ESD.34 - System Architecture
IAP 2007

Professor Edward F. Crawley
Ford Professor of Engineering
Professor of Aeronautics and Astronautics
Professor of Engineering Systems

Definitions

Architecture (alt):

The structure, arrangements or configuration of system elements and their internal relationships necessary to satisfy constraints and requirements. (Frey)

Architecture (alt):

The arrangement of the functional elements into physical blocks. (Ulrich & Eppinger)

Architecture (alt):

An abstract description of the entities of a system and the relationships between those entities. (Crawley, et al. 2005)

Architecture (reference):

The embodiment of concept, and the allocation of physical/informational function to elements of form, and definition of structural interfaces among the elements and with the surrounding context. (Crawley)

Holistic:

Of the whole. To think **holistically** is to encompass all aspects of the task at hand, taking into account the influences and consequences of anything that might interact with the task.

Principles, Processes and Tools:

Architects use **principles**, **processes** and **tools**. **Principles** are the underlying and long enduring fundamentals that are always, or almost always valid. . **Processes** are the organization of methods and tasks to achieve a concrete end, which should be solidly grounded on principles. Processes are usually applicable. **Tools** are the contemporary ways to facilitate process.

Product:

TBD (to be filled in as an assignment)

System:

TBD (to be filled in as an assignment)

Complex:

TBD (to be filled in as an assignment)

Value:

TBD (to be filled in as an assignment)

Interface:

The points at which elements of form connect.

The **Product Development Process**:

The inclusive process of creating a new or modified product, bringing it to "market" and supporting its life-cycle.

Professor Edward F. Crawley
ESD.34 - System Architecture IAP 2007

The **Architect and Architecting:**

Every system has an architecture. The architecture of built systems is defined by the "architect." **Architecting** is a function. It may or may not be associated with a person explicitly called "the **architect**".

Architecting is the most abstract, highest level function in product/system development process. **Architecting**

- is done by the smallest number of people (sometimes by one),
- has some of the greatest impact on eventual success,
- factors in the greatest number of considerations,
- is not primarily concerned with detailed or quantitative data.

An **architect** must be able to think holistically, and:

1. Define boundaries, and establish goals and functions,
2. Create the concept which maps function to physical/logical elements
3. Define decomposition, abstraction, hierarchy and inter-element interfaces.

An **architect** is not a generalist, but a specialist in simplifying complexity, resolving ambiguity, and focusing creativity.

Architecting occurs at three identifiable scales:

- Architecting a product to transfer to a customer (the reference scale)
- At a smaller scale, architecting an element of a product (i.e a system for an internal "customer")
- At a larger scale, which encompasses considering the product within a family, and in the context of the technology, design and manufacturing competence of the enterprise.

Architecting is universal:

- It applies to products (the reference case)
- It applies to technical design and manufacturing processes
- It applies more generally to process and organizations
- It applies to any endeavor in which humans participate as a creative force

The approaches that architects follow during their work are varied. Generally speaking:

- no single method will work,
- out of the box thinking often bears fruit,
- must (in principle) be able to deal with and consider everything,
- must concentrate on and trade the essential things.

Professor Edward F. Crawley
ESD.34 - System Architecture IAP 2007

Roles of the Architect

1. The architect drives ambiguity from the upstream process by defining the boundary of the "product/system" which constitute the system and its design and implementation process. Specifically, the architect interprets the needs, defines the goal(s) and function(s) by:

    - interpreting corporate and functional strategies
    - interpreting competitive analysis
    - listening to "customers" or their representative
    - considering the competence of the enterprise
    - infusing technology available
    - interpreting regulatory and pre-regulatory influences

2. The architect creates the concept for the system, consisting of internal function and form.

    - proposes and develops options,
    - identifies key metrics and drivers,
    - conducts highest level trades, and optimization
    - thinks holistically about the entire product life cycle in terms of
        - design
        - implementation (sourcing and manufacturing)
        - operation
        - evolution
        - product and process
        - risk management
        - sustainability
    - anticipates failure modes and plans for mitigation and recovery

3. The architect manages the evolution of complexity so that goals are met and function is delivered, while the system is comprehendible to all during its design, implementation, operation and evolution.
    - decomposes form and function
    - allocates functionality to elements
    - defines interfaces between subsystems,
    - configures the subsystems - creates the structure of the system while considering:
        - flexibility vs. optimality
        - modularity vs. platform
        - clean slate design vs. reuse of legacy
        - vertical vs. horizontal strategies, and
        - in-house vs. outsourcing design and manufacturing

Professor Edward F. Crawley
ESD.34 - System Architecture IAP 2007

Deliverables of the Architect

The architect will deliver

- A clear, complete, consistent and attainable (with 80%-90% confidence) set of goals (with emphasis on functional goals)
- A description of the broader context in which the system sits, and the whole product context
- A functional description of the system, with at least two layers of decomposition
- A concept for the system
- A design for the form of the system, with at least two layers of decomposition
- A notion of the timing, operator attributes, cost, risks, and the implementation and operation plans
- A document or process that ensures functional decomposition is followed, and the form at interfaces is controlled.