Architecture and Business

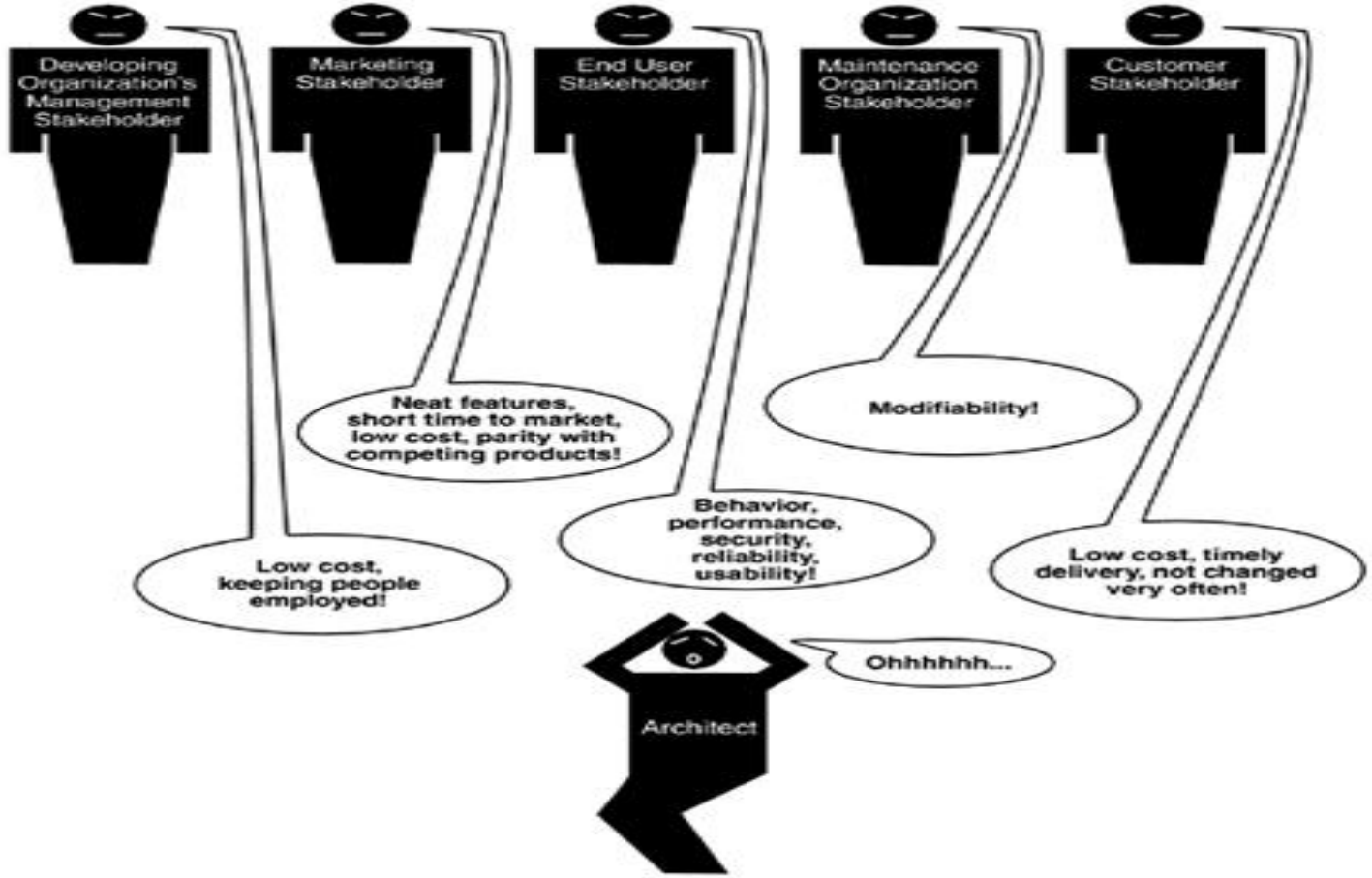
Where Do Architectures Come From?

- An architecture is the result of a set of business and technical decisions. There are many influences at work in its design, and the realization of these influences will change depending on the environment in which the architecture is required to perform.
- An architect designing a system for which the real-time deadlines are believed to be tight will make one set of design choices;
- the same architect, designing a similar system in which the deadlines can be easily satisfied, will make different choices. And the same architect, designing a non-real-time system, is likely to make quite different choices still. Even with the same requirements, hardware, support software, and human resources available, an architect designing a system today is likely to design a different system than might have been designed five years ago.

