Part 3
Rationale and Software Engineering

The importance of rationale in software engineering is underscored by rationale being featured as a key activity in recent talks on the Future of Software Engineering (Taylor and van der Hoek 2007;Whitehead 2007) and by rationale being featured as part of one of the process areas in the Software Engineering Institute's Capability Maturity Model Integration: Decision Analysis and Resolution (CMMI Team 2006).

Decisions are made throughout the software development process ranging from deciding how customer requests can be translated into software requirements to deciding when and how to adapt software in operation and on to when a system is ready for retirement (Chapters 10–14). The rationale behind those decisions documents the developers' intent and keeps this information from being lost forever due to attrition, reassignment, or by simply being forgotten.

An important aspect to software development that cross-cuts development phases is reuse (Chapter 15). As software increases in complexity and cost, it becomes critical to avoid "reinventing the wheel" and to utilize existing software applications to save time, by buying instead of building, to save money, since the price to purchase an off-the-shelf application is often less than building it yourself, and to increase reliability, by working with applications that have already received extensive evaluation. Still, while reuse can potentially meet these valuable goals, it is not without its dangers. Deciding when and how reuse should be utilized, and what the best candidates for reuse are, must be carefully deliberated.

The ability of a software system to fulfill the needs of its stakeholders is directly dependent on the degree to which those needs were taken into account by the developer. By capturing the decisions made during development and relating the alternatives chosen to the stakeholder needs, it is possible to use this Software Engineering Rationale to assess the ability of the software to meet those needs. This is the essence of Rationale-Based Software Engineering.