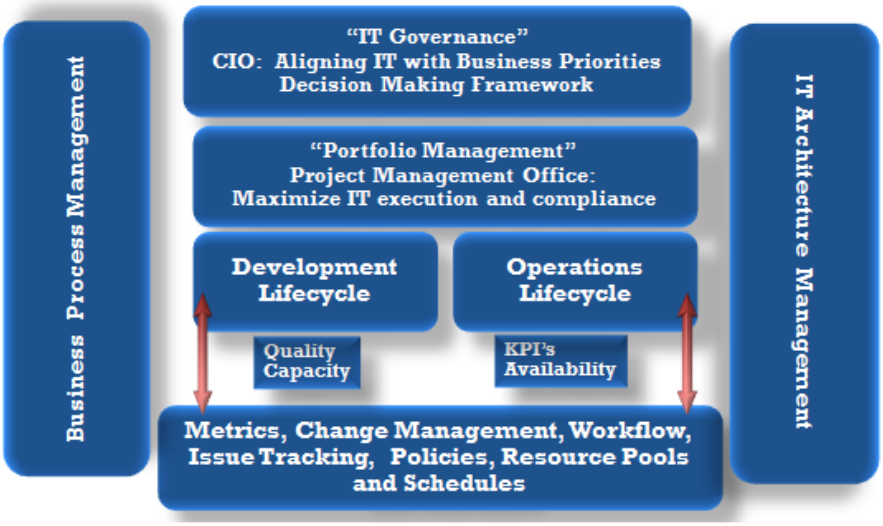
What are the key aspects of application development that all architects are required to understand?

For architects focused on the software specialization, most would likely feel passionately all architects should. For architects focused on the business, information, or infrastructure, maybe the answer would be different. What about you?

4.4 – IT GOVERNANCE MODEL

If you were speaking with a CIO, CTO, or enterprise-level architects, would you be able to describe the difference between Gartner's older governance model and the most recent one? Would you be able to quickly gain insight into the key changes if you have a meeting tomorrow and knew it may come up?

Knowing the basics of IT governance models and current trends is critical to have relevant conversations with your peers and executive sponsors. There are building blocks and related approaches to all IT governance models. Understanding them will allow you to identify the nuance and quickly come up-to-speed with specific ones. Below is an example of one typical IT Governance model.



Definition

The decision-making environment for IT within an organization that provides clearly defined roles and responsibilities relative to oversight of projects, processes, and products.

Architects are expected to show competence in designing solutions that achieve regulatory goals and objectives and allow for guidance and oversight that continuously track to the needs of the business.

Governance

- Governance is necessary to ensure delivery against plan
- A technology strategy is only as good as what is delivered
- Architects attach to the existing corporate governance mechanisms
- We are responsible for the COST and VALUE of technology delivered
- Governance defines who, when and how by:
  - Setting up governance bodies which define policies, standards and guidelines regarding a variety of what subjects
  - Monitoring and enforcing these standards throughout Enterprise

Cost of Not Knowing

- Inconsistent delivery of architecture
- Segmented architecture practices
- Lack of support for business value

Skills Analysis

- I understand the governance model used in my work environment.
- I know what portions of a project would be influence by a governance model.
- I adhere and champion my organization's governance mechanisms.
- I helped develop or maintain our company's governance strategy.

Concepts

- Governance relates to decisions that define expectations, grant power, or verify performance.
- It consists either of a separate process or of a specific part of management or leadership processes.
- IT Governance is a subset discipline of Corporate Governance focused on information technology (IT) systems and their performance and risk management
- "A board needs to understand the overall architecture of its company's IT applications portfolio"
- Primary goals for information technology governance are to (1) assure that the investments in IT generate business value, and (2) mitigate the risks that are associated with IT.

Specialization Knowledge Areas in IT Governance are:

Software Architect -Ensure projects meet governance constraints, actively involved in governance process during project lifecycles, rotate into governance positions.

Infrastructure Architect -Ensure projects meet governance constraints, actively involved in governance process during project lifecycles, rotate into governance positions.

Business Architect - Actively involved with projects to provide governance feedback, structure strategy to ensure delivery, review projects with architecture teams.

Information Architect - Ensure projects meet governance constraints, actively involved in governance process during project lifecycles, rotate into governance positions.

REFLECTION POINTS

How do architects participate in governance?

How much responsibility and accountability should an architect accept?

Are architects liable for their decisions? How does this differ from IT management?

4.5 – INFRASTRUCTURE SERVERS AND NETWORKS

Definition:

An IT infrastructure consists of facilities, network, storage, servers, middleware, and the application software relied upon by the organization's applications, along with the tools, processes, and procedures to build, deploy, install, manage, backup, and restore capability in any of these. In other words, an infrastructure is the foundation upon which application services are delivered, and the operations that keep them running.

All architects are expected to be familiar or have experience with operations, network engineering, server sizing, storage management, backup and restore technologies, and physical data center design.

Understanding your data center environment can save you millions.

It covers operation processes and optimization, device provisioning, network, data center services, messaging, power consumption & deployment. There are tools for each of the areas, for example, ITIL and, COBIT.

Cost of Knowing:

- Poor IT infrastructure expenses and services
- Breakdown between development and operations
- Lack of credibility with data center staff

Skills Analysis:

- I am aware of ITIL concepts and frameworks.
- I have studied detailed operations processes & can relate them.
- I regularly lead the operations and deployment components on projects.
- I have designed and delivered our corporate operations strategy.

Specialization Knowledge Areas in Infrastructure are:

Infrastructure includes everything it takes to run the data center and every device in the enterprise:

- Networking, data center design, device provisioning
- All specializations must be aware of core infrastructure principles

Infrastructure can be managed as a series of components and interfaces in much the same ways as applications and software.

Software Architect: Understand limits and activities of data center and operations; interact effectively with infrastructure planning and architecture.

Infrastructure Architect: Manage data center, operations technology strategy, in-depth and leadership understanding of infrastructure environments.

Business Architect: Awareness of latest infrastructure best practices and interaction with infrastructure architects for strategy planning and delivery.

Information Architect: Awareness of latest infrastructure best practices and interaction with infrastructure architects for information project delivery.

REFLECTION POINTS

Should all architects have worked in the data center?

When is the last time an architect should have designed or maintained a data center?

Are there any major trends in architecture and IT that could impact the data center?

How does that change based on employer (service, vendor, corporate)?

For architects focused on the infrastructure specialization, most would likely feel passionately that all architects should have this background. For architects focused on the business, information, or software, maybe the answer would be different. What about you?

4.6 – OPERATIONS

Definition

IT operations include the management, implementation, execution, and maintenance to provide day to day support to the IT environment.

Resource Operations

- IT resource Operations maintain knowledge of resources (application, system and network infrastructures)
- responsible for managing all these resources (e.g. networks, IT systems, servers, applications etc.) utilized to deliver and support IT services
- includes all functionalities responsible for the direct management of all such resources (network elements, computers, servers, applications, etc.) utilized within the enterprise.
- purpose of these processes is to ensure that infrastructure runs smoothly to support and deliver IT services.
- IT Resource Operations are normally behind the scene in that it supports IT infrastructure, system and applications, and does not directly interact with business users.

Service Operations

- IT services operations focuses on the knowledge of services (application, information, business transaction, etc.) and focus is on service delivery and management as opposed to the management of the underlying network and information technology.
- These functions are closely connected with the day-to-day customer experience.
- IT services operations are accountable to meet, at a minimum, targets set for Service Quality, including process performance and customer satisfaction at a service level, as well as Service Cost.

Cost of Not Knowing

- Inability to understand IT context and work operations teams
- Low operations accountability and high IT costs
- Infrastructure often represents largest line cost and needs optimization
- Low service delivery often leads to poor customer service ratings

Skills Analysis

- I understand the major knowledge areas of operations including the standards and frameworks.
- I have studied operations in detail
- I regularly participate in operations optimization and organization inside of my company
- I have led the development of operations standards, teams and frameworks in my company.

Specialized Knowledge Areas in Operations are:

Software Architects - Delivery of projects into operating environment. Impact of operations on support and quality attributes.

Infrastructure Architects - Direct control and design of operations architectural elements and processes.

Business Architects - Interaction for portfolio of applications, processes and technology. Situational awareness of partner impact and business costs.

Information Architect - Understanding of information in operations that impacts projects and communications. Awareness of operational quality attributes.

REFLECTION POINTS

Should all architects be required to work in an operations team?

What is the cost of not understanding the deployed technology stack, update and maintenance costs?

How should architects interact with the operations teams?

What aspects of operations especially in customer interactions directly impact architects?

4.7 – TECHNICAL PROJECT MANAGEMENT

Definition

Understanding of the factors involved in estimating solutions and their cost, as well as cost tracking methods and tactics for IT projects or programs.

Architects are expected to understand the existing deployed solutions (and the limitations of those solutions), and leverage that knowledge for developing the future state capabilities

required by new solutions, while balancing the cost of the new solution against the overall goals and timelines of the organization.

Technical Project Management

- Does an IT Architect = a technical PM?
- We do need some of the skills that help in project
- Scoping, Resource allocation, tracking, capability mix, use cases, OOAD, dependency identification and timeline

Tools:

- Microsoft Project, Primavera, etc.
- Project websites, reports, etc.

Cost of Not Knowing

- Ever seen a failed project? You will!
- Misaligned resources cost overrun and late project delivery.

Skills Analysis

- I am familiar with major project management concepts.
- I have trained as a project manager.
- I regularly review PM deliverables and contribute across the business.
- I have created PM to architect best practice for my organization.

Concepts

- Two of the key differences between technical project management in IT and elsewhere are;
  - the flexibility of the materials and
  - the goal of replacing human knowledge workers (as opposed to manual processes)
- Estimation for architects involves an evaluation of
  - The potential technology impact
  - The scope of the business impact
  - The estimated number of use cases/user stories (we often estimate prior)
- Project management and IT architecture contain natural tensions which should benefit the organization
  - Architects are responsible for the success of the technology strategy / PM's are responsible for on time/on budget.
  - Architects generally must live with the project over multiple versions while a PM potentially moves on to other projects.

Specialization Knowledge Areas in Technical Project Management are:

Software Architect - Estimate project effort and costs, lead stakeholders to effective technology strategy decisions.

Infrastructure Architecture - Estimate project effort and costs, lead stakeholders to effective technology strategy decisions.

Business Architecture - Set course-grained program and portfolio direction with line of business and capability stakeholders, lead strategy estimation at program and portfolio levels.

Information Architecture -Develop project and program estimation efforts for data, information and usability projects, lead project teams in information areas.

REFLECTION POINTS

Who is responsible for a project?

What success metrics should be used to evaluate a project?

When is a project over?

If the project fails, who is responsible? Who owns the decision for changing the scope of the project? What if it impacts the intent of the architecture and does not deliver the expected business value?

4.8 – CHANGE MANAGEMENT

As an architect, you should understand the ITIL or other commonly used Change Management concepts, be able to leverage or drive Change Management adoption or an upgrade process. The successful architect has likely worked on large transformational projects with emphasis on Change Management, and certainly understands the importance of known state, and is aware of and suffered the consequences of unplanned change.

Knowledge of Change Implementation Planning and Management, Transformation Management, and Information Systems Coordination and Control are essential to the IT architect.

Definition

Management of any change to the IT environment, including release management, build, configuration management, device management and proper documentation on change request.

Architects are expected to have a deep understanding of the vital nature of designing and utilizing effective change management processes and the critical role that change control plays in a quality operational design.

Change Management

- Change management is more than just version control
- Most companies do versioning only
- It covers release management, build, configuration management, device management and proper documentation on change request

    - Unfreeze, change, refreeze, branching, etc.
- Tools:
    - SCM (Software Configuration Management) tools, migration tool, build tool, etc.

Cost of Not Knowing

- Deployment costs escalating
- Breakdown between development and operations
- Unmanaged releases and version

Skills Analysis

- I have reviewed the ITIL or other standard CM concepts.
- I regularly work with CM products beyond versioning.
- I have led CM adoptions or upgrade process.
- I lead our companies CM processes.

Specialization Knowledge Areas:

All Architects - Create detailed change management recommendations for active projects based on business value and risk strategy; Recommend effective change management policies for organization context, scope and engagement model.

REFLECTION POINTS

What elements of the IT environment must be versioned?

How should an architect impact the change management policies?

Which specialization most impacts change management?

When does change management negatively impact the technology strategy?

If you were in a conversation with another architect, would you be able to defend your position? Could you understand theirs?

4.9 - ASSET MANAGEMENT

Do you understand primary concepts in asset management? Have working knowledge of asset management technologies? Have you delivered (or worked with) an asset management solution?

As an architect, you are expected to understand hardware and software lifecycle management, service, warranty, and licensing management, and sustainability management for IT.

Definition

Asset management is the development and deployment of a solution designed to manage the intellectual property of solutions and architectural components within the IT environment. Assets can include document formats, video, audio, configuration information, and any other way that knowledge is stored and transferred. Some tools in Asset Management are:

- Digital Asset Management
- Portals, wikis, blogs
- Document repository

Cost of Not Knowing

- Reinventing architecture and knowledge
- Lack of organizational knowledge
- Individual loss of continuity

Skills Analysis

- I am aware of primary concepts in asset management.
- I have studied asset management technologies.
- I have delivered asset management solutions.
- I lead enterprise level engagements for asset management and instruct others.

Specialization Knowledge Areas in Asset Management are:

Software Architect - Update architecture repository with software architecture documents; Identify key knowledge and business areas impacting technology strategy.

Infrastructure Architect - Update architecture repository with infrastructure architecture documents; Manage data center and operations asset components for architectural development.

Business Architect - Ensure key technology strategy assets are communicated and managed across LOB/capability; Coordinate appropriate division and update of assets impacting key knowledge areas.

Information Architect - Update architecture repository with information architecture documents; Identify key knowledge and business areas impacting technology strategy.

4.10 – TESTING METHODS, TOOLS AND TECHNIQUES

Do you have knowledge of primary testing methods and tools? Do you have In-depth knowledge of testing methods? Do you take responsibility to ensure all deliverables are tested properly? As an architect you are expected to stay current of testing processes, quality management, quality standards, and quality assurance.

Definition

Testing methods, tools and techniques is defined as the identification of testing methods and tools that are optimized for value, providing the appropriate levels of quality without over-burdening project deliverables.

Architects are expected to be able to detail best practices in testing theory, techniques and tools, and are expected to demonstrate competence in the scientific method and important testing techniques.

Testing Methods, Tools, and Techniques

- Ensure testing methods and tools are optimized for value
- Describe appropriate levels of quality without over-burdening project deliverables
- Too much testing is as bad or worse than too little
- You have to test quality attributes too
- Tools: Unit, black box, white box, integration tests

Dev environment testing

- User integration testing
- Automated vs Human

The Cost of Not Knowing

- Deployment costs escalating
- Breakdown between development and operations
- Over-tested (more expensive) code or technology base

Skills Analysis

- I am aware of primary testing methods and tools
- I know when to use what testing on a given project
- I have ensured projects are tested properly
- I lead our architectural inputs and testing processes

Concepts

- Multiple types of testing exist and are the responsibility generally of differing groups
- Unit testing (development)
- User Interface testing (development and QA)
- User Requirement testing, black box/ white box testing (QA)
- Testing environments often duplicate the production environment
- Architects must provide significant input on the design of testing environments
- To ensure value and to evaluate test data structures for evaluation of quality attribute coverage

Specialization Knowledge Areas in Testing Techniques, Tools and Techniques are:

Software Architect - Primary role in setting testing levels based on value for software delivery

Infrastructure Architect - Primary role in setting testing levels based on value for infrastructure delivery

Business Architect - Map test and assurance to business value and champion with stakeholders

Information Architect -Primary role in setting testing levels based on value for information project delivery

REFLECTION POINTS

What are the architect's key responsibilities to the QA group?