ARCHITECTURES ARE INFLUENCED BY SYSTEM STAKEHOLDERS

- Many people and organizations are interested in the construction of a software system. We call these stakeholders:
- The customer, the end users, the developers, the project manager, the maintainers, and even those who market the system are a few examples.
- Stakeholders have different concerns that they wish the system to guarantee or optimize, including things as diverse as providing a certain behavior at runtime, performing well on a particular piece of hardware, being easy to customize, achieving short time to market or low cost of development, gainfully employing programmers who have a particular specialty, or providing a broad range of functions.

Cont…


Developing Organization's Management Stakeholder
Marketing Stakeholder
End User Stakeholder
Maintenance Organization Stakeholder
Customer Stakeholder
Low cost, keeping people employed!
Neat features, short time to market, low cost, parity with competing products!
Behavior, performance, security, reliability, usability!
Modifiability!
Low cost, timely delivery, not changed very often!
Architect
Ohhhhhh...

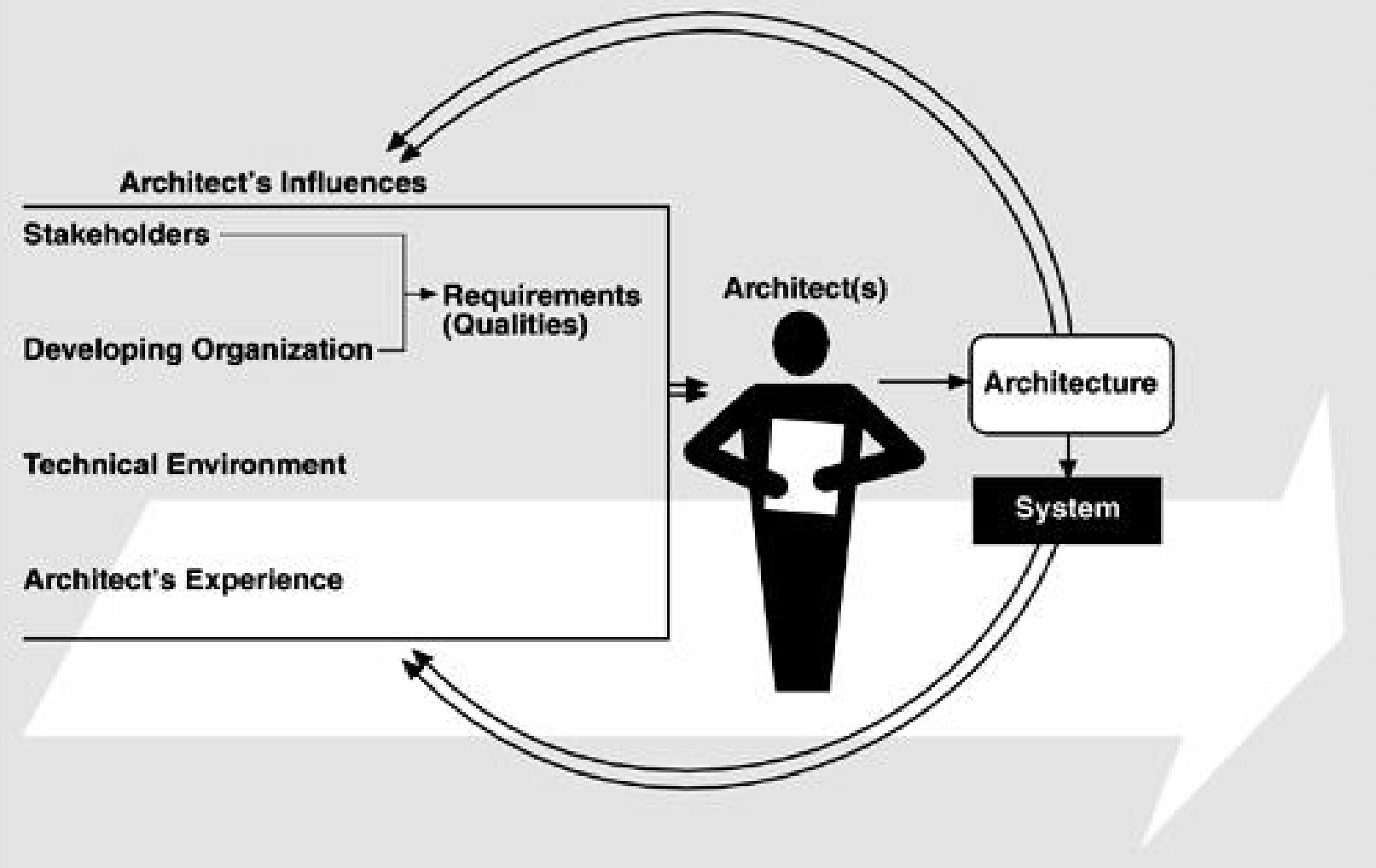
Cont…

- Having an acceptable system involves properties such as performance, reliability, availability, platform compatibility, memory utilization, network usage, security, modifiability, usability, and interoperability with other systems as well as behavior.
- Indeed, we will see that these properties determine the overall design of the architecture. All of them, and others, affect how the delivered system is viewed by its eventual recipients, and so they find a voice in one or more of the system's stakeholders.

Cont…

- Other influences to and from architectures.
  - **Architectures are influenced by the developing organization**
  - **Architectures are influenced by the background and experience of the architects**
  - **Architectures are influenced by the technical environment**
  - **Ramifications of influences on an architecture**
  - **The architectures affect the factors that influence them**

The Architecture Business Cycle



Cont..

Here is how the cycle works:

1. The architecture affects the structure of the developing organization. An architecture prescribes a structure for a system; as we will see, it particularly prescribes the units of software that must be implemented (or otherwise obtained) and integrated to form the system. These units are the basis for the development project's structure. Teams are formed for individual software units; and the development, test, and integration activities all revolve around the units. Likewise, schedules and budgets allocate resources in chunks corresponding to the units. If a company becomes adept at building families of similar systems, it will tend to invest in each team by nurturing each area of expertise. Teams become embedded in the organization's structure. This is feedback from the architecture to the developing organization

Cont…

2. The architecture can affect the goals of the developing organization. A successful system built from it can enable a company to establish a foothold in a particular market area. The architecture can provide opportunities for the efficient production and deployment of similar systems, and the organization may adjust its goals to take advantage of its newfound expertise to plumb the market. This is feedback from the system to the developing organization and the systems it builds.

