Massachusetts Institute of Technology

Architecture of Complex Systems
WEEK 5: SYSTEM ARCHITECT

Key Takeaways

Week 5 took a step back and positioned you in the role of the architect. You looked at several key ideas that outlined the role in terms of deliverables and tasks. These will help you through the product development process.

Architects manage key design trade-offs in the system. They define the vision of the system and communicate that vision to stakeholders. In order to be successful, architects must perform the following three key tasks:

- **Reduce ambiguity:** Define the boundaries, goals, and functions of the system.
- **Employ creativity:** Create the concept of the product that captures the essential vision of the system.
- **Manage complexity:** Represent the system to ensure it is comprehensible to all.

More on ambiguity

Architects drive ambiguity from the upstream processes. They interpret upstream processes to deliver the system boundary and a concrete set of goals while providing feedback upstream. Some tasks include interpreting corporate and functional strategies, interpreting competitive analysis, listening to stakeholders, taking enterprise considerations into account, and recommending standards, among others. In essence, architects identify the necessary, consistent, and important information.

Types of ambiguity:

- **Fuzziness**: When an activity is subject to more than one interpretation.
- **Uncertainty**: When an event's exact outcome is uncertain, like a coin toss. You know the potential outcomes very clearly, but the exact outcome is uncertain.
- **False information**: When you are presented with incorrect information.
- **Unknown information**: When you do not have all of the information needed to make an informed decision.
- **Conflicting information**: When two or more pieces of information oppose each other.

More on creativity

Once the system's goals have been defined, architects employ creativity to create a concept for the system. Some of the tasks in defining a concept are:

- Proposing and developing concept options;
- Identifying key metrics and drivers;
- Selecting a concept;
- Listing possible failure modes and creating mitigation/recovery plans.

More on managing complexity

Architects manage complexity and its evolution in the system to ensure the system goals are achieved. Information is managed and represented in the final architecture. This can be achieved by (please note: this is not a comprehensive list):

- System decomposition of form and function;
- Defining allocation of function to the elements of form;
- Interface definition;
- Subsystem configuration;
- Product evolution control.