How does QA impact an architect's key value metrics?

When is the last time an architect should have designed a testing process?

What is the difference between Test and QA?

How do you evaluate the quality of a system designed to interact with a million devices?

4.11 - PROJECT PRIORITIZATION

Typically, project prioritization is managed by the PMO or a governing committee that may or may not have architects on it, and you may or may not be involved.

As an architect, you should have visibility across the enterprise and know what projects are being planned, and be able to influence others to ensure strategic projects should be rated highly.

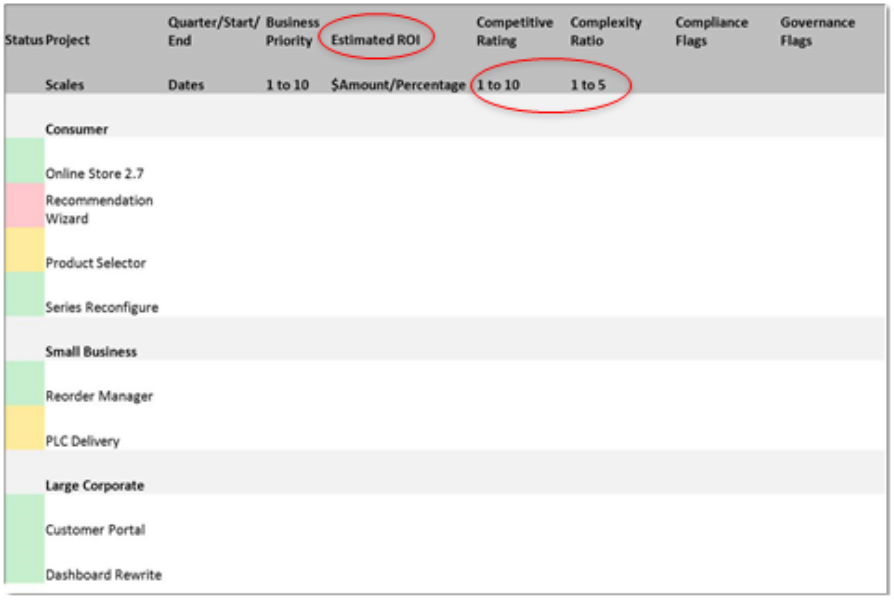
An understanding of the IT environment, how it is governed, how assets and knowledge are maintained, and fundamentals of project management, systems and networking engineering, and application development will aid in making the best choices in project prioritization.

As part of portfolio management, all projects that are planned, and later all that are funded are organized together to provide initial rationalization on what order projects will be started and finished, and budget resources that will be spread across projects.

This type of planning exposes project dependencies and provides a way to determine what additional resources will be needed to complete projects in the given timeframe. However, there is not a lot of information that helps describe the business value, technical complexity, competitive advantage, and governance issues that may need addressing.

By leveraging the skills represented in the IT Environment pillar, architects can add value and influence project prioritization by including additional business-related decision points into the process. Using a tool such as the one pictured below, you can capture not only start

and end dates to budget resources, but can capture information of the business priority for each project. This can be a simple rating system using a 1 to 10 scale or 1 to 5 scale. You can also include information of the Return of Investment (ROI) calculations, with either hard dollars or percentages for each project, and could even add a column for indication how many months after the start of a project the expected returns will start.



Status	Project	Quarter/Start/ End	Business Priority	Estimated ROI	Competitive Rating	Complexity Ratio	Compliance Flags	Governance Flags
	Scales	Dates	1 to 10	$Amount/Percentage	1 to 10	1 to 5		
	Consumer							
	Online Store 2.7							
	Recommendation Wizard							
	Product Selector							
	Series Reconfigure							
	Small Business							
	Reorder Manager							
	PLC Delivery							
	Large Corporate							
	Customer Portal							
	Dashboard Rewrite							

Ranking each project by the competitive advantage the project will provide can provide additional value. For instance, ratings from 1 through 3 could be used for projects that help your organization catch up to a competitor while ratings of 4 through 10 could be used for projects that provide business advantage.

Complexity rating could be used to describe the risk of failure and advanced resource requirements, while compliance and governance flags could be used to indicate the additional time required to legal evaluation or approval for an exception to the governance polices.

When added up, this could change the overall prioritization of projects and ultimately provide additional realized business value.

Most architects tend to have 6 to 10 projects that they are responsible for. In some organizations architects are used initially but do not provide oversight during development or delivery of a solution and are only involved if a project enter crisis mode.

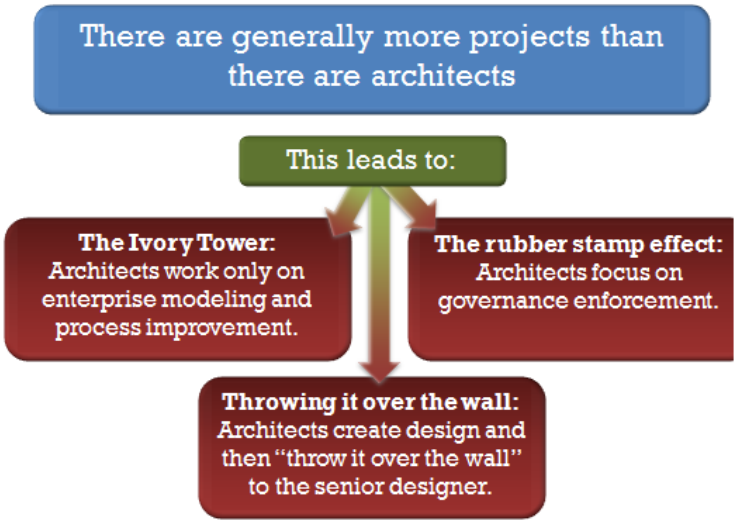


Figure 22

Architectural assignment and oversight of projects must be monitored and there must be a feedback mechanism in place to safeguard against some of the negative outcomes that can occur.

Some projects should have a fulltime architect assigned. Reasons for assigning a resource fulltime may be due to complexity of a solution, risk to the overall business should the project fail, or to gain positive visibility for the architecture team.

As an architect, you must decide what decisions are architecturally relevant. Additionally, you must determine how much of your time is needed when. For instance, in a project using a waterfall methodology architects are required fulltime in the beginning of the project and less as the project enters development. Using an agile approach, the workload of the architect is spread over the course of the project (Iasa offers an Agile courses, see www.Iasaglobal.org for more information).

4.12 - HOW TO SELECT PROJECTS

Not all projects require an architectural resource. Can you think of solutions that have well known patterns or do not introduce a high amount of risk or complexity? As a consulting architect or a staff architect, you want to make sure that there is some reasoning used to assigning architectural resources.

Some reasons to employ architects are technically based due to risk or complexity, while others may be more politically motivated. Having clear goals for assigning architects and understanding what return will be realized is critical.

For projects with no architects assigned, ensure the governance model provides guidance. This might be in the form of milestone review meeting, requesting specific artifacts be created and stored for each of those projects, or providing guidance of technologies, products, testing methods.

Assign architects to high value projects

- Revenue generating
- Expensive
- Political

Key to technology capability growth

- Do not assign to projects within common patterns
- No significant new technology
- Standardized processes
- Minimal architecture complexity or business value
- Use governance mechanisms to handle non-assigned projects

A word of caution, using governance and setting approval gates that allow for architectural review can provide coverage for more projects, but can also generate a negative bias towards the architecture team and cause teams to try and sidestep the governance model.

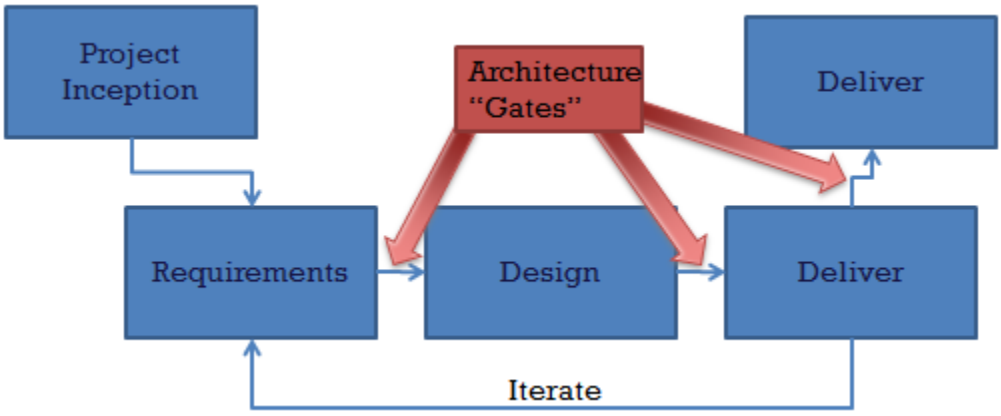


Figure 23

- "Gated" delivery process allows architects to review and approve
- Covers many more projects
- Risks: teams will ultimately side-step or overwhelm reviews and Architecture will become a roadblock—or feel like it

Architects should be concerned with generating the greatest business value possible through the use of IT. The greatest opportunity an architect has to provide business value occurs at the project level. By selecting the best approach, providing a flexible, maintainable solution and designing the right level of quality attribute are some examples.

- A vast majority of the cost savings and revenue generation happens at the project level

- A good goal for an architecture team is to have at least one project with a full time architect to:
  - Demonstrate value
  - Build stakeholder buyoff

In Summary:

The IT Environment pillar describes the skills required to create solutions that will fit into and be manageable by an organization. The specializations will have greater strength with some of the skills, but all architects need basic knowledge in each of these. For instance, a software architect will need greater skill in application development than a business architect, but both need foundational knowledge for the topic. These skills help provide the broad vision needed for success.

CHAPTER 5: QUALITY ATTRIBUTES

5.1 - OVERVIEW

Currently, Quality Attributes go by many names, including non-functional requirements, systemic qualities, and "the -ilities". Iasa supports IEEE and ISO in using the term Quality Attributes.

As an architect, you are expected to work with executive sponsors and other stakeholders to determine what level of which quality attributes are needed. Everyone wants the most, the best, and a solution that is always available. Some may even suggest they need 5 nines uptime without understanding what that means, having read or heard the term. You must help the stakeholders understand the tradeoff between quality attributes, which ones complement each other, and what the long term cost for different levels of quality will be.

Definition

A quality attribute is a non-functional characteristic of a component or a system. It represents a cross-cutting architectural concern for a system or system of systems.

IEEE 1061

The degree to which software possesses a desired combination of quality attributes

ISO/IEC 9126-1

Defined with six categories of characteristics: functionality, reliability, usability, efficiency, maintainability, and portability which are divided into sub-characteristics

Characteristics

- Quality attributes cut across all IT architectural concerns
- Also called systemic qualities, illities, and illiancies
- Example: security, performance, manageability, flexibility, usability, resiliency, availability, etc.

Balance

- Time, cost, requirement, resources constraints can become a trade off in applying quality attributes
- Sometimes we need to make a trade off decision

Warning

- The risk of ignoring quality attributes are many times greater than the cost of effectively managing from the start.

Key Points

- Attributes are good
- Cross cutting
- Pick the ones you need
- Select and trend metrics
- Remember why you are tracking them

IASA Groupings

Usage Related

- Usability
- Localization
- Accessibility
- Personalization
- Customizability

Development Related

- Manageability
- Maintainability
- Supportability
- Extensibility
- Flexibility

Operation Related

- Performance
- Reliability
- Availability
- Scalability
- Security

5.2 - CHARACTERISTICS

Quality attributes provide a way for you to explore the level of service needed with stakeholders.

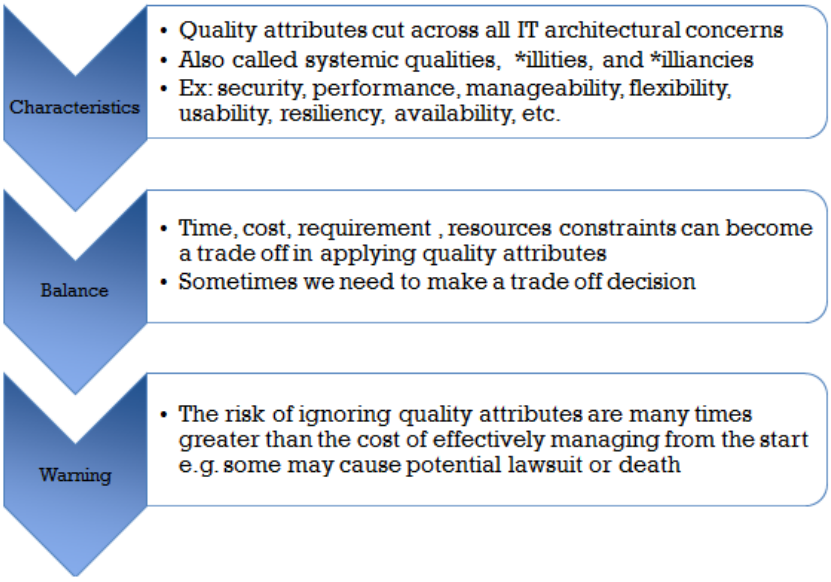


Figure 24

While written in technical terms, you can translate what each means into the business value it will provide. There is considerable expense in providing levels of service beyond what is needed and your job as an architect is to explain the costs associated with each attribute and balance those against the business need.

Additionally, some quality attributes compliment, while others are counter to each other.

Iasa organizes key quality attributes under usage related, support related, operation related, and security. This makes it easier to consider attributes you need to consider in your design and provides a way to discuss tradeoffs with stakeholders. For instance, providing greater personalization will have a negative impact on supportability. Please note, this is a small portion of the quality attributes that are in use in the industry.

For the usage related quality attributes, an architect is expected to understand globalization and localization management, techniques for usability evaluation and impact, and systems ergonomics.

Figure 25

For support related quality attributes you should be skilled in configuration management, operations change management, and service desk and incident management.

For operations related quality attributes, you should understand availability management, capacity management, continuity management, and performance management.

With Security, an architect should be familiar with information management, security administration, safety engineering, and safety assessment.

REFLECTION POINTS

How would you discuss the cost of 4 nines versus 5 nines in business terms over the life of a solution?

List some typical IT Architecture concerns in regard to quality attributes.

How do we measure the "goodness" of IT Architecture?

Choose one of your most recent IT projects. How do you rank the important of such quality attributes given limited resources? (i.e. time and budget constraints). Choose top 5 attributes.

A mistake some architects make is to treat the executive sponsor or other key stakeholders like a customer, accepting what they say they want without exploring what they actually need and providing insight on the costs and alternatives of providing various levels of quality.

As an architect, you deliver business value by partnering to explore what is needed and provide a strategy that provides the most value.

5.3 – USAGE RELATED QUALITY ATTRIBUTES

Are you familiar with globalization and localization management? What is the evolution of usability and its impact on your architecture? What do you know about systems ergonomics? As an architect you are responsible for design a solution with usage related quality attributes that map to the requirements of the end users.

Definition

Usage related quality attributes is the demonstrated understanding of usability and human factor fundamentals, with the ability to describe related internationalization strategies and current issues, as well as demonstrated competence in implementation techniques.