3. The architecture can affect customer requirements for the next system by giving the customer the opportunity to receive a system (based on the same architecture) in a more reliable, timely, and economical manner than if the subsequent system were to be built from scratch. The customer may be willing to relax some requirements to gain these economies.

Cont..

4. The process of system building will affect the architect's experience with subsequent systems by adding to the corporate experience base. A system that was successfully built around a tool bus or .NET or encapsulated finite-state machines will engender similar systems built the same way in the future. On the other hand, architectures that fail are less likely to be chosen for future projects.

5. A few systems will influence and actually change the software engineering culture, that is, the technical environment in which system builders operate and learn. The first relational databases, compiler generators, and table-driven operating systems had this effect in the 1960s and early 1970s; the first spreadsheets and windowing systems, in the 1980s.

Software Processes and the Architecture Business Cycle

- Software process is the term given to the organization, ritualization, and management of software development activities. What activities are involved in creating a software architecture, using that architecture to realize a design, and then implementing or managing the evolution of a target system or application? These activities include the following:
  - Creating the business case for the system
  - Understanding the requirements
  - Creating or selecting the architecture
  - Documenting and communicating the architecture
  - Analyzing or evaluating the architecture
  - Implementing the system based on the architecture
  - Ensuring that the implementation conforms to the architecture

Cont..

- Creating the business case for the system: Creating a business case is broader than simply assessing the market need for a system. It is an important step in creating and constraining any future requirements. How much should the product cost? What is its targeted market? What is its targeted time to market? Will it need to interface with other systems? Are there system limitations that it must work within?

- **Understanding the Requirements:** There are a variety of techniques for eliciting requirements from the stakeholders. For example, object-oriented analysis uses scenarios, or "use cases" to embody requirements. Safety-critical systems use more rigorous approaches, such as finite-state-machine models or formal specification languages.

Cont…

- **Creating or Selecting the Architecture:** show how to create an architecture to achieve its behavioral and quality requirements.
- **Communicating the Architecture:** For the architecture to be effective as the backbone of the project's design, it must be communicated clearly and unambiguously to all of the stakeholders. Developers must understand the work assignments it requires of them, testers must understand the task structure it imposes on them, management must understand the scheduling implications it suggests, and so forth. Toward this end, the architecture's documentation should be informative, unambiguous, and readable by many people with varied backgrounds.