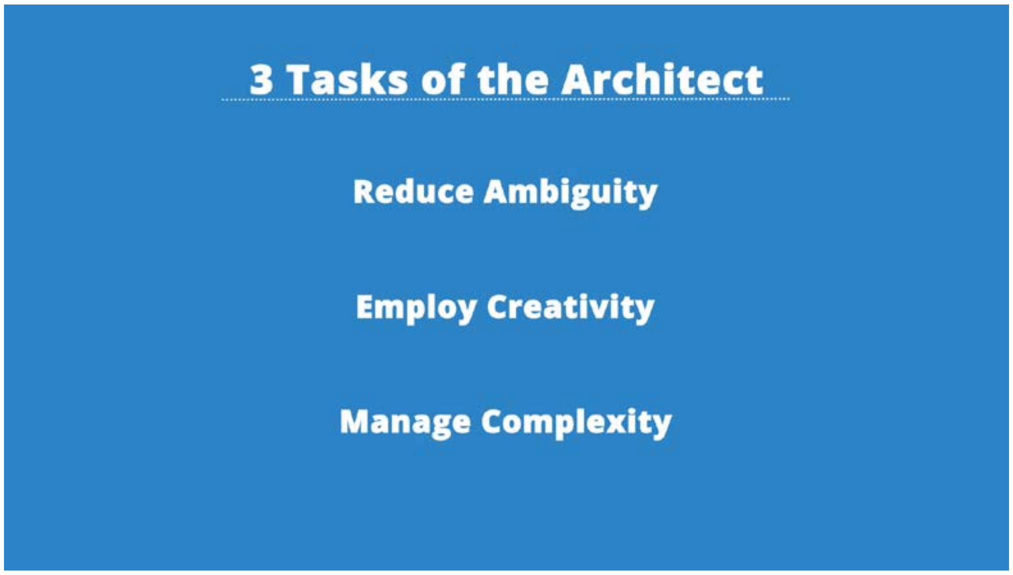
Deliverables of the architect

- An architect's role is to deliver the following:
- A clear, complete, consistent, and attainable set of goals (with an emphasis on functional goals).
- A description of the broader context in which the system will sit and the whole product context.
- A concept of the system.
- A concept of operations for the system, including contingency and emergency operations.
- A functional description of the system, with at least two layers of decomposition, including description of primary and secondary externally delivered function; process flow with internal operands and processes, including non-idealities, supporting processes, and interface processes, with a process to ensure that the functional decomposition is followed.
- The decomposition of form to two levels of detail, the allocation of function to form, and the structure of form at this level.
- Details of all external interfaces and a process for interface control.
- A notion of the developmental cost, schedule, risks, and the design and implementation plan.

Finally, in Week 5 you reviewed three key **architecture frameworks** used in the industry:

- **Zachman**: An ontology presented in a 6x6 matrix that helps describe the enterprise.
- **DoDAF**: This framework delivers accurate architecture descriptions to the specific stakeholders through models or viewpoints organized by various views.
- **TOGAF**: An iterative model that provides the tools to produce, leverage, maintain, and implement architectures.