Usage-Related Quality Attributes

- Usage affects every aspect of your success:
    - In value recognition
    - In user involvement
    - In future engagements
- Usability should be evaluated based on value metrics:
    - Productivity
    - Time to complete task
    - Error rates and impact
    - Perception
        - Usage patterns
        - Value recognition
- Warning: Too much focus on usability CAN be a waste, make sure you truly understand the value

Cost of Not Knowing

- Lack of recognition of solution value
- Productivity losses
- Ineffective solution regardless of functionality

Skills Analysis

- I have reviewed the concepts in usage attributes
- I have studied usage attributes in detail
- I regularly manage usage attributes on projects
- I lead and mentor others on usage attributes in our company

Concepts

Usability

If a solution is delivered but never used, it isn't providing value, and you will never know if it is being used or how effectively if you don't design an approach to measuring usage into you design. While end users don't always know what they need or is possible, they are more comfortable with small changes in the solutions you provide than wholesale change.

Localization

As an architect you should understand what elements of a solution change to support localization.

Accessibility

One of the most common oversights in accessibility is not considering people that are colorblind in designing your solution. Over the last two decades innovative solutions have been introduced to help with accessibility concerns. Consider researching current solutions.

5.4 – OPERATION RELATED QUALITY ATTRIBUTES

Are you familiar with how Operations manages and maintains the IT environment? Do you know how to provision a solution for monitoring and what tools are in use? As an architect you should be skilled in availability management, capacity management, continuity management, and performance management.

Definition

Operation related quality attributes includes the demonstrated management/mitigation of issues that relate to performance, reliability, availability, and scalability.

Demonstrated understanding of quality attributes such as portability and efficiency and competence in implementation techniques.

Operation-Related Quality Attributes

- Have a direct customer impact
- Facts about operations Quality Attributes:
    - Performance is expensive
    - Everyone wants more than they need
    - No one gives accurate requirements
- Always get a concurrency and performance count before you create an IT Architecture solution
- Tools and resources:
    - Static code analysis and profilers
    - Performance testing suites

Cost of Not Knowing

- Huge customer impact
- IT becomes a stopper to business operations

Skills Analysis

- I am aware of the elements of enterprise performance
- I know of at least three tools currently available to manage operations
- I have led the adoptions of performance standards
- I regularly mentor teams on performance analysis and delivery

Performance Concept

When exploring performance with end users or other stakeholders, you might get a requirement that is difficult to deliver and not necessary. For instance, you make get a requirement of sub second response time, but what does that mean and is that what was meant when the requirement surfaced?

A good approach to scoping performance is to use examples the stakeholder is familiar with. If they are waiting for a screen refresh in your solution and as for sub second response, ask

if the time it takes for a web browser search to complete is a good measure. If so, time a typical refresh and use that time.

Scalability Concept

As an architect you are not just concerned with does it scale but, but also can it scale and how will it scale. You should be able to describe how many, how much, how fast to increase, how hard, and the cost to increase. Additionally, you should know the scale up and scale out your solution.

Reliability Concept

In the past 5 years have you heard news stories about confidential or private information being exposed or stolen? Reliability of your solution might just allow a person at a desk to do their job, but it may also keep bad things from happening. Knowing how to build in reliability and the ability to test a solution is critical.

Availability Concept

High availability describes systems that exhibit almost no downtime or even the appearance of downtime.

Do you know the difference between 3 nines, 4 nines, and 5 nines in availability? Have you calculated and discussed the cost of offering each level of availability to an executive sponsor? As an architect you should be able to have that relevant discussion, and be able to design a solution where available is well defined and measurable.

Do you know the level of availability offered by most Internet-based solution providers? Do you know how availability is defined? This example shows the classes of availability and what class of machine typically requires what level. Consider having a chart like this available as you discuss availability with stakeholders.

5.5 – SUPPORT RELATED QUALITY ATTRIBUTES

Are you familiar with configuration management? Operations change management? As an architect you should be, and your solutions should be designed to provide the support related quality attributes required for the solution, and the solution should be manageable and maintainable by the operations organization.

Definition

Support related QA is the demonstrated management/mitigation of issues relative to manageability, maintainability, supportability, extensibility. Demonstrated competence in implementation techniques is imperative.

Development Quality Attributes

- Development attributes affect the management and building of solutions
- Reduce the cost to change or operate a solution
- 80% of solution cost after deployment

- Balance usage attributes with need to deliver on time and budget
- Warning: Architects often over-optimize development attributes

Cost of Not Knowing

- Regular re-delivery of solutions
- Maintenance costs escalate
- Overall architecture failure due to inability to manage effectively

Skills Analysis

- I have reviewed the development quality attributes
- I have studied each development quality attribute in detail
- I regularly deliver solutions with optimized development quality attributes
- I lead and mentor inside our company on development quality attributes

Concepts

The ability to gather information about the past and present state of something and to control it, and predict future needs.

Do you know what management tools are used by Operations? Do you understand configuration management techniques they use? As an architect, you must create a solution that allows information to be discovered and gathered, and then used to control the solution.

Specialized Knowledge Areas

Software Architect - Define and deliver development quality attributes for software systems

Infrastructure Architect - Define and deliver development quality attributes for software systems

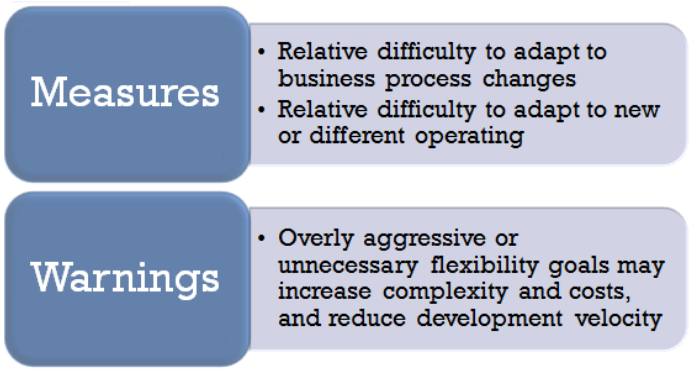
Business Architect – Maps the solution development of quality attributes to line of business or capability and regularly coordinates across projects

Information Architect - Define and deliver development quality attributes for software systems

5.6 – A CLOSER LOOK

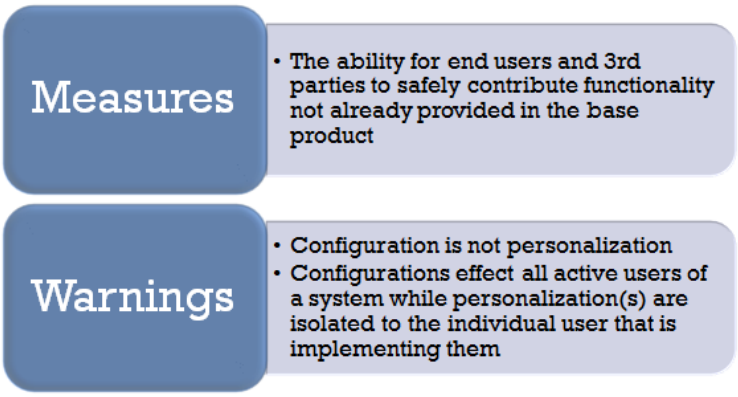
As architects, we should future-proof our architecture to the best of our abilities. While we should only design what is needed, we should be thinking of the future and how we can build a solution that provides the greatest amount of flexibility to adapt in the future.

Flexibility - Flexibility is the ease with which a system can be modified to achieve goals other than those for which it was specifically designed.



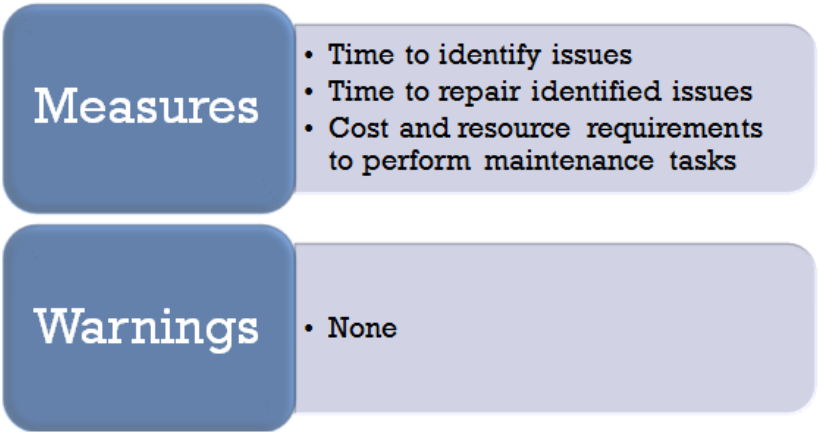
As an architect, we determine what an architecturally relevant concern is. The architecture we create may need to provide support for customizability, in areas that will not impact the integrity of the architecture or cause the solution to be less maintainable.

Customizability - Customizability is both the design-time extension and the run-time modification of a solution.



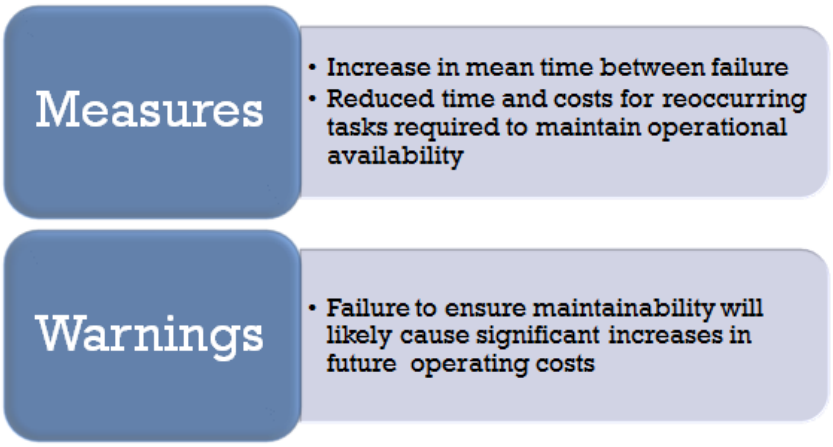
One of the only quality attributes that does not have a warning is supportability. Ultimately, the better our solution can facilitate detection and recovery from anomalies, the greater the business value the solution will provide.

Supportability - Supportability is the inherent quality of a system to facilitate detection, isolation, and timely repair/replacement of system anomalies.



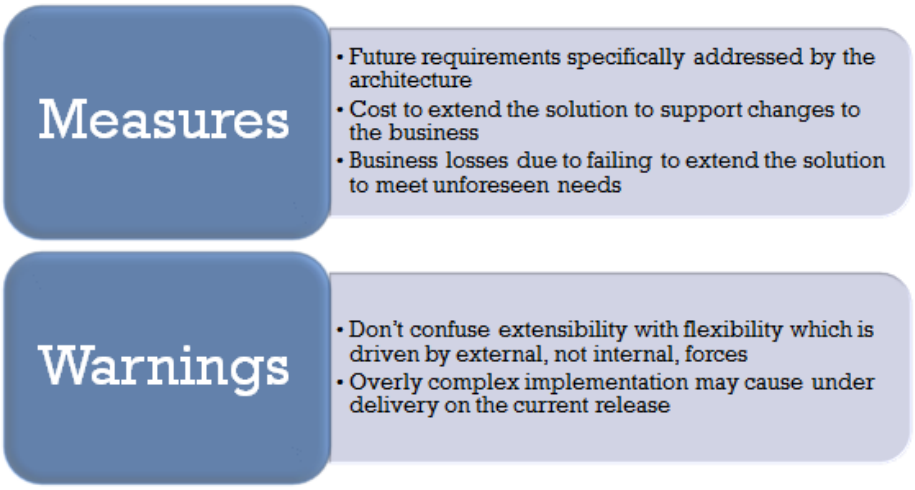
As you enter into a project and are scoping the existing IT environment, you determine the quality and level of staff available to help deliver a solution, and then maintain a solution. To mitigate the negative impact of lower level or less qualified operations teams, or the loss of highly qualified operational staff, designing for maintainability is critical.

Maintainability - Mil spec MIL-HDBK-470A defines maintainability as the relative ease with which an item can be retained in, or restored to, a specified condition.



A mantra for architects is to deliver only what is needed. However, by keeping current of trends in technology and approaches, and communicating with a broad community of architects, you can design solutions that are extensible. Consider the difference between internal and external forces and extensibility.

Extensibility - Extensible architectures both enable the future inclusion and prevents future exclusion of functionality.



5.7 - SECURITY

Definition

Demonstrated understanding of security, privacy, authenticity, access privileges, information protection and disaster recovery, asset management techniques, threat modeling and recovery, and related issues. Demonstrated competence in basic implementation techniques.

Security

- Provides assurance on confidentiality, integrity, and access
- IT security covers a broad field cutting across all aspect of IT projects:
    - Guide technical teams in security implementation
- Develop security strategy to:
    - secure baseline
- Tools and resources:
    - Various securities tools from products to frameworks
    - http://www.itsecurity.com

Cost of Not Knowing

- Monetary loss from hackers
- Poor customer relations
- Regulatory compliance issues

Skills Analysis

- I am aware of basic security principles and concepts
- I stay current on emerging security concerns
- I have used industry standard security components on projects (NOT HTTPS)
- I regularly review and update security strategies for the enterprise

Measures

- Level of effort required to breach trust boundaries
- Likelihood of consistent implementation of policies
- Compliance with industry standards and applicable law

Warnings

Always make security a pervasive, unavoidable component of the solution and the development process--never an afterthought.

Concepts

Physical security, network security, platform security anti-virus, audit logging, monitoring policies and standards

Security conceptual and detailed design

Understand user base, data classification, technology and existing and emerging security standards

Specialization Knowledge Areas

Software Architects:

Develop and understand security requirements and their impact on the software strategy in place ensure security strategies support bottom line business value.

Infrastructure Architects:

Develop and understand security requirements and their impact on the infrastructure strategy in place ensure security strategies support bottom line business value.

Business Architects:

Coordinate technology strategy related to security concerns across line of business or capability.

Information Architects:

Develop and understand security requirements and their impact on the software strategy in place ensure security strategies support bottom line business value.

5.8 - BALANCING QUALITY ATTRIBUTES

Balancing Quality Attributes is about tuning your solution or a family of solutions to deliver the greatest business value. To do this successfully, you should chart the various quality attributes for your solution and identify supportive pairs and pair that have dependencies. Then, you can discuss the tradeoffs and decisions you've made, and create artifacts that capture what was decided and why it was decided, and provide traceability back to the business requirements.

Tuning an Individual Solution

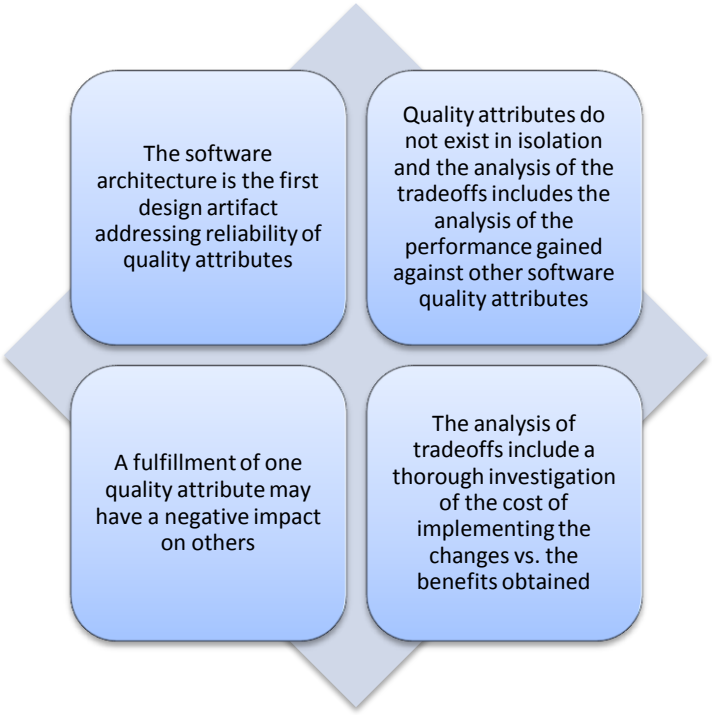


Figure 26

As an architect, you must have some process for balancing quality attributes when creating your architecture. Since quality attributes impact one another, you must analyze the impact of any change you make to quality attributes as you model and evolve your solution. Knowing what other solutions are in the pipeline and how they will impact any shared resources is also critical.