Cont…

- **Analyzing or Evaluating the Architecture:** Evaluating an architecture for the qualities that it supports is essential to ensuring that the system constructed from that architecture satisfies its stakeholders' needs. Becoming more widespread are analysis techniques to evaluate the quality attributes that an architecture imparts to a system. Scenario-based techniques provide one of the most general and effective approaches for evaluating an architecture.
- **Implementing Based on the Architecture: This** activity is concerned with keeping the developers faithful to the structures and interaction protocols constrained by the architecture. Having an explicit and well-communicated architecture is the first step toward ensuring architectural conformance. Having an environment or infrastructure that actively assists developers in creating and maintaining the architecture (as opposed to just the code) is better.

Cont…

- **Ensuring Conformance to an Architecture: Finally**, when an architecture is created and used, it goes into a maintenance phase. Constant vigilance is required to ensure that the actual architecture and its representation remain faithful to each other during this phase. Although work in this area is comparatively immature, there has been intense activity in recent years.

What Makes a "Good" Architecture?

- There is no such thing as an inherently good or bad architecture. Architectures are either more or less fit for some stated purpose. A distributed three-tier client-server architecture may be just the ticket for a large enterprise's financial management system but completely wrong for an avionics application.
- An architecture carefully crafted to achieve high modifiability does not make sense for a throw-away prototype. One of the messages of this book is that architectures can in fact be evaluated—one of the great benefits of paying attention to them—but only in the context of specific goals.
- Nevertheless, there are rules of thumb that should be followed when designing an architecture. Failure to apply any of these does not automatically mean that the architecture will be fatally flawed, but it should at least serve as a warning sign that should be investigated.

Cont..

- We divide our observations into two clusters: process recommendations and product (or structural) recommendations. Our process recommendations are as follows:
  - The architecture should be the product of a single architect or a small group of architects with an identified leader.
  - The architect (or architecture team) should have the functional requirements for the system and an articulated, prioritized list of quality attributes (such as security or modifiability) that the architecture is expected to satisfy.
  - The architecture should be well documented, with at least one static view and one dynamic view .
  - The architecture should be circulated to the system's stakeholders, who should be actively involved in its review.

Cont…

- The architecture should be analyzed for applicable quantitative measures (such as maximum throughput) and formally evaluated for quality attributes before it is too late to make changes to it.
- The architecture should lend itself to incremental implementation via the creation of a "skeletal" system in which the communication paths are exercised but which at first has minimal functionality. This skeletal system can then be used to "grow" the system incrementally,
- The architecture should result in a specific (and small) set of resource contention areas, the resolution of which is clearly specified, circulated, and maintained. For example, if network utilization is an area of concern, the architect should produce (and enforce) for each development team guidelines that will result in a minimum of network traffic. If performance is a concern, the architects should produce (and enforce) time budgets for the major threads.

Cont…

- Our structural rules of thumb are as follows:
- The architecture should feature well-defined modules whose functional responsibilities are allocated on the principles of information hiding and separation of concerns. The information-hiding modules should include those that encapsulate idiosyncrasies of the computing infrastructure, thus insulating the bulk of the software from change should the infrastructure change.
- Each module should have a well-defined interface that encapsulates or "hides" changeable aspects (such as implementation strategies and data structure choices) from other software that uses its facilities. These interfaces should allow their respective development teams to work largely independently of each other.
- Quality attributes should be achieved using well-known architectural tactics specific to each attribute.
- The architecture should never depend on a particular version of a commercial product or tool. If it depends upon a particular commercial product, it should be structured such that changing to a different product is straightforward and inexpensive.

Cont..

- Modules that produce data should be separate from modules that consume data. This tends to increase modifiability because changes are often confined to either the production or the consumption side of data. If new data is added, both sides will have to change, but the separation allows for a staged (incremental) upgrade.
- For parallel-processing systems, the architecture should feature well-defined processes or tasks that do not necessarily mirror the module decomposition structure. That is, processes may thread through more than one module; a module may include procedures that are invoked as part of more than one .
- Every task or process should be written so that its assignment to a specific processor can be easily changed, perhaps even at runtime.
- The architecture should feature a small number of simple interaction patterns
- That is, the system should do the same things in the same way throughout. This will aid in understandability, reduce development time, increase reliability, and enhance modifiability. It will also show conceptual integrity in the architecture, which, while not measurable, leads to smooth development.

Thanks