

What is Architecture?





► Introduction and objectives

What is architecture?

IT Architects

Summary and references



At the end of this module, you should be able to:

- Explain what is meant by architecture and how we use it to help us design viable IT systems
- Understand why architecture is important in the modern age of fast-paced change
- Describe what an IT Architect is and what one does in the design on IT systems
- Describe the relationship between:
 - Enterprise architecture and solution architecture
 - Business architecture and IT architecture



Spend five minutes thinking about the following questions:

- What is your definition of architecture?
- What is the difference between architecture and design?
- Why is architecture important to IT systems design, and why do projects need it?
- What is the role of the IT Architect?
- What diagrams and models do you currently use to represent architectures?
- What do *you* think is meant by architectural thinking?



Here are some popular misconceptions about architecture:

- Architecture is just paper.
- Architecture and design are the same thing.
- Architecture and infrastructure are the same thing.
- <My favorite technology> is the architecture.
- A good architecture is the work of a single architect.
- Architecture is simply structure.
- Architecture can be represented in a single blueprint.
- Architecture cannot be measured or validated.
- Architecture is a science.
- Architecture is an art.



Module outline



Introduction and objectives

► **What is architecture?**

IT Architects

Summary and references



Architecture (as an artefact):

*The architecture of a system describes its overall **static structure and dynamic behaviour**. It models the **system's elements** (which for IT systems are software, hardware, and its human users), the **externally manifested properties** of those elements, and the static and dynamic relationships among them.*



Architecture (as a discipline):

*Architecture is an **engineering discipline** that studies methods of designing **IT systems** that provide **a solution to a business problem**. The solution must **satisfy** functional and non-functional **requirements** in a way that best balances **competing stakeholders' concerns** and must take **constraints** into account.*



Think about the following statements and see if you agree with them:

- I don't have time for architecture – it's just a document that no one needs anyway. My solution will work perfectly without the need to produce these documents and/or diagrams
- There are too many architectural artefacts that the method defines. I don't have time to complete all this work even if I think it's got value
- We do not need architecture today. Modern ways of delivery use agile methods that make architecture superfluous
- We use cloud based and "Platform as a Service" environments that "decide" on the architecture and spin out the necessary resources for me behind the scenes
- We use a "Commercial off the Shelf" product (i.e. a package) or "Software as a Service" (SaaS) cloud based solution. It has its own architecture that is hidden from me. Therefore I don't need to develop architecture
- We use Design Thinking workshops to work on the solution. This removes the need to develop architecture



What is the role of architecture? Some of the imperatives for producing architecture:

- Solution blueprint that can be reviewed and validated
- Communication tool for stakeholders and teams (business, IT delivery and BAU – IT Operations), future reference for maintenance and change, developing neighbouring systems and integration
- Delivery Environments: Input into the delivery environments' design and specification, tools and platforms of choice for solution delivery
- Delivery Planning: Starting point for developing the project plan (Traditional delivery), solution and feature backlog (Agile delivery), resource requirements and team structure
- Estimates: Starting point for cost estimation (labour, licences, platform costs, etc)
- Foundation for segregation of responsibilities between various parties – e.g. Client and Solution Provider, Solution Provider and Partner(s)/Subcontractor(s) (e.g. RACI matrices)



Why do we need it?

- In the age of continuing advances in technology and delivery methods (e.g. DevOps, Continuous Delivery, etc), architectures are as important as ever:
 - to ensure completeness and feasibility of functional capabilities
 - to account for Non-functional and Quality of Service requirements
 - within the context of often significant constraints
- Unless we are dealing with a very basic solution that has a very low level of complexity (e.g. a simple mobile or web app developed in isolation), lack of existing solution architecture often leads to:
 - misinterpretation of functional capabilities, non-functional qualities, integration required and the risks involved. This usually manifests itself towards the end of the delivery cycle at the times of testing (or even worse, at early production stages). A decision not to have architecture developed does not solve this problem but simply pushes it off to a later stage within the delivery cycle when the cost of rectification is significantly higher, if such rectification is achievable at all
 - poor understanding of the solution structure and how parallel development can be achieved and how inter-component dependencies can and should be managed



Why do we need it?

- Both Functional capabilities and Non-functional requirements & Service Levels may be contractually binding:
 - Failure to achieve targets may result in financial penalties and/or a loss of business for the Client
 - It is an issue of significant severity if an architecture that meets the specified requirements cannot be established. This may be due to lack of a viable solution in principle (e.g. we cannot solve a given problem with the amount of funding available) or due to poor design and the latter needs to be improved. Regardless, if the answer is negative, the sooner this is known the better for all the stakeholders
- The effort expended should always be proportionate to the risk and return involved. Without a developed architecture, it is rare that all significant risks are known and well understood
- Lack of architecture means that there is no basis for solution cost and delivery time estimates

Architecture can be at any scale, and the scale is defined by the characteristic dimension of the architecture's elements (1 of 4)



Paraphrasing the definition: **Structure** and **behavior** of the **elements** of a system

- What is a system?
- What is an element?



Architecture can be at any scale, and the scale is defined by the characteristic dimension of the architecture's elements (2 of 4)



Paraphrasing the definition: **Structure** and **behavior** of the **elements** of a system

- What is a system?
- What is an element?

It's all a question of **scale**:

- The elements of a hi-fi are its tuner, amplifier, speakers, cables, and remote controls.
- The elements of an amplifier are its case, the power supply, and the circuit board.
- The elements of a circuit board are its resistors, fuses, and integrated circuits.
- The elements of an integrated circuit are transistors and other components.



Architecture can be at any scale, and the scale is defined by the characteristic dimension of the architecture's elements (3 of 4)



Paraphrasing the definition: **Structure** and **behavior** of the **elements** of a system

- What is a system?
- What is an element?

It's all a question of **scale**:

- The elements of a hi-fi are its tuner, amplifier, speakers, cables, and remote controls.
- The elements of an amplifier are its case, the power supply, and the circuit board.
- The elements of a circuit board are its resistors, fuses, and integrated circuits.
- The elements of an integrated circuit are transistors and other components.



Whatever the scale of the system we're focused on, we say:

- 'Above' that scale are the '**requirements**' of the world in which the system sits.
- 'At' that scale is the '**architecture**' between the elements of the system.
- 'Below' that scale is the '**design**' of the insides of the elements of the system.



Architecture can be at any scale, and the scale is defined by the characteristic dimension of the architecture's elements (4 of 4)



Paraphrasing the definition: **Structure** and **behavior** of the **elements** of a system

- What is a system?
- What is an element?



It's all a question of **scale**:

- The elements of a hi-fi are its tuner, amplifier, speakers, cables, and remote controls.
- The elements of an amplifier are its case, the power supply, and the circuit board.
- The elements of a circuit board are its resistors, fuses, and integrated circuits.
- The elements of an integrated circuit are transistors and other components.