Tuning a Product Family

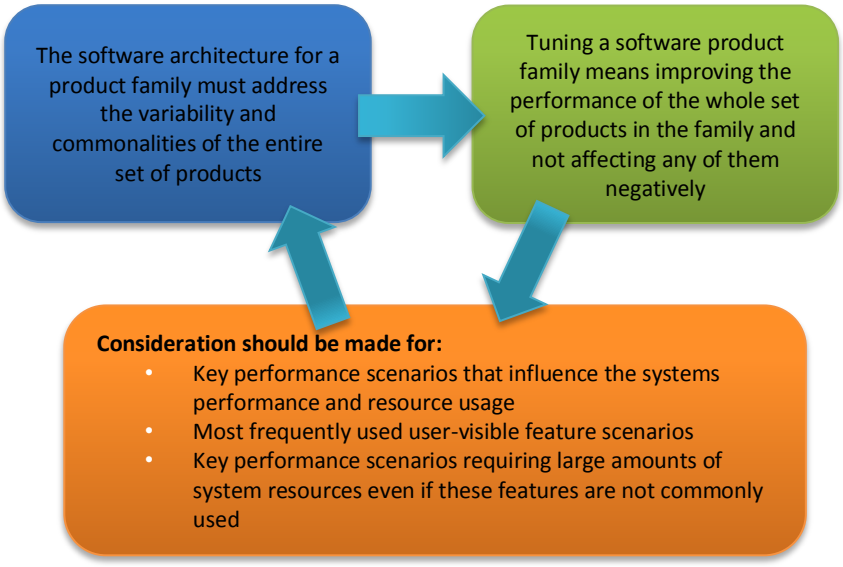


Figure 27

As you balance quality attributes for a family of products or solutions, you must use a different approach.

The software architecture for a product family must address the variability and commonalities of the entire set of products

Tuning a software product family means improving the performance of the whole set of products in the family and not affecting any of them negatively

Consideration should be made for:

- Key performance scenarios that influence the systems performance and resource usage
- Most frequently used user-visible feature scenarios
- Key performance scenarios requiring large amounts of system resources even if these features are not commonly used

Quality Attributes Pairs

	Development					Usability				Operations					
↓ Imapcts →	Flexibility	Personalization	Localization	Maintainability	Extensibility	Reliability	Customizability	Availability	Accessibility	Performance	Scalability	Security	Deployability	Observability	Supportability
Flexibility		+	✓	-			-			-	✓		-		
Personalization	-		+	-	-		+			✓	-		-		-
Localization	✓	+		+	✓		✓		✓						✓
Maintainability			✓			✓		✓		✓	✓	✓	✓	✓	✓
Extensibility			✓			✓		✓	✓	✓	✓	✓	✓	✓	✓
Reliability				✓	✓			✓			✓	✓	✓	✓	✓
Customizability	✓	✓	✓						✓			✓	✓		✓
Availability				✓	✓	✓			✓		✓	✓	✓		✓
Accessibility			✓		✓		✓	✓				✓			✓
Performance	✓			✓	✓	✓					✓	✓	✓		✓
Scalability	✓	-		✓	✓	✓		✓		✓			✓		✓
Security				✓	✓		✓	✓	✓	✓					✓
Deployability	✓	✓		✓	✓	✓	✓	✓		✓	✓				✓
Observability				✓	✓	✓									
Supportability		-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		

Key				
helps	hurts	neutral	Not applicable	
+	-	✓		

Note: This chart is an example of a tool you can create. Impact can vary based on the solution.

Figure 28

Creating a chart with the quality attributes requested and you want to design into your solution and showing the tradeoffs is a good way to make what can be an abstract notion tangible for your stakeholders. It can also be used as a vehicle to work through cost and benefit with the executive sponsor of your solution.

Flexibility and Personalization

This is an example of two quality attributes that support each other. Coupling the previous chart with a document like this helps to have relevant quality attribute conversations with all of your stakeholders.

This chart may help when speaking with the usability team or the end users to describe why you are making certain choices.

Additionally, by showing a chart of the quality attributes you are designing for and then describing the relationships you can more easily tune a solution and capture the critical thoughts behind the decisions you make.

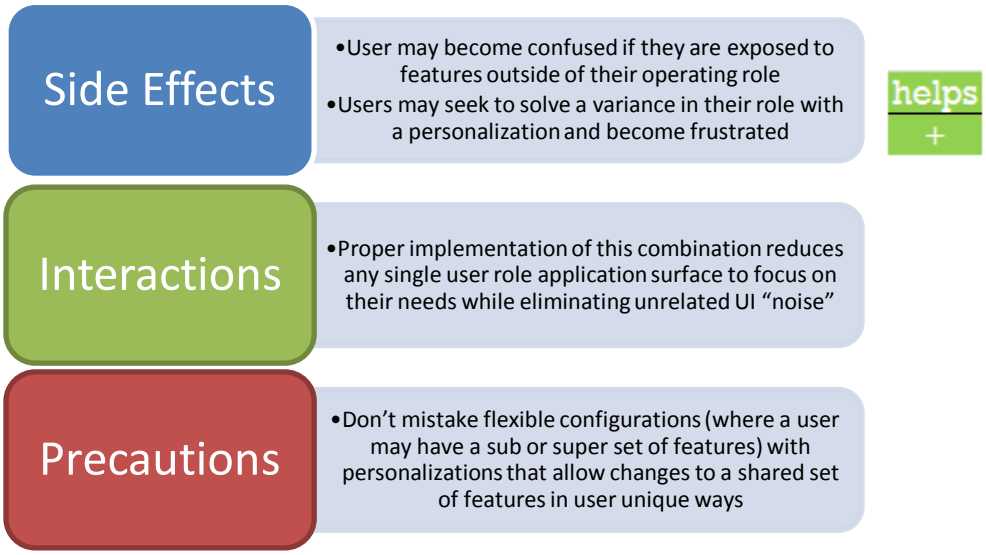


Figure 29

Flexibility and Personalization both directly impact the user

- They are typically combined in larger organizations where one application actually supports multiple user roles, frequently within a single or related group of work flows

Flexibility and Localization

This shows a pair of quality attributes that have little relationship or impact on each other and can help as you are balancing attributes.

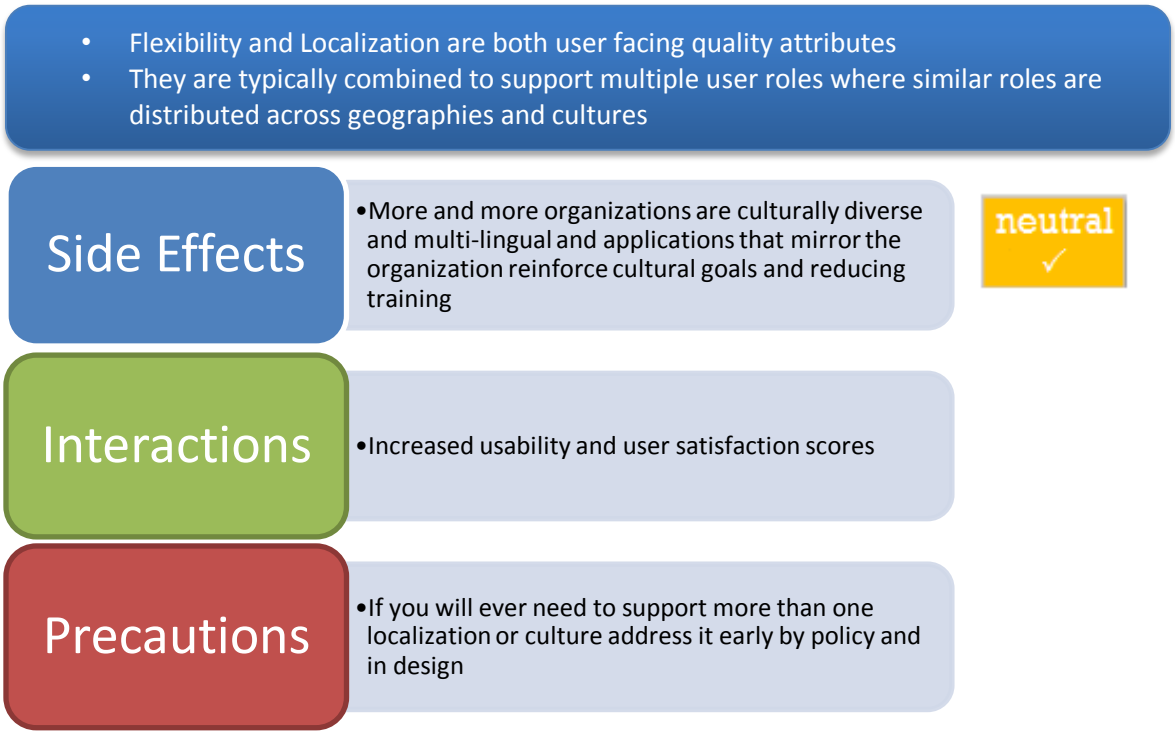


Figure 30

Flexibility and Maintainability

This example shows a tradeoff between two quality attributes. In this case, providing flexibility will make the solution harder to maintain, generating additional operational costs. Using this tool with your sponsor when exploring business requirements will help you to determine what is most important for their organization.

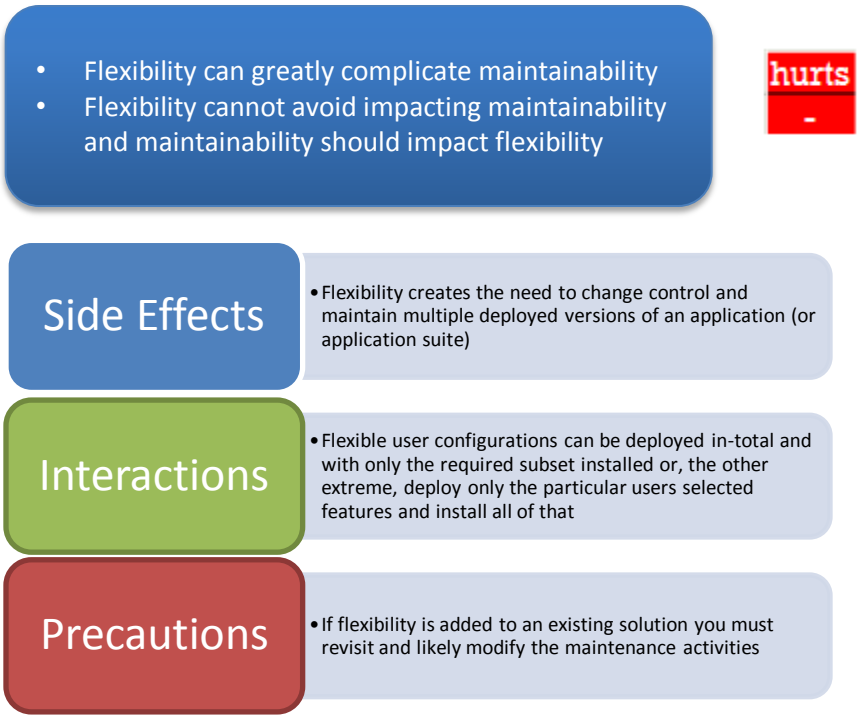


Figure 31

Here are a few more examples of pairing quality attributes. Please note that instrumentation is very important. Architects are good about building certain quality attributes in, but think little about how the solution will be monitored and maintained. If a solution cannot be measured, it is difficult to tune and improve, and even more difficult to repair. Logs and measurements are critical to the operations team. Below is an example of supportive pairs.

Figure 32

These are examples of dependencies between pairs. As you consider the quality attributes you are building into your solutions, consider the tradeoffs you are making and how you will record your decisions to provide traceability.

Figure 33

5.9 - PACKAGING AND DEPLOYMENT

As an architect you are expected to be knowledgeable in systems installation and decommissioning techniques and the issues that may arise during these phases of a solution's life.

Simply gathering and validating business requirements and creating architecture that provides the greatest business value, and driving it through development is not enough. The solution that is not successfully delivered into Operations; cannot be maintained and managed and does not have an approach for decommissioning at end of life, is not providing the intended value. As an architect, you must consider packaging, delivery, and post deployment in your architecture.

Definition

Demonstrated understanding of the expectations, process and management of IT products following the completion of development and prior to "normal" day-to-day operating conditions.

Techniques used and preferred for data conversion management, deployment strategies, documentation and training, user acceptance testing, and installation and maintenance planning.

Demonstrated competence in highest-priority management techniques.

Packaging and Deployment

- Deployment cuts across systems and moves projects into prime time
- Must ensure the production quality attributes are as expected
- Full team freeze for delivery

- Closely linked to Quality Assurance and Change Management Skills
- Tests your architecture!

Cost of Not Knowing

- Deployment costs escalate
- Breakdown between development and operations
- Unmanaged releases and versions

Skills Analysis

- 1. I have reviewed the concepts in packaging deployment
- 2. I regularly work with deployment capabilities
- 3. I have led a deployment process upgrade
- 4. I lead our companies packaging and deployment and regularly mentor others

Specialized Knowledge Areas

Software Architect

Ensure the full expected value and quality attributes of deployed applications

Infrastructure Architect

Ensure the full expected value and quality attributes of deployed infrastructure

Business Architect

Work with architecture team to set deployment metrics targets

Information Architect

Ensure the full expected value and quality attributes of deployed information systems

5.10 - MANAGING AND MONITORING QUALITY ATTRIBUTES

Managing and monitoring quality attributes. As an architect your solutions should be "Designed for Operations", which means it is tested and meets the security requirements of the organization, can be deployed, operated, maintained, and scaled. You should understand and be conversant in service level management, system software maintenance, database administration, and problem management.

In the current state of the architect profession and practice around the world, architects design solutions that meet the level of quality defined by the business requirements, but typically do not consider how they should "design for Operations". This makes it difficult to manage and maintain solutions, and drop the overall business value of and satisfaction with the solution.

As you evolve your practice of architecture, consider seeking out an operations manager to provide insight into the inner workings of the Operations organization and learn how you can create solutions that are better able to be maintained.

Definition

Managing and monitoring is a demonstrated understanding of specific quality management imperatives, techniques and tools.

Demonstrated proficiency in problem analysis, capacity planning, service level agreement (SLA) creation and management, and issue response techniques.

Managing and Monitoring QA

- Managing quality attributes in a standard and objective way
- ACROSS all IT projects
- THROUGHOUT the enterprise
- It requires tremendously advanced IT Architecture discipline
- Create metrics systems that must be integrated into governance and IT Architecture processes
- Tools and resources:
    - Requirements management
    - Quality assurance

Cost of Not Knowing

- Over spending, firefighting style
- Negative Impact on value perception & huge losses

Skills Analysis

- I understand the basic set of quality attributes
- I know when and how to balance tradeoffs between quality attributes
- I have led a team in making quality attribute tradeoffs
- I implemented a quality attribute tracking mechanism for my company

Concepts

- Quality Attribute
- Architecture Concern
- Monitoring System and Tools
- Architecture Repository

Specialized Knowledge Areas

Software Architect

Describe software project ongoing quality attribute monitoring SLA, attach system to QAtt metrics process, and justify financial value of QAtts

Infrastructure Architect

Describe infrastructure project ongoing quality attribute monitoring SLA, attach system to QAtt metrics process, justify financial value of QAtts

Business Architect

Continually monitor QAtt allocation for line of business/capability, ensure stakeholder awareness of financial value of QAtt goals

Information Architect

Ensure QAtt delivery for information usability across the enterprise, direct data and information QAtt guidelines for access, retrieval and storage

Summary

As an architect, you:

Are expected to work with executive sponsors and other stakeholders to determine what level of which quality attributes are needed.

