

WHAT IS AN ARCHITECTURE?

Suzanne Barber

What does an Architect do?

...converts customer (User) wants and needs into working software.

“Converts”: ...transforms requirements into artifacts (blueprint specification, code, reports, GUI's, user guides...)

“Working”: 20-20-20 yield conundrum for software architects

- 20% of all code written finds its way into a finished program
- 20% - of finished programs are delivered to a user
- 20% - of the features in delivered programs are doing 80% of the work (feature use)

“Working” yield might be as low as 0.008%

Architectures ...

- Architecture starts when you *carefully put two bricks together*.
- Architecture is not an inspirational business, it's *a rational procedure to do sensible and hopefully beautiful things*; that's all.
- Even Picasso at his most bizarre was bound by a kind of formal architecture *within the frame*.
- It is at best that magic framework of reality that we sometimes touch upon when we use the word *order*. (FLW). What people want, above all, is *order*.
- Don't tell me what you want, *tell me how you live*. (FLW – I think!)

1. Defined functionality and interfaces
2. The why
3. Scope Boundaries
4. Organization and order
5. Requirements, requirements, requirements

Architecture Definitions



Software Architecture in Practice

Bass, Clements, Kazman; Addison-Wesley

- Realization of **early design decisions** made regarding the decomposition of the system into **parts**
- **Abstract view** that distills away details of implementation, algorithm, and data representation and concentrates on the behavior and interaction of “black box” components.
- Summary result of a set of **business and technical decisions**

Software Architecture in Practice

Bass, Clements, Kazman; Addison-Wesley

“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.”

□ Observations from this definition

1. *architecture defines elements*
2. *systems can and do comprise more than one structure*
3. **every software system has an architecture**
4. *behavior of each element is part of the architecture*



ANSI/IEEE Std 1471-2000, Recommended Practice for Architectural Description of Software-Intensive Systems


*“...the fundamental **organization of a system**, embodied in its components, their relationships to each other and the environment, and the **principles governing its design and evolution.**”*

Kruchten: *The Rational Unified Process*, Booch, Rumbaugh, and Jacobson: *Unified Modeling Language User Guide*



“An architecture is the set of significant **decisions about the organization of a software system**, the selection of the structural elements and their interfaces by which the system is composed, together with their behavior as specified in the collaborations among those elements, the composition of these structural and behavioral elements into progressively larger subsystems, and the architectural style that guides this organization---**these elements and their interfaces, their collaborations, and their composition.**”

Garlan and Shaw.



*“...beyond the algorithms and data structures of the computation; designing and specifying the overall system structure emerges as a new kind of problem. **Structural** issues include gross organization and global **control structure**; protocols for **communication**, synchronization, and data access; **assignment of functionality** to design elements; **physical distribution**; composition of design elements; scaling and performance; and selection among design alternatives.”*

Hayes-Roth

From the DSSA community



“...an abstract system specification consisting primarily of functional components described in terms of their behaviors and interfaces and component-component interconnections.”

Barry Boehm

USC Center for Software Engineering

- A software system architecture comprises
 - ▣ A collection of software and system **components, connections, and constraints**.
 - ▣ A collection of system **stakeholders' need statements**.
 - ▣ A **rationale** which demonstrates that the components, connections, and constraints define a system that, if implemented, would satisfy the collection of system stakeholders' need statements.

Architectural Model Views

Mary Shaw, 1st Int. Wkshp on Architectures for Software Systems, 1995.

- **Structural models** all hold that software architecture is composed of components, connections among those components, plus (usually) some other aspect or aspects, including (grouping suggested by the authors):
 - ▣ configuration, style
 - ▣ constraints, semantics
 - ▣ analyses, properties
 - ▣ rationale, requirements, stakeholders' needs
- **Framework models** are similar to the structural view, but their primary emphasis is on the (usually singular) coherent structure of the whole system, as opposed to concentrating on its composition. Framework models often target specific domains or problem classes. Work that exemplifies the framework view includes domain-specific software architectures, CORBA [55] or CORBA-based architecture models, and domain-specific component repositories (e.g., PRISM).
- **Dynamic models** emphasize the behavioral quality of systems. "Dynamic" may refer to changes in the overall system configuration, setting up or disabling pre-enabled communication or interaction pathways, or the dynamics involved in the progress of the computation, such as changing data values.
- **Process models** focus on construction of the architecture, and the steps or process involved in that construction. In this view, architecture is the result of following a process script. This view is exemplified by work in process programming for deriving architectures.

What is the common thread in these definitions?





What are some common threads for these architectural definitions?

Architecture ROI



Architecture as an asset



- “Architecture is an asset that holds tangible value to the developing organization beyond the project for which it was created” [Bass, et.al.]

The Utility of Software Architectures



- “Architectures serve as an important communication, reasoning, analysis and growth tool for systems (pg. 3).” [Bass et.al.]

What are the business drivers for Product Architecture?

Important slide for final project


- ☐ Time
- ☐ Quality
- ☐ Change
- ☐ Maintenance
- ☐ Reuse
- ☐ Communication

Improve time for product delivery

Architecture Derivation



What Makes a “Good” Architecture?

- 
- “If it is true that, given the same technical requirements for a system, two different architects in different organizations will produce different architectures, how can we determine if either one of them is the right one?” [Bass et.al.,]
 - There is no such thing as an inherently good or bad architecture.
 - Architectures are either more or less fit for some stated purpose.

Recommendations



- Architecture is a product of a single architect or small group with a leader
- Architect (or architect team) should have technical requirements and prioritized qualities the system is expected to satisfy.
- Architecture should be well documented using an agreed-on notation that all stakeholders understand with minimum effort.
- Architecture should be circulated to the stakeholders, who should be actively involved in its review.

Reflection



- How and why do software architectures ...
 - ▣ Reduce time to delivery?
 - ▣ Increase quality?
 - ▣ Help manage change?
 - ▣ Decrease maintenance costs?
 - ▣ Increase reuse?
 - ▣ Improve communication?