Whatever the scale of the system we're focused on, we say:

- '**Above**' that scale are the '**requirements**' of the world in which the system sits.
- '**At**' that scale is the '**architecture**' between the elements of the system.
- '**Below**' that scale is the '**design**' of the insides of the elements of the system.



So, in our world of IT:

- Our IT system's requirements are modeled as interactions (use cases, and so on.) with actors and other IT systems
- Our IT system's architecture's elements are components and nodes, and so on.
- Our IT system's design or designs model the insides of components, nodes, messages, and connections

Architecture also helps us ensure that our systems are good systems



Architecture is also used to describe best (or preferred) practice, documented as *architecture guidance*, that helps ensure our system is a good system.

Architecture guidance includes:

- Good things
 - Tried and tested parts
 - Tried and tested structures of parts (reference architectures)
- Good ways of doing things
 - Principles and patterns that help ensure the architecture is created in a good way

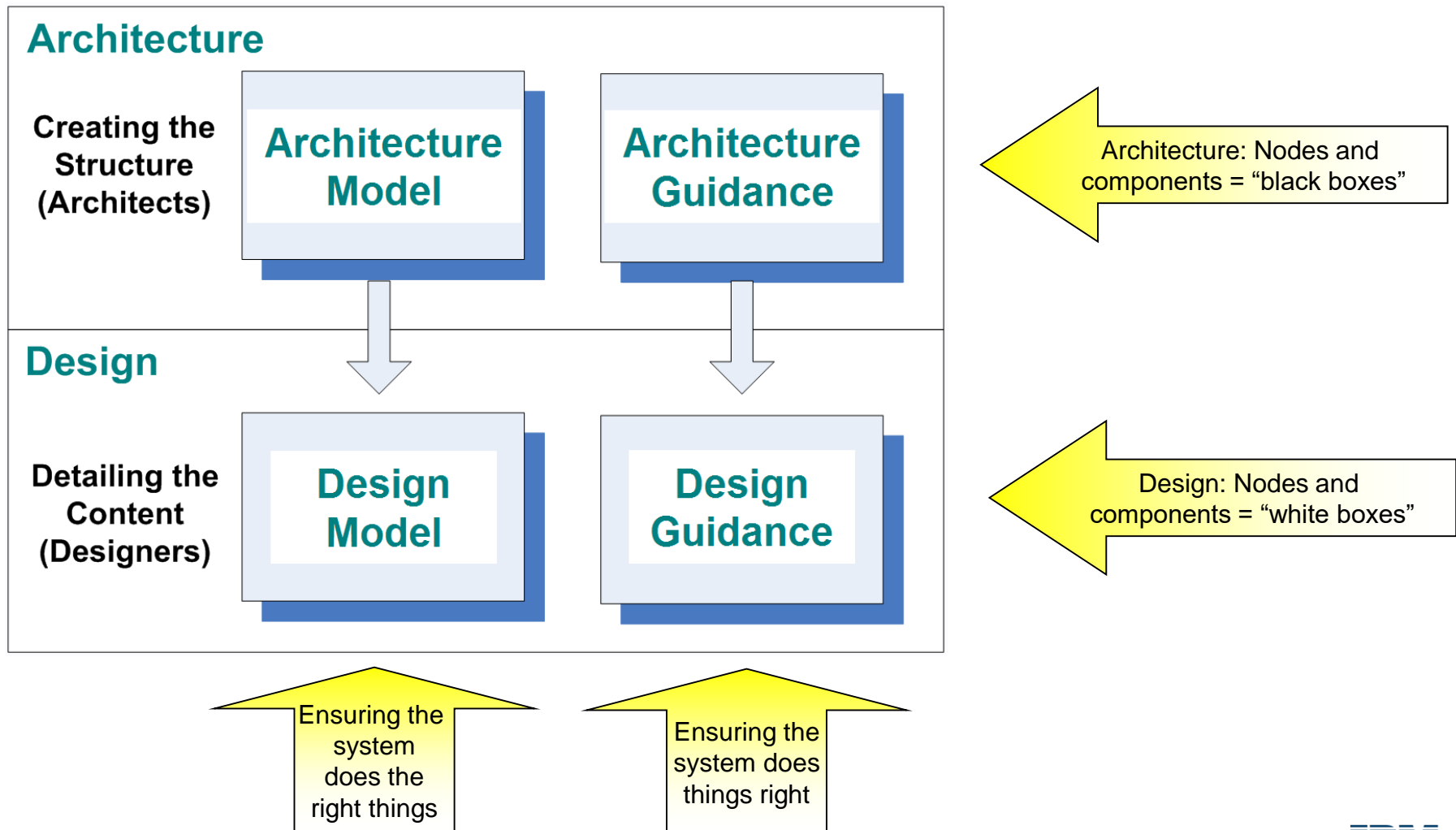
This architectural catalogue of guidance may be found in:

- An enterprise's own architecture asset databases
- The World Wide Web
- Reference Architectures and other design guidance available from hardware and software vendors

The focus of architecture and design on the target system



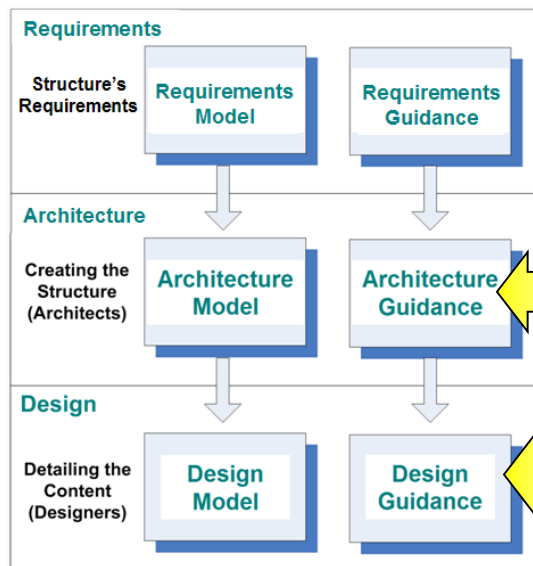
Overall, architecture and design both focus on the content and the context of the target system, ensuring that it does the right things and does things the right way.



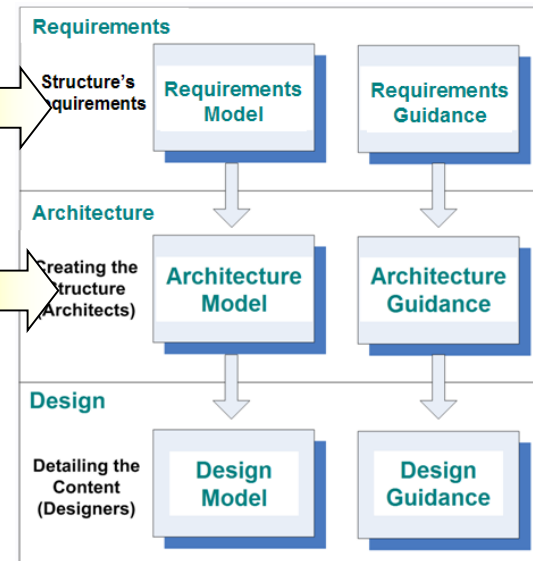
This simple framework can be extended, enabling us to see a repeating pattern across the scales



Architecting the Amplifier



Architecting the Circuit Board





Time for a question



What is the correct definition of architecting?