Should understand and be conversant in service level management, system software maintenance, database administration, and problem management.

CHAPTER 6: DESIGN PILLAR

6.1 OVERVIEW

Good design is justifications, reasons, and trade off considerations.

Many architects think that architecture is the artifacts that are created and that they can reverse engineer architecture. While they can reverse engineer an existing solution back to the architectural artifacts, they cannot typically determine the reasons decisions were made or what the business requirements were that drove the solution.

While the artifacts are an important part of architecture, the value is in the reasoning and the traceability back to the business requirements.

"Design is the business of finding a way to meet the functional requirements within the specified constraints using the available technology."

— *Ensor & Stevenson, 1997*

Design is often confused with UML or diagrams. The diagrams are the output of the architectural process. Since a number of solutions could satisfy a business need, the reasoning behind picking one approach over another is truly where the value of architecture lies.

Design requires use of techniques, tools, patterns, methodologies, and a host of other tools to help us in providing predictable, repeatable success. Everybody has a bad day, and leveraging tools and methodologies helps us remember what we've forgotten to consider in our architecture.

There is both art and science in architecture, but leveraging tools and best practices will help you be effective at designing solutions.

Most everything done or accomplished is done through design, and certainly when creating the complex solutions required in today's IT environment. Design is a human activity and you are successful in creating architecture when you effectively work with other people and have a common goal or vision. As an architect, you must create a vision that others can understand and support.

- Design is a fundamental human activity
- Design is the conception and planning of all the products made by human beings
- When we create things intentionally, we engage in design.
- Design is a purposeful creative action

The rest of this chapter reviews the key concepts and cost of not knowing about core design skills. You will learn:

- Whole systems design,
- design methodologies and processes,
- requirements modeling,
- decomposition and reuse,
- patterns and styles,
- design analysis and testing,
- architectural description,
- views & viewpoints,
- traceability throughout the life cycle.

6.2 - WHOLE SYSTEM DESIGN

Do you have an integrative view of architecture; understand the synthesis of systems, networks, and services design? Does the process you use to create architecture include design evaluation, redesign, and decoupling? As an architect, taking a whole systems approach means you must understand the entire IT environment, including the people, infrastructure, development environment, and organizational structure just to name a few.

Not keeping site of the whole system will increase the cost of delivering and supporting your solution and will likely impact the business value it provides to the organization. A

critical piece of systems thinking is balancing short-term and long-term perspectives. You should design what is necessary only, but need to consider what may be necessary in the future.

Definition

Whole system design is:

The "whole system" of interconnected elements that participate in, impact, and influence the design process, including the nature and rich tradition of design theory and practice, relevancy of understanding design as a discipline.

The systems sciences, systems theory, and systems thinking are; developing “whole systems” perspective and its importance to architects, including recognizing and addressing complex systemic problems and architecture praxis.

Modeling is as essential design action, modeling tools, and business patterns, including the importance of context and the architect’s role in the creation of a design culture.

Design judgment and the construction of meaning, including work redesign, industry perspectives, and the increasing importance of architectural influence on design

Whole System Design

- Whole systems design requires that you consider the “whole system” of interconnected elements.
- The Design includes context: people, processes, structures, technology, as well as patterns of interaction
- Don’t think of design narrowly, in terms of single software components or a computing system that has interfacing applications serving a bounded scope.

Cost of Not Knowing

The cost of not knowing includes:

- Poor overall design efficiency
- Increased costs
- Decreased effectiveness of solution
- Decreased quality attributes support

Skills Analysis

- I have researched frameworks that support whole systems design
- I have a rich understanding of the organization’s IT environment
- I am part of the architectural team for the overall IT environment
- I write and speak about whole system design

Concepts

Systems Thinking:

- Thinking of "the big picture"
- Balancing short-term and long-term perspectives
- Recognizing the dynamic, complex, and interdependent nature of systems
- Taking into account both measurable and non-measurable factors
- Remembering that we are all part of the systems in which we function and that we both influence those systems and are influenced by them

Specialization Knowledge Areas

Software Architects - Ensure designs are effective for the entire impacted system(s).

Infrastructure Architects - Ensure designs are effective for the entire impacted system(s).

Business Architects - Ensure designs are effective for the entire impacted system(s).

Information Architects - Ensure designs are effective for the entire impacted system(s).

6.3 - DESIGN METHODOLOGIES AND PRACTICES

Definition

Design methodologies and practices are a collection of methods, procedures, standards, tools, and recommended practices used for creating architecture.

Architects must recognize major components of the process in various methods and be able to determine when a given method is appropriate to the problem at hand.

Architects must also understand common development cycles and their role in design.

Design Methodologies and Practices

Design is a fundamental part of IT, and it:

- Requires collaboration
- Uses modeling
- Enables removal of complexity
- Design methodologies and processes are NOT about technology alone, but about simplified solutions to a problem that delivers business value

Design tools:

- ・Visio or other design/UI tools
- ・Visualization tools / White board
- ・Case tool / UML tool

Cost of Not Knowing

The cost of not knowing includes:

- ・Waste of resources and poor designs
- ・Chaotic, unmanageable IT

Skills Analysis

- I have researched at least one design methodology or tool I use common design principles in my architecture and have applied design patterns in my software design
- I have introduced or review quality design concepts to my teams
- I work with industry design leaders to integrate design concepts

Specialization Knowledge Areas

Software Architect

Participate in methodology and process during SDLC and work with architecture team to update processes for design

Infrastructure Architect

Participate in methodology and process on infrastructure projects and work with architecture team to update processes for design

Business Architect

Participate in core design around business processes with technology strategy and work with architecture team to update processes for design

Information Architect

Participate in methodology and process information projects and work with architecture team to update processes for design.

Introduction

Design should follow a methodology and is a collaborative effort. The artifacts created should be as concise and simple as possible, and ideally will be understandable by the intended audience.

Design Practice

- Follows a methodology
- Uses modeling and non-verbal communication media as the language of design
- Uses synthesis and pattern formation as thinking tools
- Engages in collaborative (group) work
- Results in products that are adaptable and flexible

Methodologies and Concepts

- Models are critical for simplifying both software and the software process.
- Simple design focuses on delivering a system that balances delivering on the immediate needs while considering future needs.
- There are multiple levels or concerns in software development
- Do not mix modeling notations in the same concern
- Each model should represent the same level of granularity

Support Tools

Support tools Leverage more than one company for tool support and choose widely supported standards

Notation

Use the notation as it was meant to be used

Automation

Models need to be traceable in every concern.

Prototypes

Look for a level of automation. Model-driven architecture (MDA) consists of using UML models and generating code from those models

The use of models and modeling to determine the best approach you can take is a critical skill. Here is an example of a model.

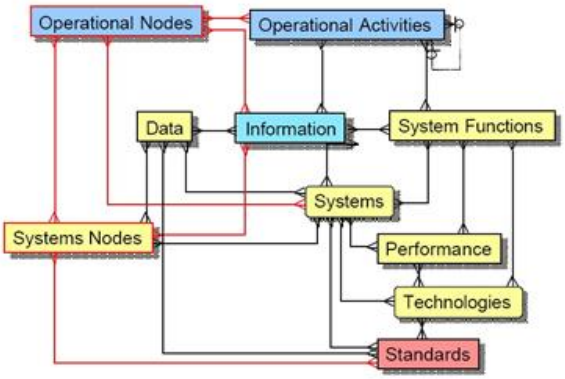


Figure 34

Patterns provide an approach that has been successfully used to solve a problem. It also provides a good communications tool for communicating with other architects and technologists. Additionally, there are anti-patterns that can be used in your architectural practice.

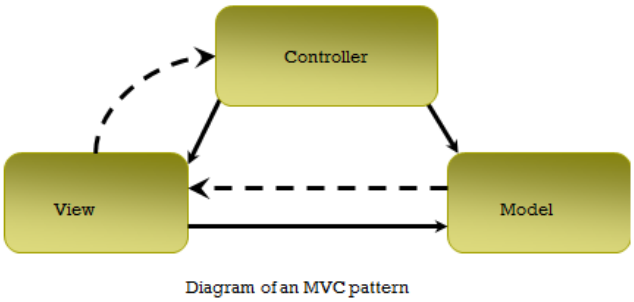


Figure 35

Here's an example of a simple deployment diagram using UML.

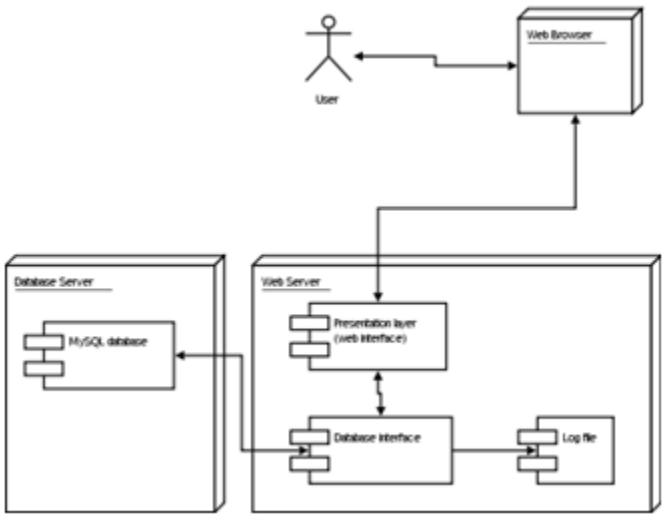


Figure 36

6.4 - REQUIREMENTS MODELING

Do you have a strong understanding of requirements definition and management? Business Modeling? Functional requirements modeling? As an architect, you should be familiar with using these skills as part of your architectural practice.

Definition

A requirement is a condition or capability to which an IT system must conform. Good requirements should be necessary, non-ambiguous, concise, consistent, complete, reachable, verifiable and clear.

Requirements modeling is the set of tasks that allow creation of requirements that meet the qualities above and are actionable, measurable, testable, and link technology decisions to business benefit.

An architect must be able to model business and technical requirements, architectures, and designs, as well as be able to transform models of one type into another.

This requires domain-specific modeling languages and other modeling approaches.

Requirements Modeling

Helps describe the functionality of the system from the users' perspective

- Good starting point is UML
- Serves as a high-level view of the system and good communication medium across user/IT teams
- Used by the project manager for project planning and resource allocation
- Can complement and define the "right" prototype
- Should be understandable by business users
- Will be used by testers for illustrating different test cases on a system

Cost of Not Knowing

The cost of not knowing includes:

- Poor mapping of solution to actual needs
- Disenfranchised users
- Lack of credibility with business analysis teams

Skills Analysis

- I have studied the basics of requirements modeling
- I have done requirements modeling on a project
- I lead my organization and mentor others in requirements modeling
- I write and speak on requirements modeling

Specialization Knowledge Areas

Software Architect

Develop high-level requirements that map to software and project components; ensure business analysis is focused on valuable requirements.

Infrastructure Architect

Develop primary requirements that map to infrastructure and project components.

Business Architect

Develop high-level requirements at the line of business and capability level; ensure project requirements are mapped to value-tracking metrics across projects within LOB and capability.

Information Architect

Develop primary requirements that map to information and project components; ensure usability and information requirements conform across line of business and capability.

Here is an example of use case modeling using standard notation. Another approach that is widely used is to leverage storyboards as part of requirements modeling.

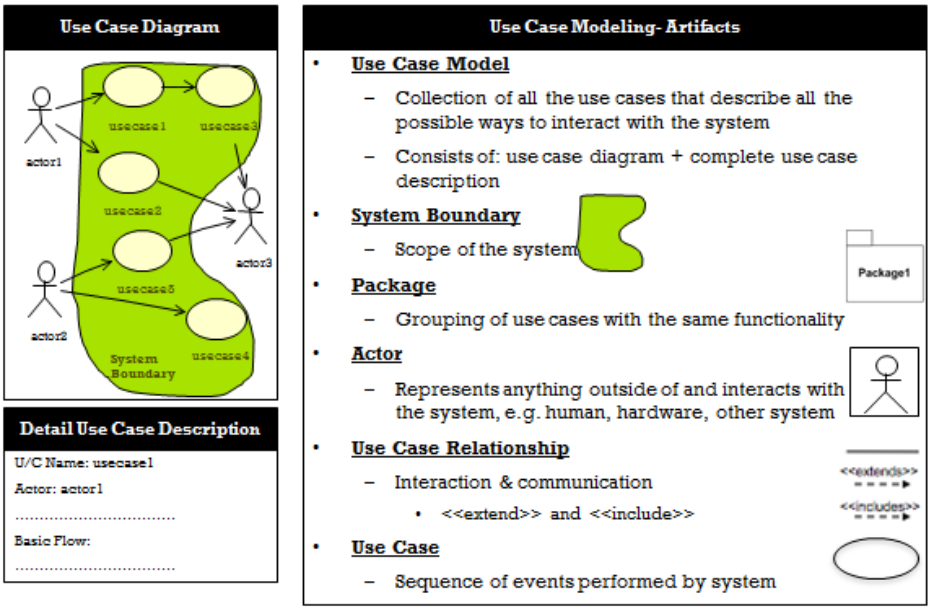


Figure 37

6.5 - DECOMPOSITION AND REUSE

As an architect you should be familiar with business decomposition, IP reuse and leveraging IP, factoring and partitioning. Subject matter experts and junior architects tend towards building over buying, and building over reusing existing IP. This can lead to unnecessary cost is new development and additional maintenance costs.

Definition

Decomposition is the process of resolving a functional relationship into its constituent parts in such a way that the original function can be recomposed. This modularity allows the architect to create a compressed representation of the global function and allows for reuse.

Reuse is the process of using capabilities for more than one solution. Rather than recreate a duplicate capability, the architect can identify components that can fulfill the requirements with little or no modification.

Decomposition and reuse becomes more challenging if an architecture team does not have a consistent and discoverable way to catalogue architectural artifacts.

Decomposition and Reuse

- Decomposition is the separation of a system into smaller or basic sub-systems
- Reuse is the process of creating systems from existing systems:
- Should be considered much like a quality attribute
- Needs to follow standard data access and interface protocols
- May include component based engineering
- Decomposition enables reuse

Cost of Not Knowing

The cost of not knowing:

- Recreation and duplication of existing components
- Unnecessary increase in maintenance costs
- Longer project timelines

Skills Analysis

- I have studied the basics of decomposition and reuse
- I have used decomposition and reuse in my role
- I created a repository that enables decomposition and reuse by my team
- I write and speak in industry on decomposition and reuse

Concepts

Synthesis and composition are at the core of a systems approach

- Identify a whole (system) of which the component (being described) will be a part
- Identify the components (that which can be changed) and the environment (that which cannot be changed) (boundary identification)
- Describe the behavior and properties of the whole system

- Describe the behavior and properties of the component (being described) in terms of its role(s), functions(s), processes, patterns, technologies, and human interrelationships within the whole system

Specialization Knowledge Areas

Software Architects

Design with reuse in mind; publish details within reuse framework; identify opportunities for software reuse through component decomposition.

Infrastructure Architects

Design with reuse in mind; publish details within reuse framework; identify opportunities for infrastructure reuse through component decomposition.

Business Architects

Identify opportunities for reuse across line of business; identify areas of value for reuse and invest

Information Architects

Design with reuse in mind; publish details within reuse framework; identify opportunities for information reuse through information structure decomposition.