Role of the Architect>Overview of the Role of the Architect>



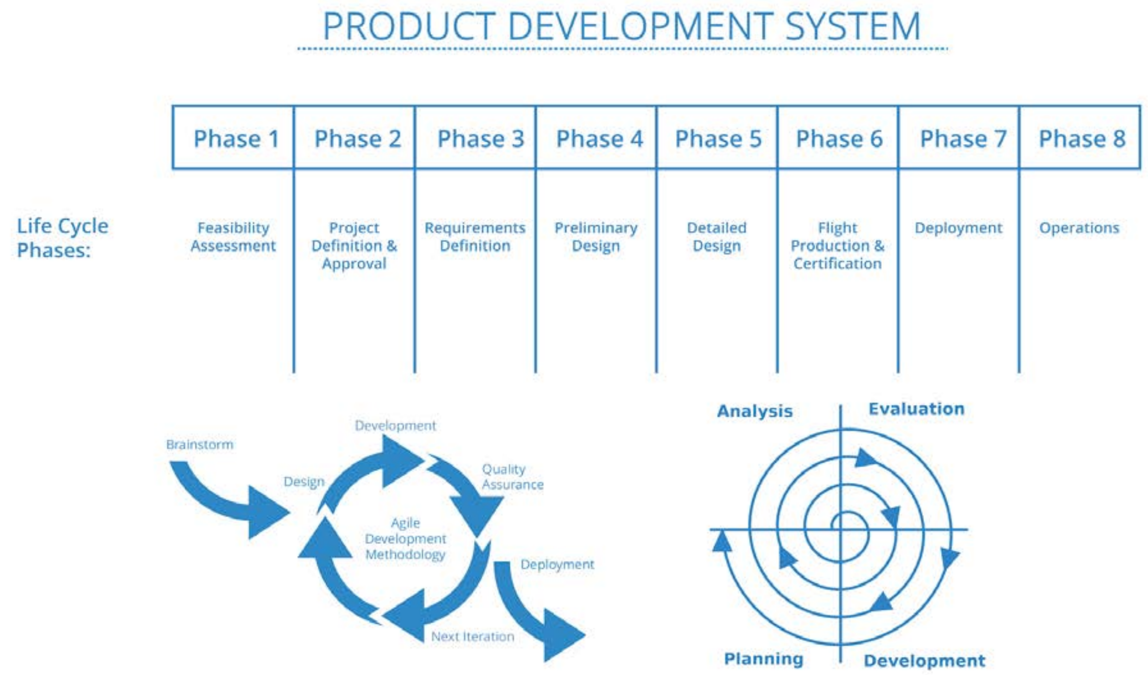
3 Tasks of the Architect

Reduce Ambiguity

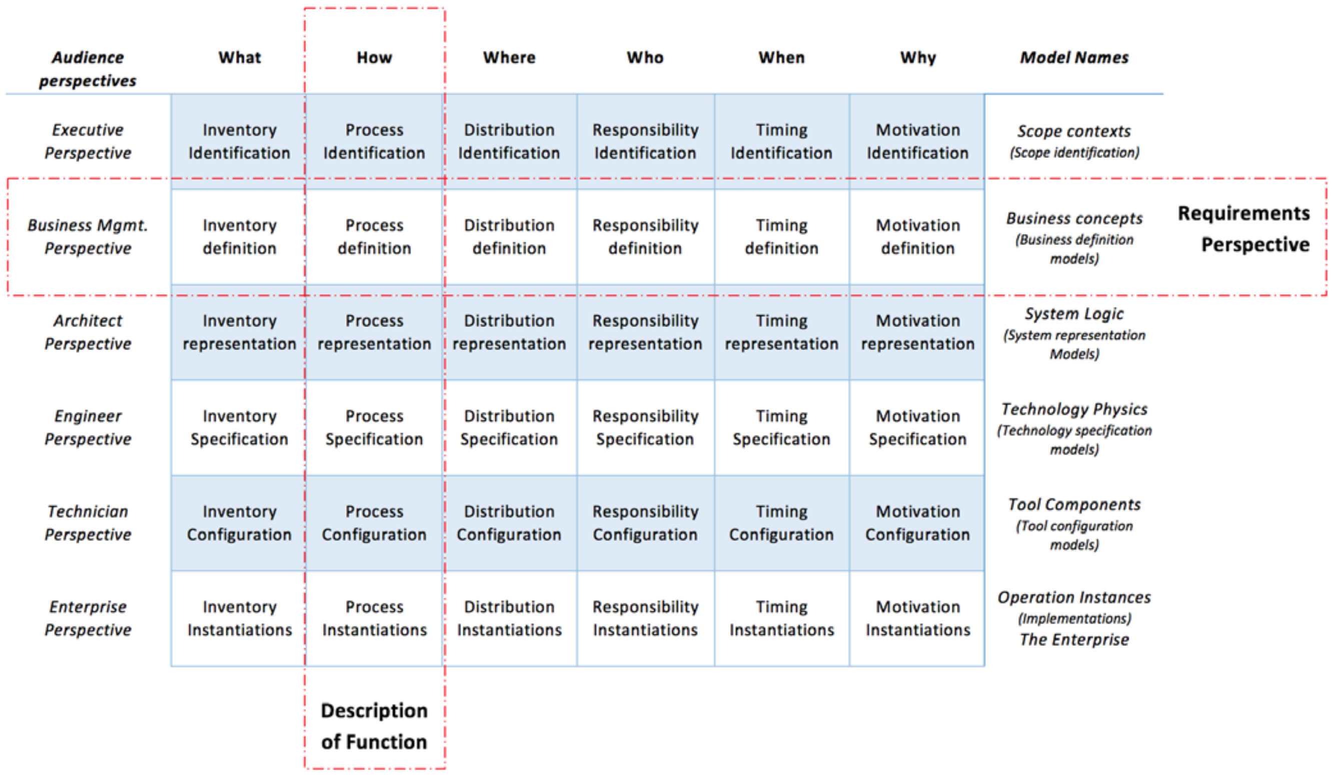
Employ Creativity

Manage Complexity

Deliverables of the Architect>The Architect's List>

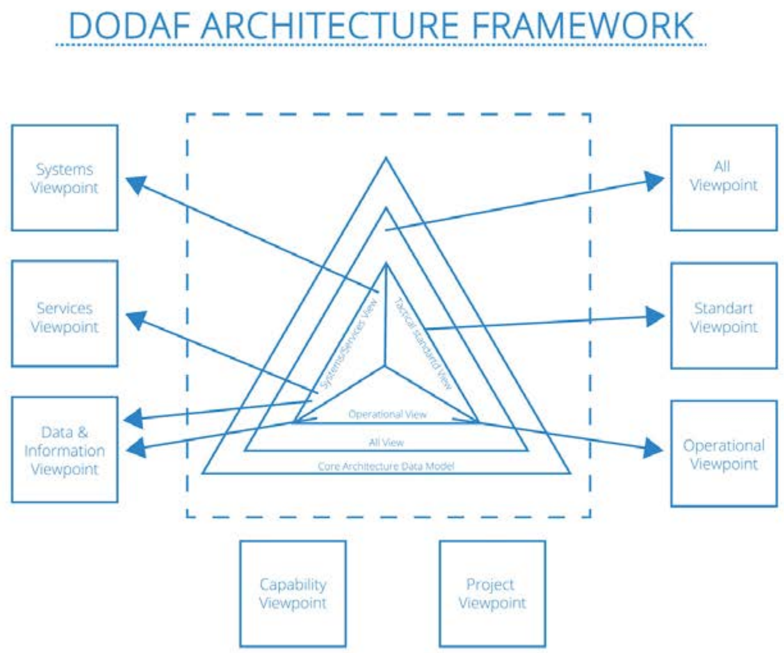


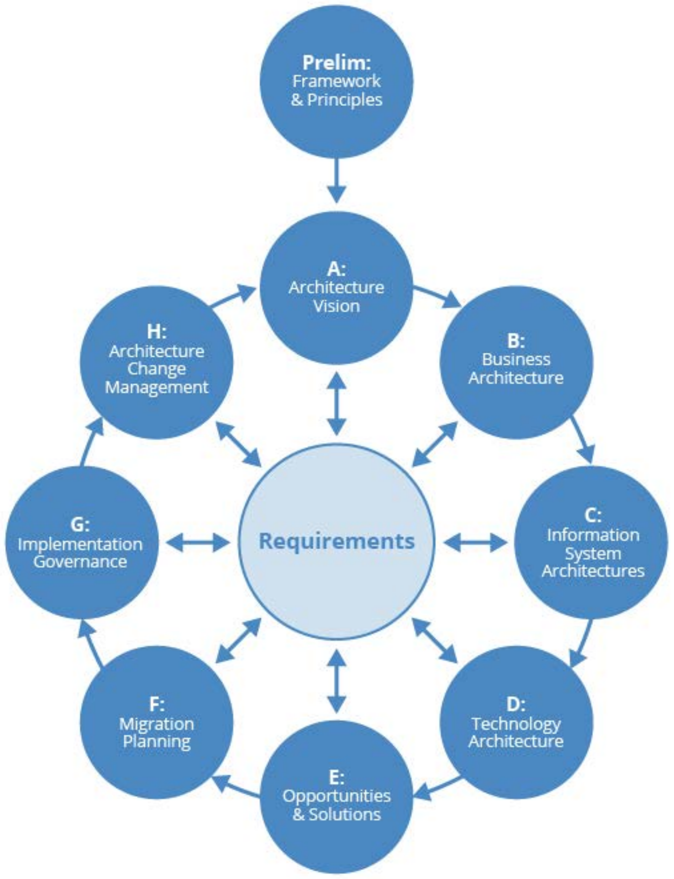
Architecture Frameworks>The Zachmann Framework>



Audience perspectives	**What**	**How**	**Where**	**Who**	**When**	**Why**	***Model Names***	
Executive Perspective	Inventory Identification	Process Identification	Distribution Identification	Responsibility Identification	Timing Identification	Motivation Identification	*Scope contexts (Scope identification)*	
Business Mgmt. Perspective	Inventory definition	Process definition	Distribution definition	Responsibility definition	Timing definition	Motivation definition	*Business concepts (Business definition models)*	**Requirements Perspective**
Architect Perspective	Inventory representation	Process representation	Distribution representation	Responsibility representation	Timing representation	Motivation representation	*System Logic (System representation Models)*	
Engineer Perspective	Inventory Specification	Process Specification	Distribution Specification	Responsibility Specification	Timing Specification	Motivation Specification	*Technology Physics (Technology specification models)*	
Technician Perspective	Inventory Configuration	Process Configuration	Distribution Configuration	Responsibility Configuration	Timing Configuration	Motivation Configuration	*Tool Components (Tool configuration models)*	
Enterprise Perspective	Inventory Instantiations	Process Instantiations	Distribution Instantiations	Responsibility Instantiations	Timing Instantiations	Motivation Instantiations	*Operation Instances (Implementations) The Enterprise*	
		Description of Function						

Architecture Frameworks>Department of Defense Architecture Framework>





Public Domain Image Courtesy of Stephen Marley (NASA/SCI) on Wikipedia.