- A. The process of analyzing and solving, at a high level, technical problems using various business theories
- B. The process of analyzing and solving, at a high level, business problems using various technologies
- C. The process of solution design

What is the value of architecting?



Time for a question



What is the value of architecting?

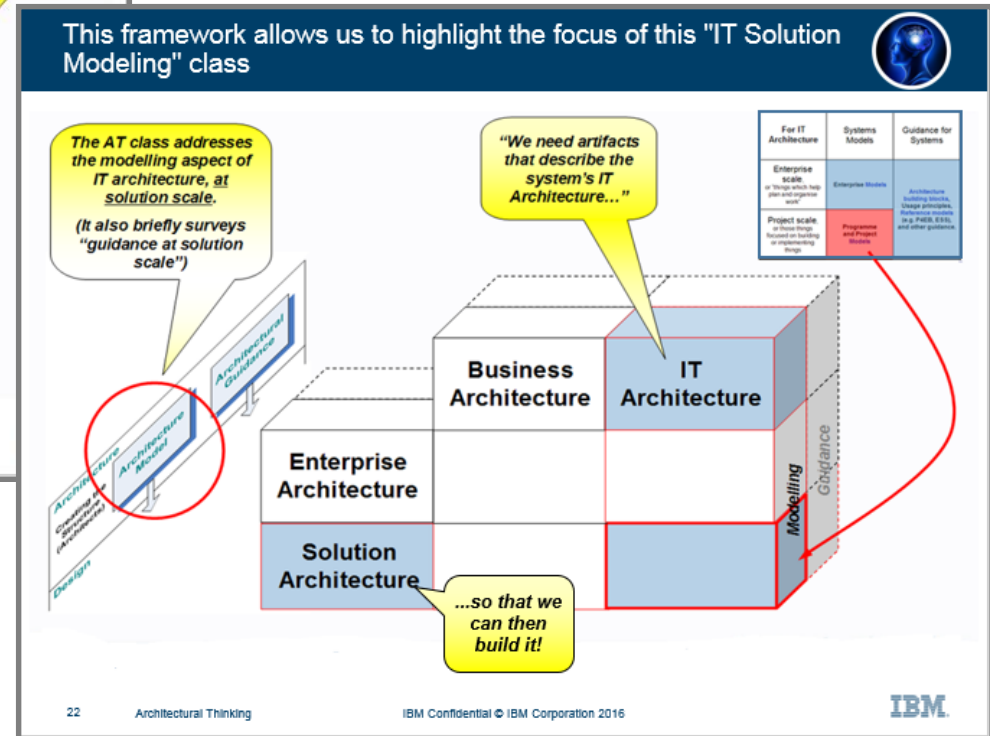
- A. To deliver the project on time and within budget
- B. To address a business problem with a technical solution involving technologies, people and processes
- C. To bring the vision of the most effective solution into reality

These multiple architectures can be simplified for each dimension, into a "2 by 2" reference framework



Four interlocked Architectural Domains...

This framework allows us to highlight the focus of this "IT Solution Modeling" class



Thus this class focuses on the models needed to describe the IT architecture of solution-level systems



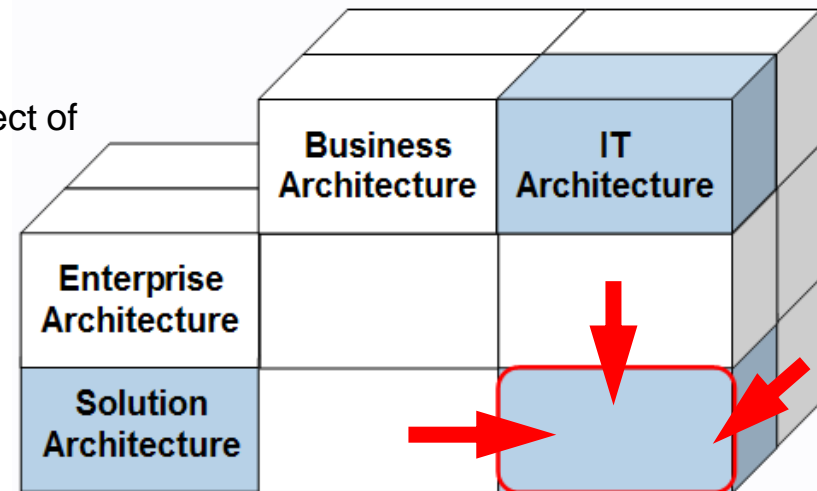
The artifacts covered in this class present models of the target IT system.

They can be **driven by models of the broader business system**, of which this target IT system is a part.

- The target IT system must meet the specific business requirements placed on it.
- Whether it meets those requirements or not is the subject of **Solution Assurance**.

They can be **guided by artifacts** documenting architectural principles, patterns, and building blocks.

- The target IT system must meet the wider architectural requirements placed on it.
- Whether it meets those requirements or not is the subject of **Architectural Conformance**.