6.6 - PATTERNS AND STYLES

As an architect you are expected to understand the application of metaphors, analogies, and similes, be able to recognize patterns that are forming, and be skilled at harvesting and reuse. Additionally, you are expected to leverage style, patterns, and anti-patterns. Please note, as with many architectural terms, styles and patterns are used interchangeably by some.

As an architect you are expected to understand the application of metaphors, analogies, and similes, be able to recognize patterns that are forming, and be skilled at harvesting and reuse. Additionally, you are expected to leverage style, patterns, and anti-patterns. Please note, as with many architectural terms, styles and patterns are used interchangeably by some.

Definition

A pattern is a three-part rule that expresses a relationship between a certain context, a problem, and a solution. As Christopher Alexander suggests, “Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over and over, without ever doing it the same way twice.”

Styles are broader than patterns and more generalized than patterns. For instance, layered architecture is a pattern that follows a call and return style of interaction. Model-Driven Development (MDD) and Representational State Transfer (REST) are examples of styles.

As a practicing IT architect, you should have a demonstrated understanding of metaphor and pattern concepts, styles versus pattern, and demonstrated competence in deriving and communicating pattern and style.

Design Patterns and Styles

- An organization of functionality for implementation into a coherent set of software code
- Used to reduce overall cost of solutions where build is necessary
- A style is broader than an architecture pattern
- An architecture pattern is broader than a design pattern

Cost of Not Knowing

- Increased build costs
- Lower flexibility in output
- Reduced overall quality attribute levels

Skills Analysis

- I have studied design patterns and styles
- I have used design patterns and styles in my role
- I lead my organization and mentor others in design patterns and styles
- I write and speak in industry on design patterns and styles

Concept

- Architecture style has a distinct purpose for communication of a specific understanding
- Good patterns identify recurring designs that capture proven design practice
- The focus of a pattern is on the problem and the problem's context (instead of just naming a solution)
- Pattern-based thinking is the interplay between problem forces and solution consequences used to document recurring solutions to problems in other design styles and domains
- Design styles and domains-enterprise architecture, concurrency, organizational structure, user-interface design, etc. were about building architecture before they were about software architecture…Christopher Alexander was the guy's name

Specialization Knowledge Areas

Software Architects

Identify patterns in software projects that increase value metrics

Infrastructure Architects

Identify patterns in infrastructure projects that increase value metrics

Business Architects

Identify patterns in business technology strategy that increase value metrics

Information Architects

Identify patterns in information projects that increase value metrics

6.7 - DESIGN ANALYSIS AND TESTING

As an architect you should be able to perform pattern analysis, provide validation of design elements, and identify and remediate poor design. There are several methodologies and tools that can be used for design analysis, with Architectural Tradeoff Analysis Method (ATAM) and Perspectives-Based Architecture (PBA being two of the best known.

Design analysis and testing allow you to discover issues with your solution earlier in the project. The earlier in the lifecycle to identify and mitigate a problem, the less impact to schedules and costs.

Definition

Architecture analysis techniques and related tools allow an IT architect to evaluate a design relative to alternatives, describe the structure and state of a design, and analyze the design's dynamic behavior in response to external events. They can then test a design as an artifact for completeness, correctness, efficiency, and a number of other criteria.

Highest priority techniques such as selected ATAM techniques, failure analysis, optimization, prototyping, simulation, scaled modeling, and various forms of walk-throughs and design reviews can be used to perform analysis and testing.

Design Analysis and Testing

IT Architecture analysis and testing techniques are becoming common

- Mitigate/reduce risk
- Allows team to test effectiveness of an IT architecture
- Generally based on use case scenarios and perspective analysis
- Tools:
- ATAM (Architecture Tradeoff Analysis Method) by SEI-CMU
- Perspectives-Based Architecture (PBA/PBAAM)

Cost of Not Knowing

The cost of not knowing includes:

- Poor design not discovered until implementation or deployment
- Poor architecture value

Skills Analysis

- I have read and/or used at least one industry accepted analysis tool
- I have participated in an architecture analysis process
- I introduced or mentor others in analysis
- I have developed or speak about design analysis methodology

Concepts

The assessment method can be chosen according to:

- the software quality attributes to be assessed
- the input required by the assessment method and the architecture description format used
- the process stage in the software lifecycle when the assessment is performed
- the resources required and the time allotted to the assessment
- the scope of the assessment, e.g., the entire software architecture or system

Specialization Knowledge Areas

Software Architects

Analyze designs using standard analysis tools and methods; participate in analysis process on software projects.

Infrastructure Architects

Analyze designs using standard analysis tools and methods; participate in analysis process on infrastructure projects.

Business Architects

Participate in design analysis across LOB and capability.

Information Architects

Analyze designs using standard analysis tools and methods; participate in analysis process on information projects.

Evaluation Methods

- Scenario-based Architecture Analysis Method (SAAM)
- Architecture Tradeoff Analysis Method (ATAM)

- Active Reviews for Intermediate Design (ARID)
- SAAM for Evolution and Reusability (SAAMER)
- Architecture-Level Modifiability Analysis (ALMA)
- Architecture-Level Prediction of Software Maintenance (ALPSM)
- Scenario-Based Architecture Reengineering (SBAR)
- SAAM for Complex Scenarios (SAAMCS)
- Perspectives Based Architecture (PBA)

6.8 - ARCHITECTURE DESCRIPTION

Do you know how to create conceptual descriptions and notations? Logical descriptions and notations? Have you ever created implementation descriptions and notations? As an architect you are expected to understand the difference between these different architectural artifacts, when to use each and who the audience is for each.

The architecture you create is complex and likely hard to understand. There are several methodologies and tools that can be used to create architectural description. Whether you are a staff architect or a consulting architect, your organization likely has certain templates and required artifacts you will be required to make. In order to broaden you understanding of AD, consider researching UML and Archimate, comparing and contrasting the two as a step to growing you skills.

Definition

Architectural description provides detailed architecture communication and includes diagramming notation, architecture views and viewpoints, and is organized into an architecture description language (ADL).

- ADLs provide a way to communicate an architecture to all stakeholders, not just technologists, and must:
- Be suitable for communicating an architecture to all interested parties.
- Support the tasks of architecture creation, refinement, and validation.
- Provide a basis for further implementation, so it must be able to add information to the ADL specification to enable the final system specification to be derived from the ADL.
- Provide the ability to represent most of the common architectural styles.
- Support analytical capabilities or provide quick generating prototype implementations.

Architecture Description

The goal is to transform the collected and organized architectural information and intents into viable models

- Describes the functional and non-functional requirements of the architecture
- Creation and adoption of effective architecture approaches, methodologies, patterns, reference models, and so on

Cost of Not Knowing

The cost of not knowing includes:

- Inability to deliver value as an architect
- Unfocused and unclear solutions
- Inability to communicate with stakeholders

Skills Analysis

- I have studied approaches to creating architecture descriptions
- I create architecture descriptions for the projects I am involved in
- I lead my organization and mentor others in creating architecture descriptions
- I write and speak in industry on creating architecture descriptions

Specialization Knowledge Areas

Software Architects

Create formal architecture descriptions for software projects; develop views/viewpoints and communication based on project value.

Infrastructure Architects

Create formal architecture descriptions for infrastructure projects; develop views/viewpoints and communication based on project value.

Business Architects

Create formal architecture descriptions for business architecture projects; connect project descriptions and track them across business architecture.

Information Architects

Create formal architecture descriptions for information projects; connect information views from projects and manage to information architecture.

Getting Started

- Architecture description is the end-product of an architecture development process
- First step is to understand the purpose of the system, its goal, and possible usage scenarios
- To make the architecture description content effective and well accepted, identify the scope of architecture coverage and the intended audience

Keys to Success

May describe both “as-is” and “to-be” states

- Successful architecture description has to consider the background of the audience and provide:
- Architecture vision and views
- Summarized major competitive products
- Analysis of pros/cons in features and in technical implementations
- Technology options for the described architecture are to convince and provide guidance for the engineering teams in actual product implementation
- Effective information collection and synthesis are important to establish a baseline for architecture description

Connecting to Stakeholders

- Purpose, scope, and audience provide help, guide position, and aid in acceptance and buy-in.
- The architecture will be more acceptable to a stakeholder if reflected from their view
- Create different models and views for the same concepts to suit people in different roles, responsibilities, and background
- Market analysis, technology options, and implementation roadmaps are helpful in bringing architecture to reality in an iterative and phased manner

Architecture Description Languages

- ACME / ADML (CMU/USC)
- Rapide (Stanford)
- Wright (CMU)
- Unicon (CMU)
- ByADL (Build Your ADL) - University of L'Aquila
- LePUS3 and Class-Z (University of Essex)
- ABACUS (UTS)

6.9 - VIEWS AND VIEWPOINTS

Views and viewpoints are terms that can be challenging to use with discussing with other architects. In the 1990s views and viewpoints had one definition that was used is some very popular books, and IEEE introduced definitions that were different. The Iasa has adopted the IEEE definitions for views and viewpoints. Nick Rozanski and Eoin Woods do a good job presenting the definitions for these terms using the IEEE definition. Have you ever read the 1471 standard? Consider reading it as part of your lifelong learning roadmap.

Definition

A view is a representation of one or more structural aspects of an architecture that illustrates how the architecture addresses one or more concerns held by one or more of its stakeholders.

A viewpoint is a collection of patterns, templates, and conventions for constructing one type of view. It defines the stakeholders whose concerns are reflected in the viewpoint and the guidelines, principles, and template models for constructing its views.

An example would be to use an operational viewpoint to create a view targeted at the Help Desk manager. The viewpoint is the template that contains information relevant to the operations of the system, and a view is the end product delivered to someone interested in maintaining the operations.

IT architects must have the ability to compare/contrast concept of views, viewpoints, and perspectives, differences between them and how they work together to describe an architecture. They must be able to discern various stakeholder groups typical of IT development projects, describing the typical viewpoint of each group, and determine the set of views needed to satisfy project requirements.

Views and Viewpoints

Based on IEEE 1471

- Views are IT architecture descriptions focuses on a specific stakeholder
- Viewpoints are templates for views
- Generally we only need 2-4 views
- Create your own views as a part of your IT architecture artifacts

Tools:

- Word Processor / Text Editor
- Troux

Cost of Not Knowing

The cost of not knowing includes:

- Lack of business support
- Unable to effectively describe the architecture to different audiences

Skills Analysis

- I have studied standardized views from different sources
- I create views on most architectural projects
- I have led the adoption of view construction and delivery
- I have delivered an industry set of views and viewpoints

Specialization Knowledge Areas

Software Architect

Create views as a part of software architecture documentation; participate in viewpoint creation as a part of architecture methodology.

Infrastructure Architect

Create views as a part of infrastructure architecture documentation; participate in viewpoint creation as a part of architecture methodology.

Business Architect

Create views as a part of business architecture documentation; participate in viewpoint creation as a part of architecture methodology.

Information Architect

Create views as a part of information architecture documentation; participate in viewpoint creation as a part of architecture methodology.

Stakeholder

- A Stakeholder is any person with some vested interest in the project
- Typical stakeholders include customers, developers, maintainers, upper management, etc.

Stakeholder viewpoints

Define viewpoints to address all the important concerns we and our stakeholders have

- Clarify whether the methodology we follow suggests them or not
- Help ensure all the system's quality attributes are met
- Define the number of views based on the project size or complexity

Difference between a view and a viewpoint

A viewpoint is a pattern or a generalization of a view and defines the rules for the views created from that viewpoint

- Example - 4+1 views as RUP defines them should actually be called the 4+1 viewpoints if using standard terminology
- Viewpoints are selected according to the concerns of the stakeholders

A view is a representation of the whole system as seen via the prism of specific concerns

- A view can be regarded as an instance of a viewpoint
- A view consists of one or more models that represent it

Viewpoint repository or library

A viewpoint repository or viewpoint library is a name for the set of viewpoints an architect knows about. You can draw on these to find a viewpoint suitable for your immediate needs

Models and viewpoints

A model is an abstraction or simplified representation of a system or part of a system

- Models are created using standardized notations such as UML or E-R diagrams as well as non-standardized notations such as block or arrows diagrams
- As viewpoints are system agnostic, they are basically templates, thus we can also build ourselves a viewpoint repository and reuse viewpoints when needed

This is a simple example of using a viewpoint to create a view that results in a pertinent document for a system administrator.

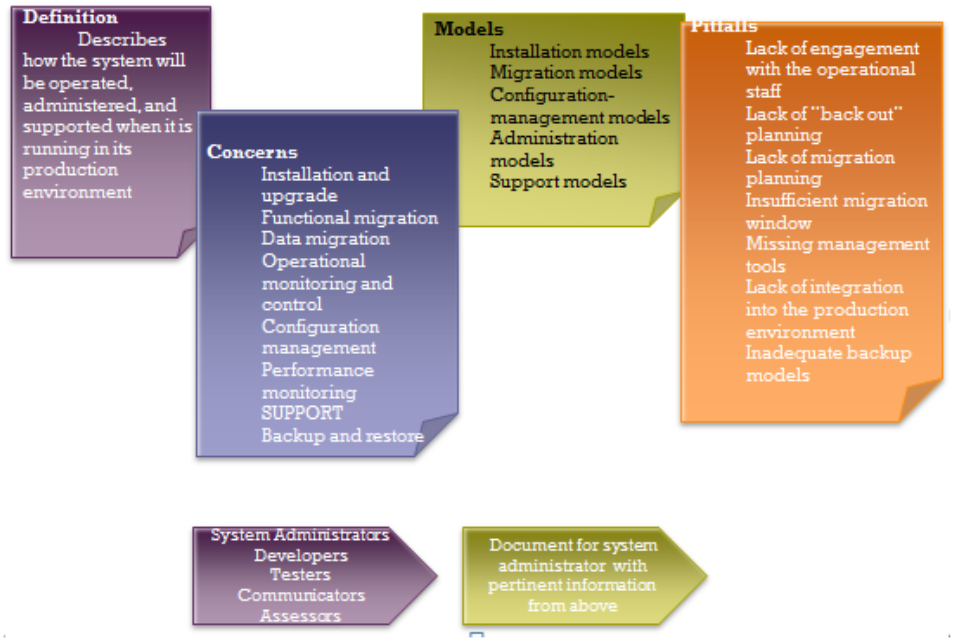


Figure 38

6.10 - TRACEABILITY THROUGHOUT THE LIFECYCLE

Have you ever had a project where the scope changed during development? How did you make decisions on what features to cut or the impact not providing functionality would have on the business requirements and value of the solution? What about the impact to other project that was also in flight?

As an architect you are expected to understand traceability management, create description of vision-to-feature traceability, and business process testing and validation. Providing

traceability is the responsibility of the architect. The architect is the only project member that is fluent in business, architecture, and technology, and provides the link between business need and IT implementation.

Definition

Traceability is the process of documenting the life of a business concern and providing bi-directional traceability to the capability or capabilities associated with that concern.

The purpose of having traceability is to facilitate an understanding of the solution under development and to provide the ability to prioritize and manage change.

Architects must be able to provide traceability from initial requirements through to the sustained system and describe the vital role of traceability throughout the lifecycle of developed products. Architects should have mastery in the use of associated techniques and tools, and demonstrate competence in highest priority techniques.

Traceability throughout the Lifecycle

Documents why critical decisions were made and how the resulting changes affect the architecture

- Links tracking decisions to effective value metrics
- Map to business metrics and show value of IT
- Creates a “heat map” of critical business areas

Tools:

- Needs significant tool development.

