the system. Stakeholders have certain concerns that the architect must address. Later, we will discuss concerns that are typically raised when trying to assure that the system has the required qualities. As we said earlier, one role of the architect is to ensure that the design of the system will meet the needs of the client, and we use quality concerns to help us understand those needs.

This example highlights two key practices of successful architects: stakeholder involvement and a focus on *both* quality concerns and functionality. As the architect, you began by asking us what we wanted from the system, and in what priority. In a real project, you would have sought out other stakeholders. Typical stakeholders and their concerns include:

- Funders, who want to know if the project can be completed within resource and schedule constraints
- Architects, developers, and testers, who are first concerned with initial construction and later with maintenance and evolution
- Project managers, who need to organize teams and plan iterations
- Marketers, who may want to use quality concerns to differentiate the system from competitors
- Users, including end users, system administrators, and the people who do installation, deployment, provisioning, and configuration
- Technical support staff, who are concerned with the number and complexity of Help Desk calls

Every system has its own set of quality concerns. Some, such as performance, security, and scalability, may be well-specified, but other, often equally important concerns, such as changeability, maintainability, and usability, may not be defined with enough detail to be useful. Odd, isn't it, that stakeholders want to put functions in software and not hardware so that they can be easily and quickly modified, and then often give short shrift to changeability when stating their quality concerns? Architecture decisions will have an impact on what kinds of changes can be done easily and quickly and what changes will take time and be hard to do. So shouldn't an architect understand his stakeholders' expectations for qualities such as "changeability" as well as he understands the functional requirements?

Once the architect understands the stakeholders' quality concerns, what does she do next? Consider the trade-offs. For example, encrypting messages improves security but hurts performance. Using configuration files may increase changeability but could decrease usability unless we can verify that the configuration is valid. Should we use a standard representation for these files, such as XML, or invent our own? Creating the architecture for a system involves making many such difficult trade-offs.

The first task of the architect, then, is to work with stakeholders to understand and prioritize quality concerns and constraints. Why not start with functional requirements? Because there are usually many possible system decompositions. For example, starting with a data model