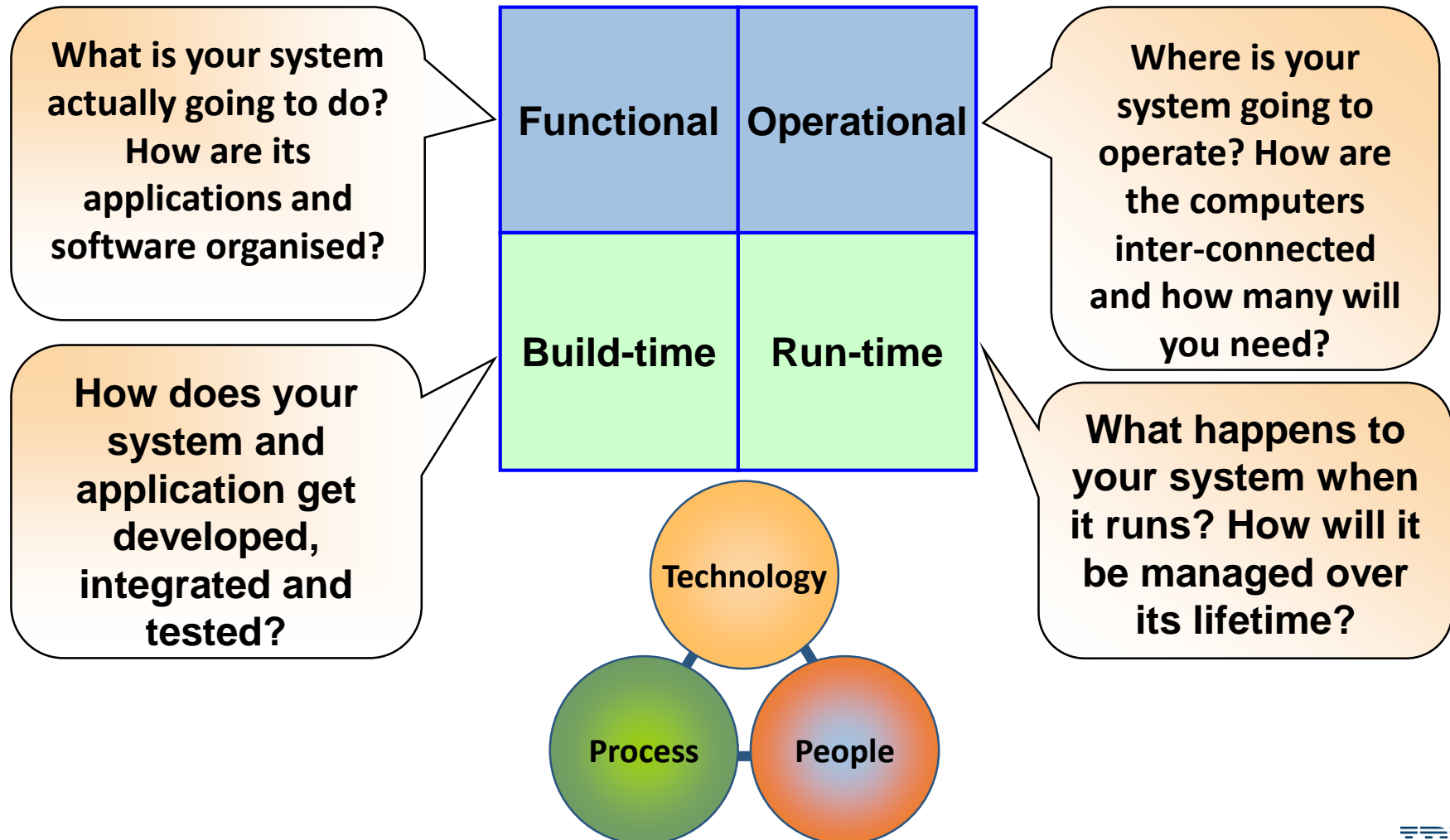


The class focuses on the use of these IT system model artifacts **at the solution scale**, by IT Architects.

- They are also used by Enterprise Architects at the enterprise scale as they plan (in outline) transition initiatives, and so on, in which case they may be **elaborated** by the IT Architect.

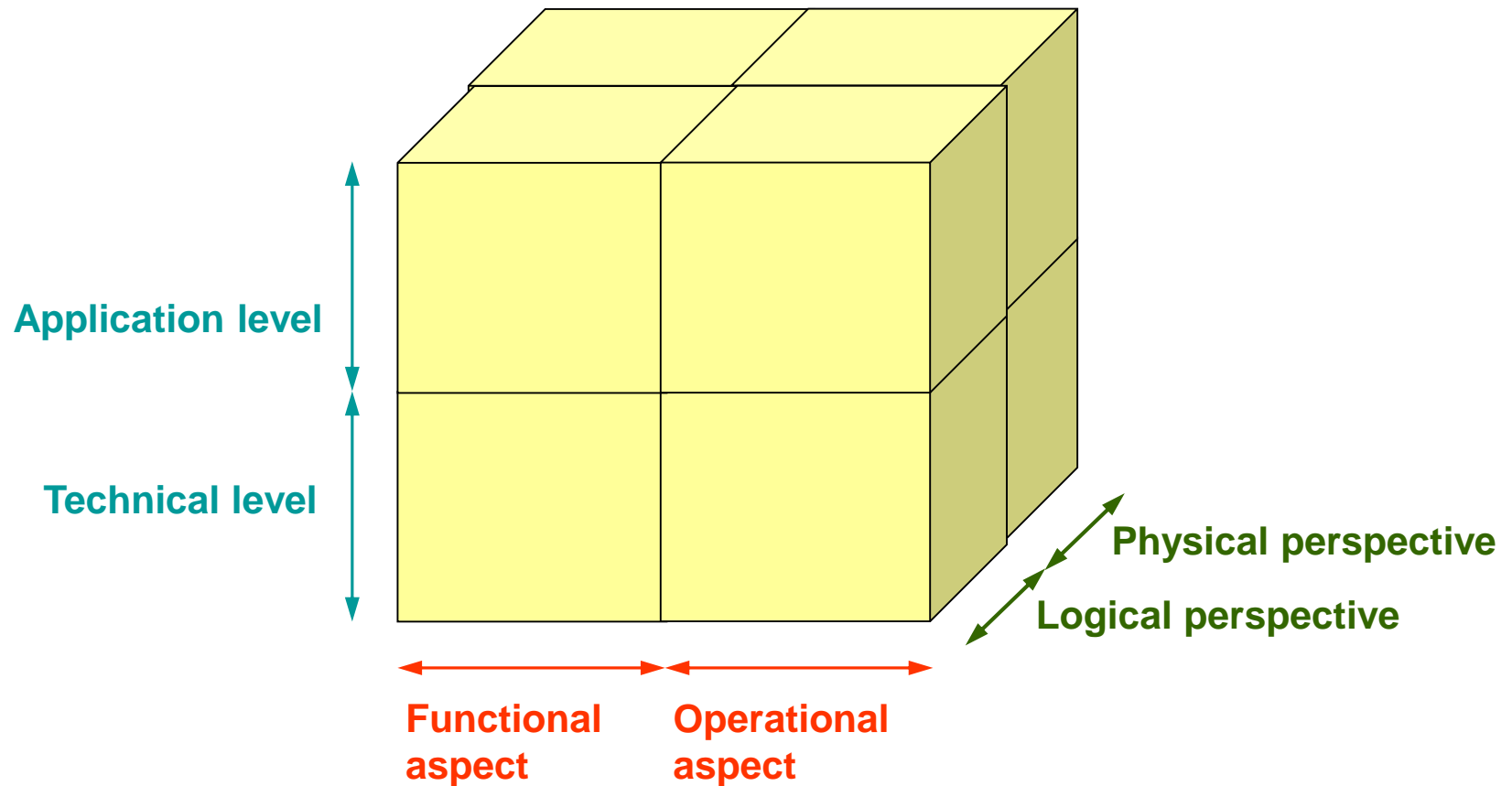


An architect asks questions for the different aspects each with technology, process and people to deliver a complete architecture





Architects look at a problem in a number of dimensions to organise their thoughts



Module outline



Introduction and objectives

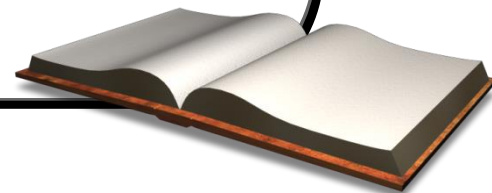
What is architecture?