Cost of Not Knowing

- The cost of not knowing includes:
- Lack of understanding of core value across projects
- Inability to understand existing state of technology strategy
- Lack of impact in chosen direction

Skills Analysis

- I have studied the basics of design patterns and styles
- I use design patterns and styles in my role
- I lead my organization or mentor others in design patterns and styles
- I write and speak in industry on design patterns and styles

Specialization Knowledge Areas

All Architects

Map core decisions into traceability framework; track effectiveness of delivery and report on tradeoffs.

Summary

IT Architecture is not just design and technology; it is about having well rounded skills and having the ability to act in the role of each of the stakeholders on a project. We don't suggest that you will or should over function, just that you need the skills of each in order to be empathetic to each stakeholder and understand what drives them.

The results of the survey are skewed somewhat towards software solution architect, but that is the largest population of people that took the survey and does represent the mix of specialization in the current architect community.

CHAPTER 7: ENGAGEMENT MODEL

7.1 - OVERVIEW

In this chapter we will:

- Describe and clarify roles, scope and context.
- Provide engagement models.
- Detail engagement attributes.
- Describe different models for organizing architects.
- Demonstrate engagement selection processes.

So what exactly is architectural engagement? Engagement defines how architects interact with an organization. It defines how the organization selects and governs its people and what processes they utilize. It is how you map the roles of architecture to the work of architecture which delivers and demonstrates value to the company. It is in effect how each architect in the company touches the rest of the business officially and unofficially. In short engagement is everything of significance you do at work.

- How architects interact with an organization
- How the organization selects and governs its people
- How you map the roles of architecture
- How each architect touches the rest of the business

Here are some of the components that are defined or described within an engagement model:

- Governance through people and process
- How architects interact with an organization
- The touch points of the architecture team
- Mapping role and skill set to activity

- Demonstrating value

Sounds complicated? Well it's not so bad. We will walk through the key points in a moment. And when managed effectively it provides the key to success in architecture teams. It puts a face on the architecture practice with real people. The focus of engagement is on skill delivery at critical points throughout your business which ultimately controls the entire perception of architecture value.

The models we are about to look into haven't been pulled from thin air. Iasa has interviewed many types of organization and has found many common threads to successful engagement model. The first such item is the ratio of architects to IT staff.

Most of the architecture teams that report success have the following percentage ratio. For example, in corporate IT a ratio of 5% architect to total IT staff has been reported as the most successful. In team layout, they report that having an architect per project of a certain complexity is the key to success.

- Puts a face on architecture
- Is about people and skills
- Will control the value perception of the architecture team

Research Evidence:

- Iasa has interviewed many organizations (corporate, service and vendor)
- Successful reporting of architecture initiatives have the following profiles

Touch Points

We just mentioned the concept of scope and combined with context these form the keystones of engagement models. The common company is made up of verticals which may be called lines of business which in turn have departments or sub groupings. In general the executive staff works at the enterprise level. Management works at the line of business level and employees at the department or workgroup level. This is what we call scope of impact in architecture. Technology strategy decisions made at higher levels in the organization have larger complexity and scope.

Types of Engagement

Figure 39

So how could an architect engage a business at all these levels of scope. First it's a given that the ownership of technology value (or strategy) is our responsibility.

Think about the number of decisions that are made at each level of scope. Strategy is meaningless without communication so there are a number of models and

documents to develop for each level and we will have meetings and casual conversation across the whole of the enterprise. All of this development of documents and models will need to be done in some order which means an architecture process which needs to be integrated into other processes at each level of scope. Of course implementing technology strategy also means a significant amount of procurement which controls what and when we buy things. Obviously implementing an engagement model will take some time.

Architecture Context

We talked a little about scope but what is the context? Architectural context is based on the organizational context within which architecture decisions are made. For example, within the enterprise (which defines the first level of scope) there is the sales department which is working on a commerce project. The commerce project has a team, the sales group has management and leadership and a set of existing business rules and processes which are a part of the organization. The company may have relationships with other organizations as suppliers or customers which also need to be considered for each architecture decision. Each of these items can be thought of as impacting the context in which you make decisions or influence them. Your goal is to optimize the technology strategy for the level of scope and the context within which you are working.

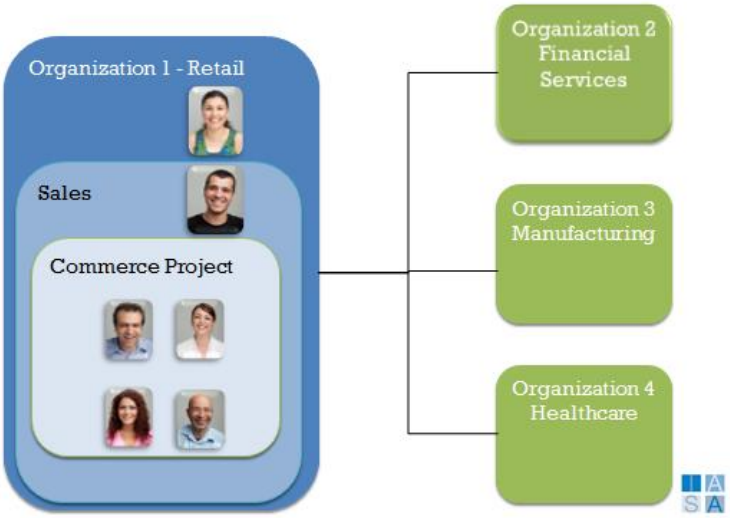


Figure 40

Let's dive into that in some more detail. Think about the commerce project we were just describing. The architect assigned to the project is dealing with quite a bit to ensure the technology strategy is best realized. Officially they do so using rigorous communication practices like the architecture description which may include different 'views' of the architecture such as the Sponsor or Project Team views in addition to others. (Note: these views are a part of the process and engagement model your company uses).

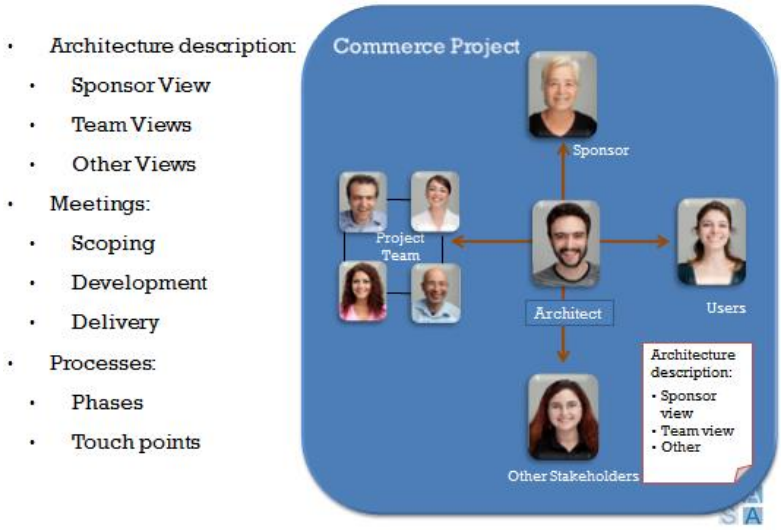


Figure 41

The architect may actively engage in scoping and development meetings as well as delivery meetings with the Project Team. The Process may define the phases in which the architect works with users and other stakeholders to define requirements which impact the strategy. Each one of these touch points defines the context at which you are working.

What about other examples of engagement? The business architect may be having long term strategy meetings with executives or other management. They may be building structure and process models and ROI and TCO reviews on the technology strategy within their context and scope.

The infrastructure architect is busily engaged in operations discussions and data center designs or building cost benefits analysis of network investment for procurement. All the while the software architect is involved in the architectural design reviews of component estimations for the commerce project while doing scoping and project estimates with the project manager.

These examples give you a since of how pervasive architecture must be in order to be successful.

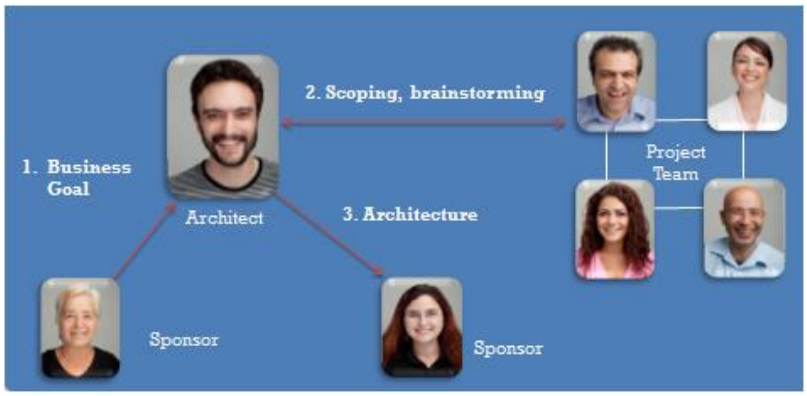
Figure 42

I also want to point out how engagement MUST become proactive. Remember the previous bad example of architecture? That mistake would have cost the company 300 thousand or more. But your engagement model CAN be proactive and designed according to your context WITHOUT having to reinvent the wheel. Of course you have to train architects for the 'dynamic engagement' that happens (like conversations around the water cooler).

Just think about the following kinds of things:

- When are architects required to interact with others?
- What do they deliver?
- To whom do they deliver and how do they interact?
- What processes control the interactions?

Look at the following diagram and write down your thoughts about the engagement model.



Mapping Roles to Activity

One other note about building an engagement model that works. Architect specializations and roles MUST be mapped properly for it to be successful. You must think about the levels (pay grades) that interact and make them match closely. The architect MUST be well trained in the areas of engagement and the situations in which it happens must be conducive to the goal of engagement (i.e. good technology strategy). The process must allow time for effective engagement and not get in the way (too much process).

- Levels (the 'pay grades') must match closely
- The architect must have appropriate skills
- Situations must be conducive
- The process must be supportive
- The process shouldn't get in the way

For example if you put a junior software architect together and alone with the VP of Finance it probably won't lead to effective technology strategy. The architect must have the capability and authority to own the strategy to be able to demonstrate value.

Architecture Frameworks and Engagement Models

Another thing to consider in engagement models are architecture frameworks. There are now dozens of quote frameworks on the market. Examples include the Federal Enterprise

Architecture Framework. The Department of Defense Architecture Framework. Others include things like MODAF and TOGAF.

Some are good in your scope and context and some won't work at all. If your organization just selects one randomly and doesn't make the decision proactively and carefully it will negatively impact your value and not the opposite. Architecture professionals should be a committed group of architects with the right skills rather than a group following a framework by rote.

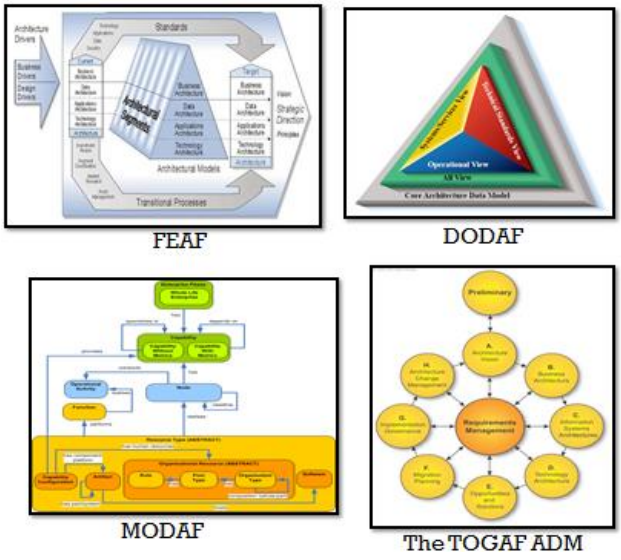


Figure 43

What the frameworks should do and the way you should evaluate them is to understand their impact on processes and activities of the architecture team and their stakeholders. Understand the amount and type of documents and deliverables from each and the implications to adopting them. Understand what the framework does and doesn't do well and how it compares with the others. Understand how it defines skills and capabilities and how they map to your team. You should also understand what the frameworks do and don't cover so that if necessary you can use multiple frameworks or fill the gap yourself.

For example, looking at TOGAF there are some noticeable missing elements when defining and engagement model. TOGAF doesn't define and provide templates for all potential

It doesn't define clear connections with the SDLC or operational models like COBIT and ITIL. If you adopt the framework you will have some customization work to do.

REFLECTION POINTS

Can you describe 3 examples of architecture engagement that you have seen work well? How about 3 examples of architecture engagement that have failed? Why have they been successful or failures?

Think about things that might be included in scope. The reason you must understand this is that scope drastically impacts the roles and structure of your architect team. An enterprise architect may have enterprise level impact but remember that they couldn't possibly be involved in each actual project. Scope is impacted then by budget, by staff size by complexity or project size and by business impact. Note, in some organizations these may be called aspects.

Think about the commerce project in Figure 41. The architect assigned to the project is dealing with quite a bit to ensure the technology strategy is best realized. Officially they do so using rigorous communication practices like the architecture description which may include different 'views' of the architecture such as the Sponsor or Project Team views in addition to others. Please note, these views are a part of the process and engagement model your company uses).

The architect may actively engage in scoping and development meetings as well as delivery meetings with the Project Team. The Process may define the phases in which the architect works with users and other stakeholders to define requirements which impact the strategy. Each one of these touch points defines the context at which you are working.

Capability or Domain Scope

Now let's take a separate type of scope. Since architecture is a technology strategy then it can be applied at all levels of scope. For example, let's pretend the sales line of business for a retail company is setting up their strategy for the next year of operations. The VP of Sales would work with marketing, finance and other directors in the sales group to set the strategy. But who would they work with to form the technology vision as a part of that strategy? The answer is the business architect. In this instance the business architect would influence each of these stakeholders but their scope of impact would be the technology value of the entire sales group throughout the coming year including all of the projects that are started because of it and all of the business strategy that is touched by it.

Now flip the last two scenarios in time. If the business architect with the strategy team decided and demonstrated the need for an e-commerce project hand in hand with the software and infrastructure architects, then the software architect would be completing that strategy by delivering the architecture for the commerce project. So the business architect impacts a broader scope of strategy but cannot complete it without other architects. A similar relationship exists between enterprise architecture and the other levels of scope.

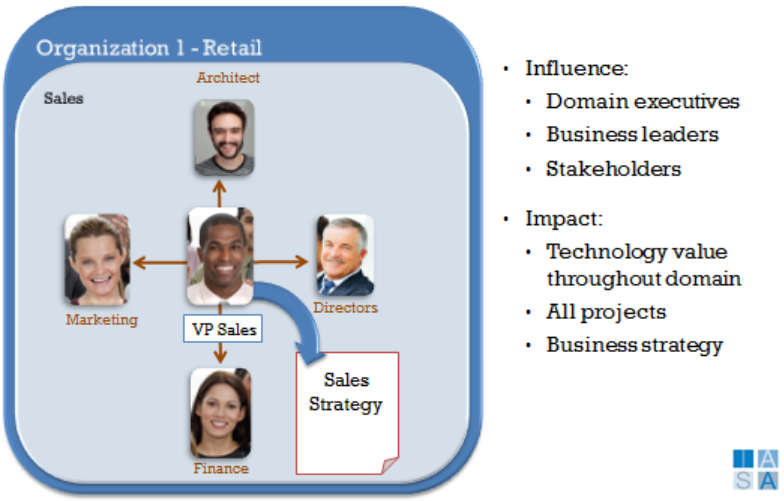


Figure 44

Scope & Roles

In many of the organizations you work with you will find that they modify titles of employees based on the concept of scope and context. We have identified 30-40 different architect titles which is the result of the slight modification of scope, domain or context. For example, security architect may be developed by putting a cross organization view of a single quality attribute.

The scope describes how much strategy you influence. It implies the complexity of the domain (content focus). When designing your engagement model you should consider the scope of each role and their impact. Does each project have an architect? Do they participate at the LOB level? How do they engage is it proactive or reactive?

