Summary of Structures

Table 1-1 summarizes the preceding software structures, how they are defined, and the concerns that they satisfy.

TABLE 1-1. Structure summary

Structure	Components	Relations	Concerns
Information Hiding	Information Hiding Modules	Is a part of Is contained in	Changeability Modularity Buildability
Uses	Programs	Uses	Producibility Ecosystem
Process	Processes (tasks, threads)	Gives work to Gets resources from Shares resources with Contained in ...	Performance Changeability Capacity
Data Access	Programs and Segments	Has access to	Security Ecosystem

Good Architectures

Recall that architects play a game of trade-offs. For a given set of functional and quality requirements, there is no single correct architecture and no single "right answer." We know from experience that we should evaluate an architecture to determine whether it will meet its requirements before spending money to build, test, and deploy the system. Evaluation attempts to answer one or more of the concerns discussed in previous sections, or concerns specific to a particular system.

There are two common approaches to architecture evaluation (Clements, Kazman, and Klein 2002). The first class of evaluation methods determines properties of the architecture, often by modeling or simulation of one or more aspects of the system. For example, performance modeling is carried out to assess throughput and scalability, and fault tree models can be used to estimate reliability and availability. Other types of models include using complexity and coupling metrics to assess changeability and maintainability.

The second, and broadest, class of evaluation methods is based on questioning the architects to assess the architecture. There are many structured questioning methods. For example, the