► **IT Architects**

Summary and references



Architect: An Architect *defines the structures of solutions* and architectures to address client business problems. The Architect *understands client needs and business constraints*, works in levels of abstraction, applies industry knowledge, and *leverages appropriate business elements and information technology to address those needs*. The solutions are manifested as *business, enterprise, or IT architectures* and can include systems, applications, and process components. They may also involve the application and integration of a broad variety of products, technologies and services, and various architectural styles and domains.





An Architect defines the structures of solutions... [based on] client needs and business constraints... [leveraging] appropriate business elements and information technology to address those needs. ...Solutions [can address] business, enterprise, or IT architecture... domains.

Architects work:

- At the front:
 - Listening to clients
 - Understanding business requirements
 - Negotiating feasible requirements and scope
 - Forming detailed definitions of solution structures
- At the back:
 - During construction
 - Serving as a customer advocate
 - Guiding the team

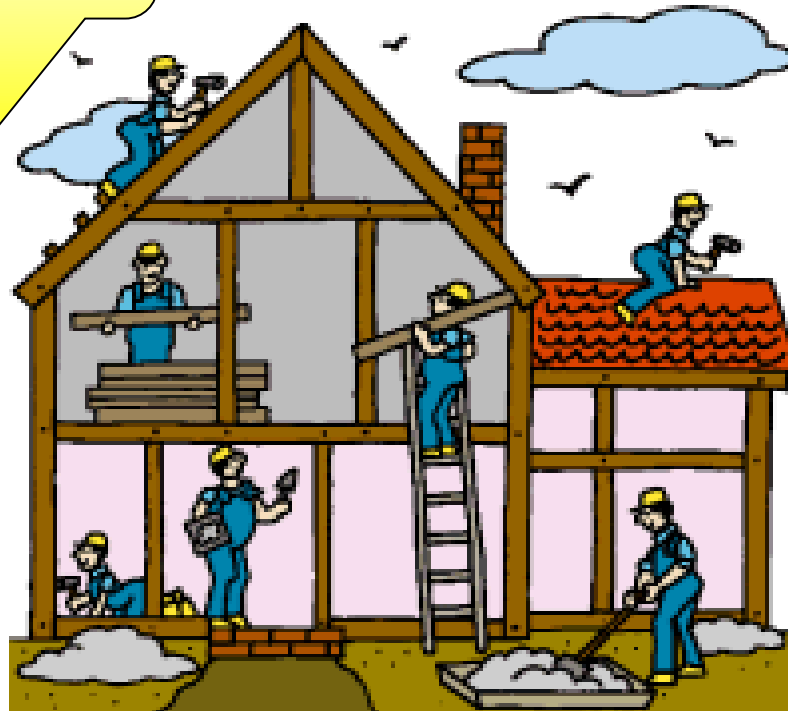
Who is needed to create a new house?



Architect

Many specialists are required to construct the house: Framers, Plumbers, Electricians, and Painters.

Builder, Subcontractors



Occupants

Occupants will live in the house and have wants and needs.

The Architect draws up the plan to meet the occupants' needs and ensures that the builders meet the specifications, understands the influence of building codes, other regulations, legacy, power and gas services, and the neighborhood architecture review committee guidelines.



Building Inspector

The Inspectors ensure that the house is built as per quality standards such as local building codes.

The Project Manager manages the construction of the house and coordinates subcontractors.

Project Manager



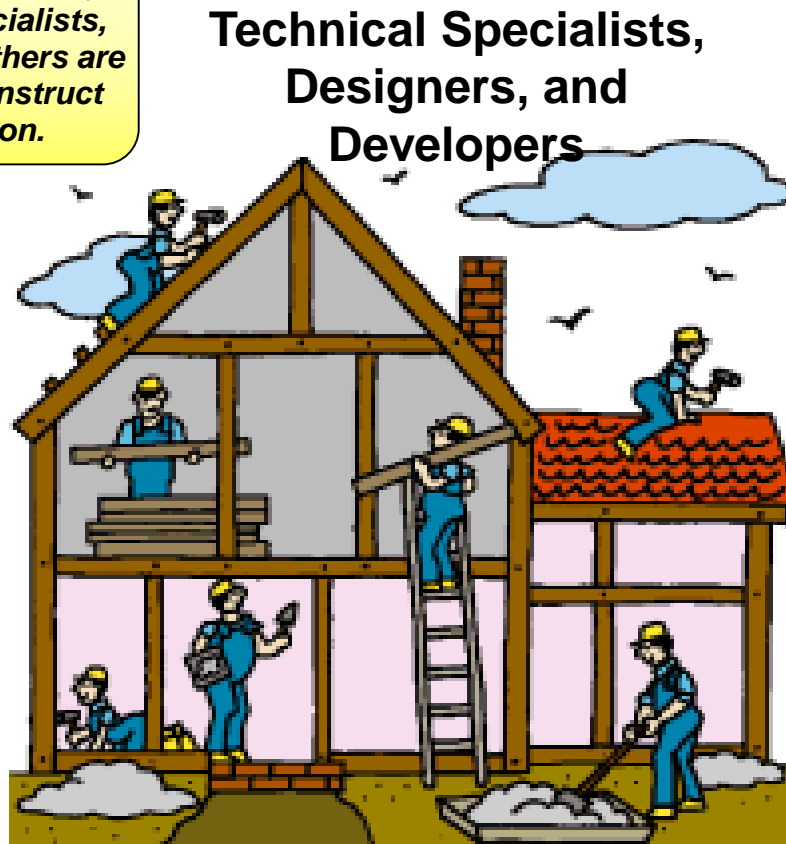
What's this got to do with architecture?



Solution Architect

Designers, Coders, Package Specialists, Testers, and others are needed to construct the solution.

The Architect creates the solution architecture to meet requirements and ensures that the development meets the specifications, regulations, and passes governance checkpoints.



Technical Specialists, Designers, and Developers



Business Sponsor, Users

Users and their delegates have their requirements.



Enterprise Architect

The Enterprise Architect ensures that the solution architect complies with enterprise standards, governance, and roadmaps.

The Project Manager manages the Solution Delivery.

