A good system architecture exhibits conceptual integrity; that is, it comes equipped with a set of design rules that aid in reducing complexity and that can be used as guidance in detailed design and in system verification. Design rules may incorporate certain abstractions that are always used in the same way, such as virtual devices. The rules may be represented as a pattern, such as pipes and filters. In the best case there are verifiable rules, such as "any virtual device of the same type may replace any other virtual device of the same type in the event of device failure," or "all processes contending for the same resource must have the same scheduling priority."

A contemporary architect might say that the object or system under construction must have the following characteristics.

- It has the functionality required by the customer.
- It is safely buildable on the required schedule.
- It performs adequately.
- It is reliable.
- It is usable and safe to use.
- It is secure.
- It is affordable.
- It conforms to legal standards.
- It will outlast its predecessors and its competitors.

The architecture of a computer system we define as *the minimal set of properties that determine what programs will run and what results they will produce*.

—*Gerrit Blaauw & Frederick Brooks*, Computer Architecture

We've never seen a complex system that perfectly satisfies all of the preceding characteristics. Architecture is a game of trade-offs—a decision that improves one of these characteristics often diminishes another. The architect must determine what is sufficient to satisfy, by discovering the important concerns for a particular system and the conditions for satisfying them sufficiently.

Common among the notions of architecture is the idea of structures, each defined by components of various sorts and their relations: how they fit together, invoke each other, communicate, synchronize, and otherwise interact. Components could be support beams or internal rooms in a building, individual instruments or melodies in a symphony, book chapters or characters in a story, CPUs and memory chips in a computer, layers in a communications stack or processors connected to a network, cooperating sequential processes, objects, collections of compile-time macros, or build-time scripts. Each discipline has its own sets of components and its own relationships among them.