There is a lot of risk in setting up an architecture team, especially a pure enterprise architecture team. The primary issue there is lack of perception of value due to an inability to cover the full scope of the enterprise. For an example, consider an enterprise with 25,000 employees. IT has 2000 staff and 100-200 concurrent projects per year (from small to very large) and a budget of 20 million. Even with 1% ratio of 20 EA team members there is very little chance that they could cover all of the projects or engage the entire business. So they end up in a governance and modeling role which ultimately makes it difficult to show any bottom line value.

- Organization of 25,000
- IT has 2000 employees
- 100-200 concurrent projects
- IT spend 20 million
- Enterprise architecture has 20 people (1%)
- Cannot cover every project
- Cannot engage the entire business
- Only possible activities are governance and modeling
- Makes it very difficult to show value

When I talk about context I'm referring to the circumstances and facts surrounding each architect role in an organization as you can see in the definition "The set of circumstances or facts that surround a particular event, situation, etc."

Put in architecture context, the primary set of circumstances or facts that impact the nature and execution of architects within an organization.

Things that impact context can generally be classified using a matrix of influences such as business size, type and business unit. For example, large financial institutions tend to have certain character of architecture context when compared with similarly sized retail organizations.

Additional contextual influences are location, including geography, language and culture, process, framework and SDLC levels, current architecture level as well as understanding. Since context affects everything about architecture internally it is important to accurately map these influencers when considering an engagement model.

- Business size
- Business type
- Business unit
- Location – geography, language, culture
- Process and framework
- SDLC
- Architecture level
- Architecture understanding
- Context impacts architecture capability and role

Here's an example mapping of an organizations context. Compare the following and describe the impacts in the course forum:

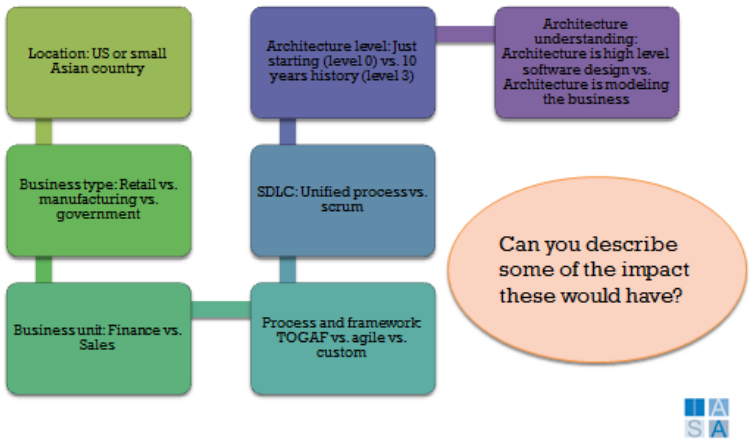
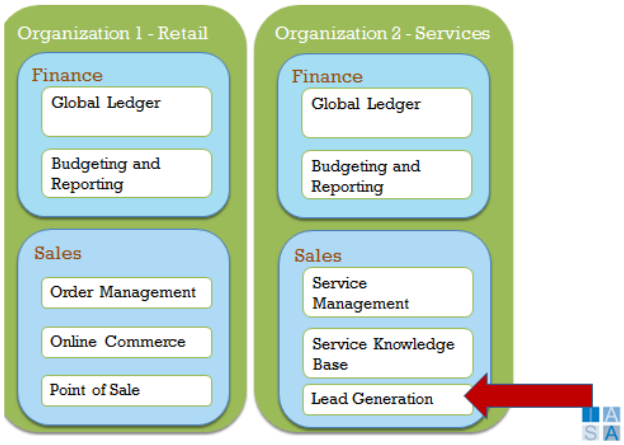


Figure 45

- The difference between a US or small Asian country based business.
- A retail vs. a manufacturing company
- A finance related business unit vs. sales
- A TOGAF vs. Agile process

- Unified process vs. Scrum SDLC
- A company that is just starting to understand and use architecture vs. one with 10 years of history
- Consider these differences

Let's look at this a little closer. Most companies have very similar operations and business units. They have a finance group that handles the global ledger, accounting, budgeting and reporting.



There will be some differences based on their organization type. But as you get into the sales groups they may drastically differ. You as an architect will have to understand where you fit into this equation. Are you externally focused or internal? Service firms often provide external architecture to their clients but often fail to adopt it formally for themselves, a situation we like to call eating their own dog food. What other differences might exist based on these two particular architecture contexts?

Context

So a general framework for understanding your context is based on a series of Q/A sessions. How many employees does your business have (this will generally limit the size of your architect team and engagement)? How much revenue do you bring in annually? Quarterly? What type of business are you in? What are your customers like? What is the key differentiator of your business? How is strategy developed? How does the budgeting process work? What is IT spending per year and per quarter? What percent is maintenance vs. new development? Has architecture ever been tried there before? If so, by whom and when? What happened? What are the processes for SDLC, procurement and project selection?

The answers to these questions define your context.

Roles

Now what about roles? Here's the definition; "the actions and activities assigned and expected of architects at an organization." *Dictionary:* "The actions and activities assigned to or required or expected of a person or group"

Architect roles define the scope, context, activities and therefore necessary skills for every single architect at your company. Stop for a minute and think about how those roles were defined. Who created the job definitions and background? Where did the skill set come from and how is it vetted (tested)? If you are like most organizations today you will find that it came about and is generally managed very loosely.

Figure 46

When working on your engagement model it's good to know how it is normally done. For the most part in industry common roles are defined outside of the company. Accountants, lawyers, finance and business roles are well described and have a large infrastructure supporting them. However for architecture it is normally HR and IT INSIDE the company that define the role.

They sometimes use external basis such as technology trends (think SOA) or frameworks such as TOGAF® to help but for the most part they simply make them up based on the knowledge and experience of the internal staff. In the main they don't govern the role itself but setup a series of implicit and explicit goals that the architects must meet. In some cases they include the history of the role as they see it such as a focus on software engineering etc.

Other questions such as how each individual in the architect role is hired, their skills framework, what constitutes a good or a bad architect and the required background and ongoing education for the architects are simply made up by the internal staff. This is on the main how architecture is managed. In the next module we will talk about a way to do it that is based on more standardized staff management models found in finance, sales and HR.

7.3 - ENGAGING ORGANIZING PRINCIPLES

We are going to complete the engagement model for you to prepare you for setting up your own architect organization or improving it. We are going to talk about explicit value management of the architect engagement, about the skills and coverage of your team. We follow that up with a discussion of managing progress and taking a frequent look at the full architect delivery capability and improving over time. Again this is not a proscriptive process but it builds a model that allows you and your team to make the most out of your organization.

Value

The first and last thing that you will be known for is your value to the organization. If having an architecture team makes significant profit for the shareholders (whatever their definition of profit is) you will simply never have to worry about them wanting to keep architecture going. However to actually provide value you have to do a number of things.

You have to map your value to understand where it is coming from. You have to demonstrate that regularly and repeat it. Then you have to ensure you are communicating that value to anyone who will listen. Remember it's not bragging, it's simply reporting. For some reason architects tend to not take enough credit for their value. You have to continually review your value and progress and then refactor your engagement approach to map to changing business.

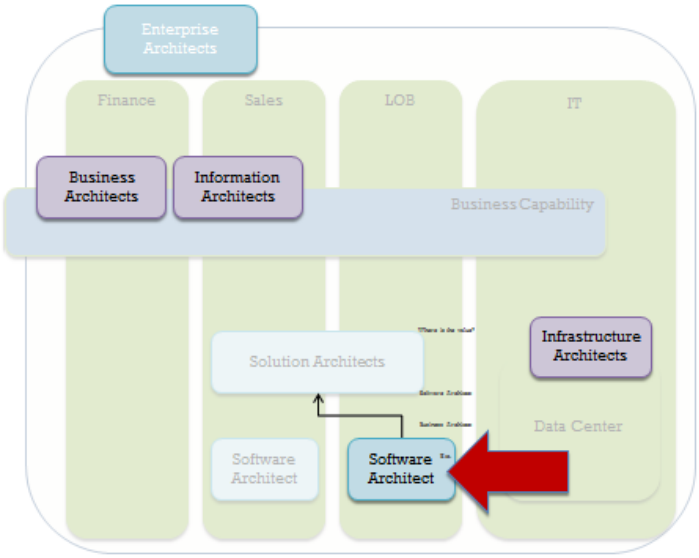


Figure 47

Iasa finds the architecture engagement model to be very valuable in mapping value. What I mean by mapping value is to determine WHERE it happens and WHEN. For example if you are creating software architectures they should be able to demonstrate in numbers exactly how much cost reduction they have done. They should also be able to define how many new capabilities they've delivered and what the value of those capabilities is to the business. If the value numbers are low then you need to focus on how to improve them. If they are high figure out how to duplicate them.

What about other mappings such as business, information, infrastructure architecture? Are you even doing those things yet? If not how would their individual value contribute to the whole? If you put advanced business technology strategists at the LOB or business capabilities level they could very likely influence or deliver significant revenue. What about information architects? How would better using, storing or handling information serve the bottom line of the company?

If you are at the enterprise architect level, are you successfully able to provide proactive value at all of these different levels? Are you proactive and rolling up value from all of the different scopes and contexts within you company? Do you have a bottom line value spreadsheet that shows exactly how much money architecture has saved and made for the company? If not, watch out and keep your resume updated.

Demonstrating value is about numbers. If you can't show it in numbers then it isn't value. Even 'soft' value can be shown in numbers if reported properly. Yes you may have to invent certain reports but so what?

Cost component	Product XYZ	Product ABC
Software license	0	40,000
Hardware	120,000	60,000
IT environment	20,000	10,000
Training	30,000	10,000
Consulting	35,000	0
FTEs	300,000	100,000
Contractors	40,000	0
Total cost of ownership	510,000	220,000
Difference		+310,000

The project demonstrated $310,000 lower cost before ever getting started!

Figure 48

Think about how 'soft' the numbers for marketing really are. It is tremendously difficult to prove that a single commercial could impact someone's purchasing decisions yet companies do this all the time and it works. As architects you need to be able to demonstrate your value through numbers.

Communicate Value

It's not enough to just provide value unfortunately. You also have to communicate value. Even in the most robust architecture teams today (the ones that comprise 5-7% of IT staff) architects still account for a tiny fraction of total employees. Think about the company with 25,000 employees. Even with 500 architects the ratio is way off. How will all those people know your value if you don't make it simple to understand and easy to find? When I consult on setting up architect organizations I spend most of my time teaching them when, how and where to communicate their value. For some reason architects seem to have issues with this, like they are somehow self-promoting.

You need to explicitly manage your value communication. Make a list of stakeholder that your team regularly works with. Have you surveyed them? How do they see architecture? Do they know that you saved the company 2 million dollars last quarter? Who is missing from the list? When you start doing this explicitly you will learn a great deal about architecture and hopefully so will the people you are communicating it to!

Process

Process is a key area to consider when building an engagement model. For example, architecture often happens before a project officially gets started. As you can see in the example project portfolio schedules you need to engage both before and after projects as well as understand the engagement across all projects. The portfolio management file

attached to the course will help you if you don't already have a process in place. Remember though architects work on projects we are responsible for technology strategy beyond the project. That means you are not done when it's over, in fact if 80% of spend on a project happens after it is deployed then your job as a team is just beginning then.

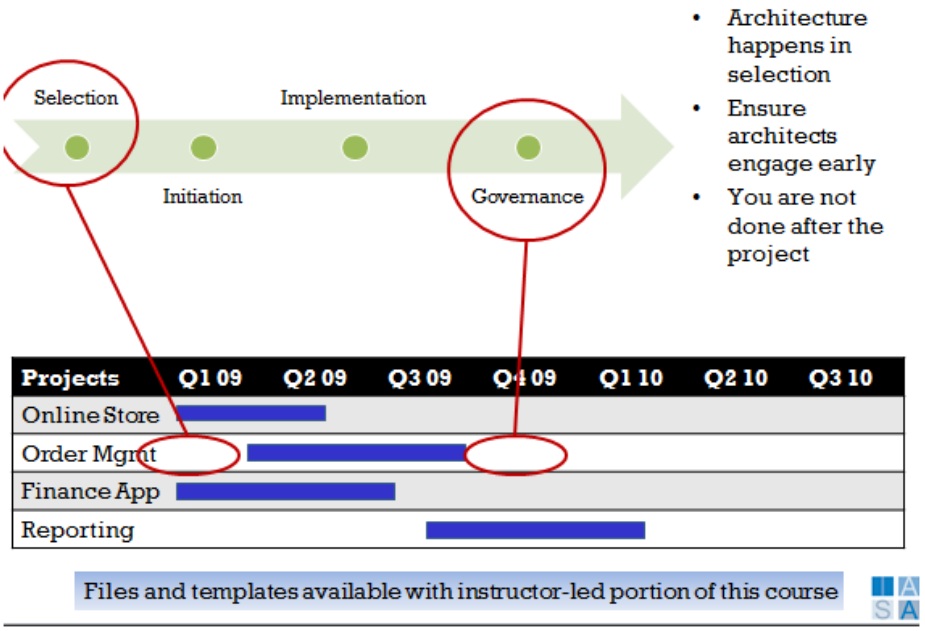


Figure 49

If you work for an architecture team, ask yourself how much coverage you are getting on engaging the entire company at all levels. Iasa likes to tell architects that until every employee, partner and consultant can describe the exact value of the architecture team and you are involved in everything that impacts the technology strategy of the company then you have not fully covered the company's architecture needs. Make that your goal and continue to work towards it.

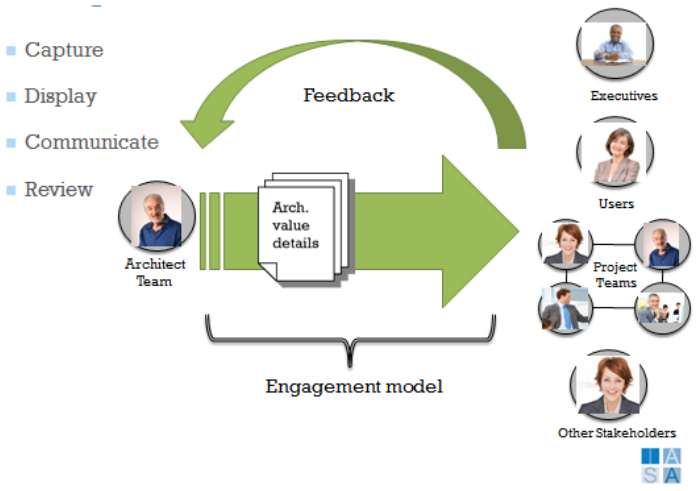


Figure 50

Again you should be managing your engagement model in a proactive manner. Even if the decision is to not have an engagement model for now you will have to update this decision

regularly. At a minimum once a quarter you should do a partial or full review of continued growth and strength of your engagement model including value, skills, education, delivery, process and coverage. This review(s) should be used to modify your model to best meet current needs and opportunities. And don't forget to market any changes back to the stakeholders! And yes you need to market it and not just inform.

Remember, you have an engagement model whether you manage it explicitly or not. Don't just get stuck with one. The first priority is to think about what would work best at your company. Research options and decide how to move it forward. The main take away motto is: 'Think Big, Start Small, and Move Fast'.

This completes the Iasa ITABoK which is the pre-work for the Iasa Architect Core course. You can use this e-book as your study guide for the open book CITA-Foundation exam deployed by Iasa Global. To find out more information on this certification and other offerings by Iasa, please visit www.iasaglobal.org.

THANK YOU!