Project Manager



Roles work together to form a project team



Architect

Defines and owns
the technical or
business solution

Project Manager

Develops and
owns the project
plan

Designer/ Developer

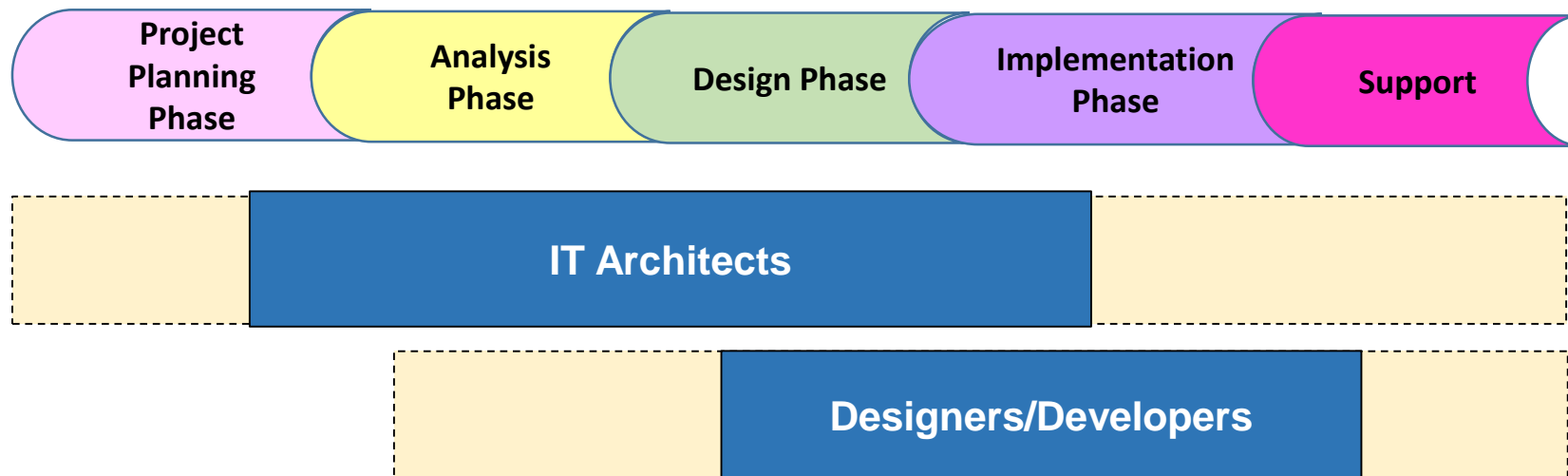
Develops and owns
the delivered
solution

Each team member contributes to the development of key artifacts for delivery excellence.

The transfer between team members is a critical component of success.

Everyone should have common goals and objectives.





IT Architects

- Define solutions to client business problems. Resultant architectures can include systems, applications, and process components.
- Focus on the front end of the solution life cycle. May be involved in the construction of the solution as a client advocate.

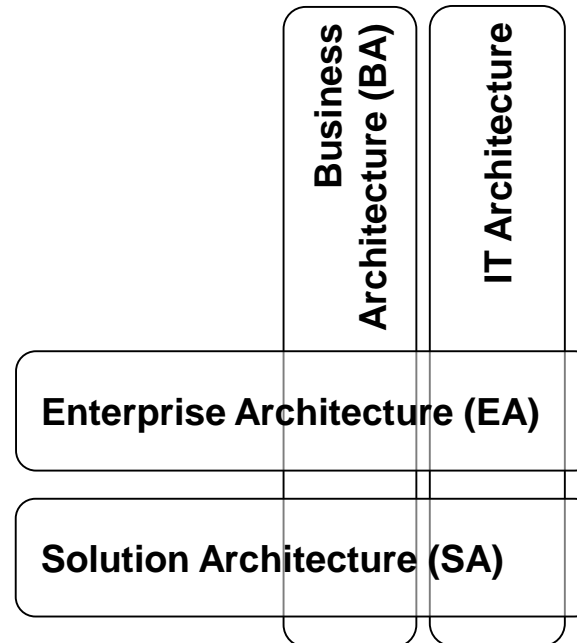
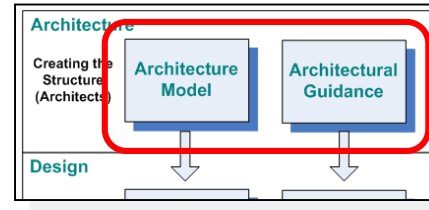
Designers/Developers

- Develop proof of concept, design, develop, build, test, and implement systems
- Are hands-on professionals with in-depth understanding of products, offerings, and services within their specialty

Architect supports two distinct dimensions (1 of 5)



Embracing both models and guidance:

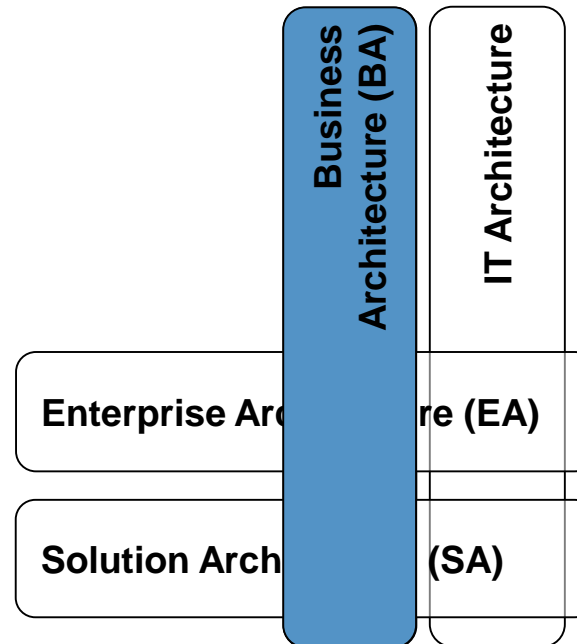
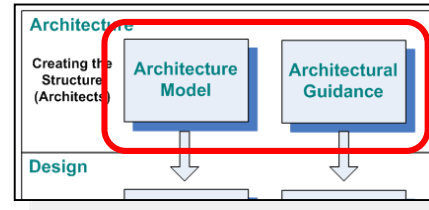


Architect supports two distinct dimensions (2 of 5)



Embracing both models and guidance:

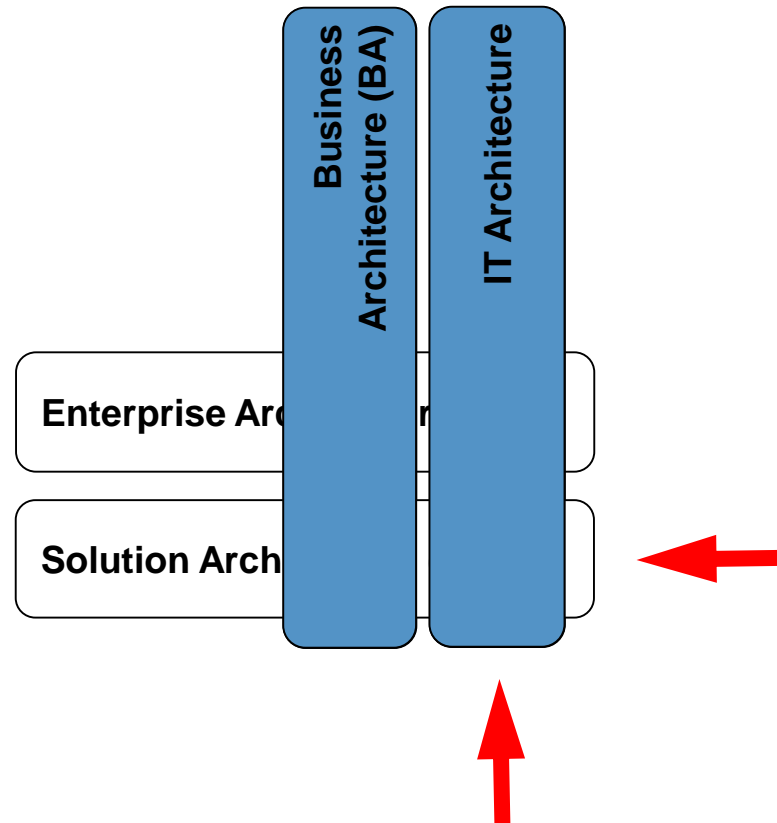
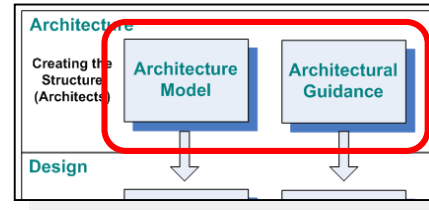
- Business Architects “do” BA
 - *At both EA and SA levels.*





Embracing both models and guidance:

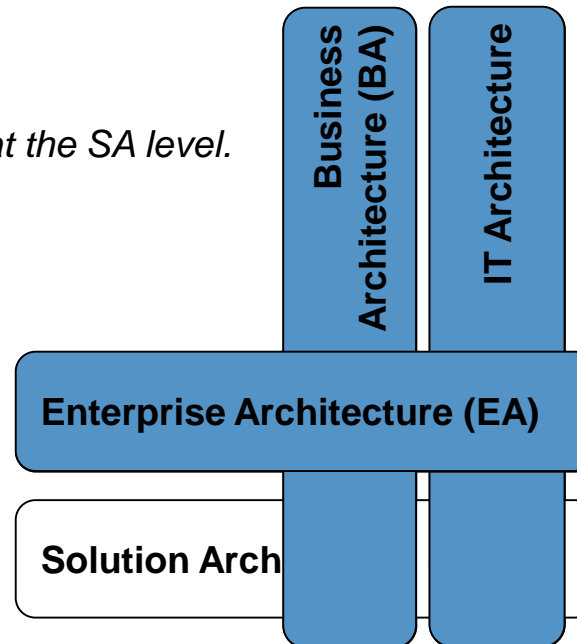
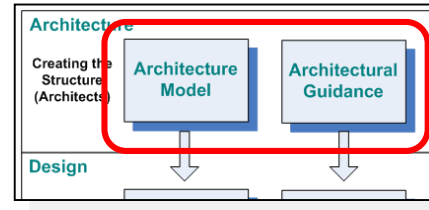
- Business Architects “do” BA
 - *At both EA and SA levels.*
- IT Architects “do” ITA
 - *Mostly at the SA level.*





Embracing both models and guidance:

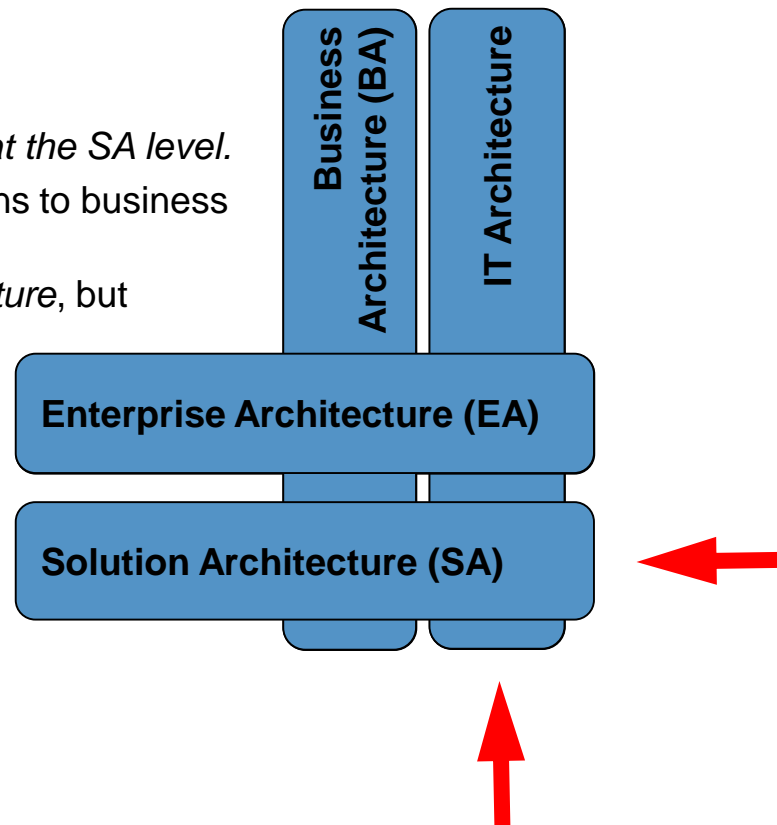
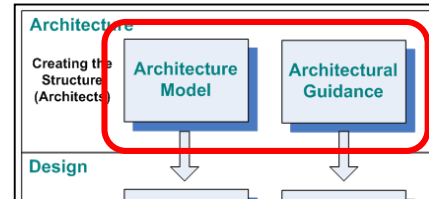
- Business Architects “do” BA
 - *At both EA and SA levels.*
- IT Architects “do” ITA
 - *Mostly at the SA level.*
- Enterprise Architects “do” EA
 - *Across both BA and ITA.*
 - *They often own guidance at the SA level.*





Embracing both models and guidance:

- Business Architects “do” BA
 - *At both EA and SA levels.*
- IT Architects “do” ITA
 - *Mostly at the SA level.*
- Enterprise Architects “do” EA
 - *Across both BA and ITA.*
 - *They often own guidance at the SA level.*
- Solution Architect create solutions to business problems – SA
 - *Typically within IT Architecture, but*
 - *Can create BAs too*





Time for a question



Which of the following choices is the better description of an Architect?

- A. A practitioner, a results-oriented person, a good communicator, a leader
- B. A “top-level” software designer, a project manager, a technology expert, a product expert, a lone scientist

Module outline



Introduction and objectives

What is architecture?

IT Architects

► **Summary and references**



- The architecture of a system describes the structure and behavior of its parts.
- The parts of an IT system are its software and hardware elements, interacting with the IT system's human and IT system users.
- An IT Architect defines the models and guidance needed to model IT or business systems to meet the business intent.
- When working on an IT system solution architecture, the IT Architect designs IT solutions to client business problems.

Back Up Slides: architecture vs Design



It is also possible to conceive of architecture and design in a rather different, possibly confusing, way (1 of 2)



In this alternative framework:

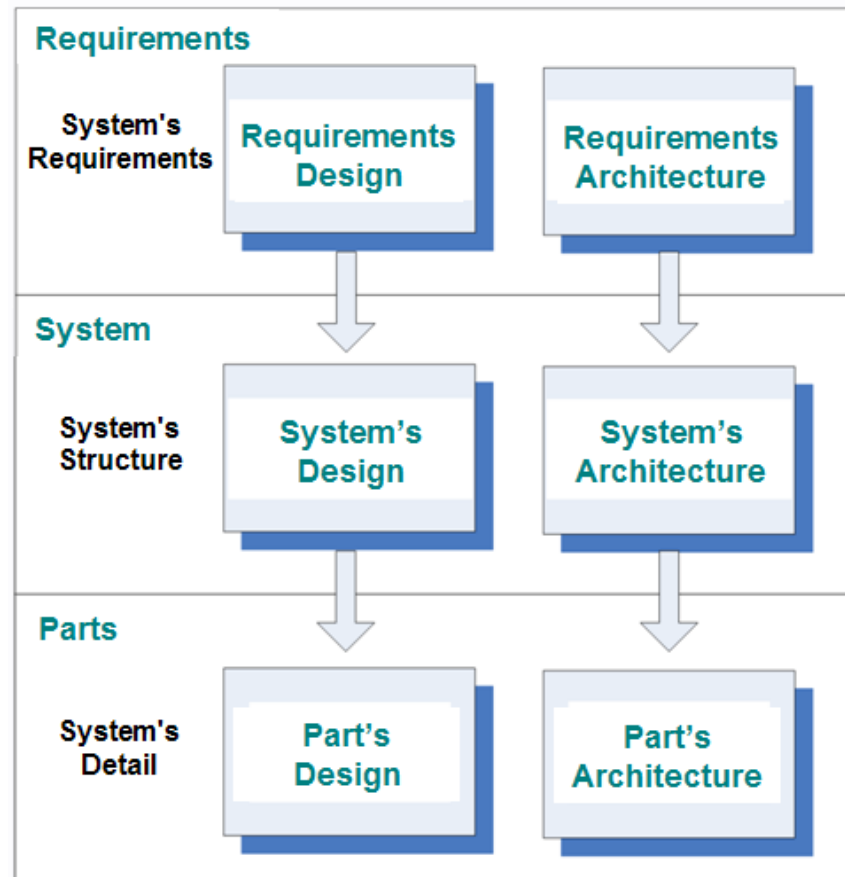
- Ensuring the system meets its specific requirements is called design.
- Ensuring it is designed in the right way is referred to as its architecture.

Ask yourself:

- What is the design of a bridge?
- What is the architecture of a bridge?

And:

- How many bridge architectures are there?
- How many bridges (that is, bridge designs) are there?



It is also possible to conceive of architecture and design in a rather different, possibly confusing, way (2 of 2)



In this alternative framework:

- Ensuring the system meets its specific requirements is called design.
- Ensuring it is designed in the right way is referred to as its architecture.

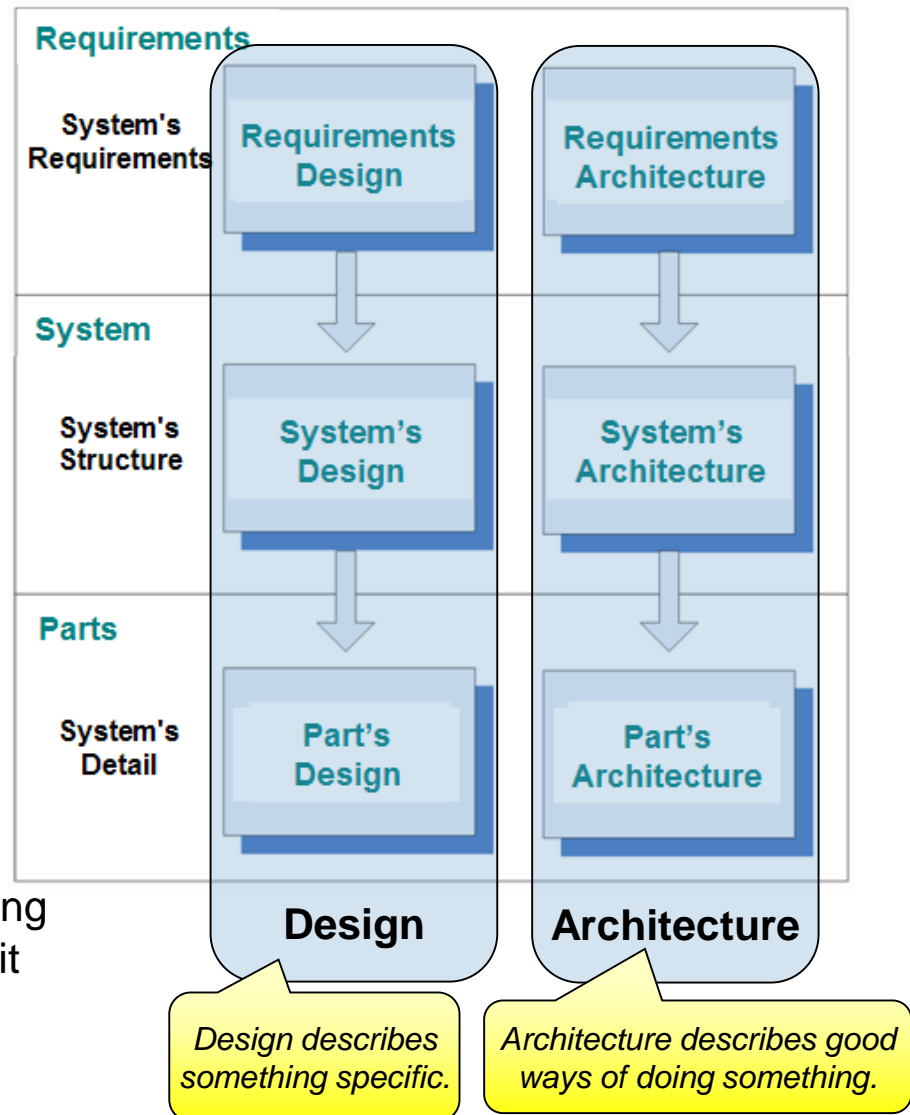
Ask yourself:

- What is the design of a bridge?
- What is the architecture of a bridge?

And:

- How many bridge architectures are there?
- How many bridges (that is, bridge designs) are there?

This terminology is common, particularly among our clients and in enterprise architecture, but it has not been adopted by IBM's Architect profession.





[COOK07] Cook, Denise, Cripps, Peter, Spaas, Philippe.

[An Introduction to the IBM Views and Viewpoints Framework for IT Systems](#), IBM,
December 2007

THANK YOU



IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "[Copyright and trademark information](#)." Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and its affiliates.

Linux is a trademark of Linus Torvalds in the United States, or other countries, or both. Microsoft, Excel, PowerPoint, Project, Visio, Word, and Windows are trademarks of the Microsoft Corporation in the United States, other countries, or both. UML is a registered trademark of Object Management Group, Inc. in the United States and/or other countries. TOGAF and UNIX are registered trademarks of The Open Group in the United States and other countries. Other company, product, and service names may be trademarks of IBM or other companies. Citrix is a trademark of Citrix Systems, Inc. and one or more of its subsidiaries, and may be registered in the United States Patent and Trademark Office and in other countries.