Nonfunctional Requirement	Definition
Maintainability	Amount of effort required to maintain (and enhance) application in production
Multiple environment support	Need to run multiple environments on a single server (development, system test, user test, and so on)—this could mean that the application components can never "hold and lock" vital operating system ports or resources, like a CD-ROM drive or port 8080 to the Internet
Operability	Ease of everyday operation; amount of qualification and training required for operators to oversee and troubleshoot the application
Performance	Constraints in batch (overnight window) and online
Personalization	Ability for individual users to personalize their view of the application (My Yahoo! Style)
Portability	Ability to easily move the application to different hardware platforms, operating systems, database management systems, network protocols, etc.
Privacy	Ability to hide transactions from internal company employees (transactions encrypted so even database administrators [DBAs] and network architects cannot see them)
Reliability	Confidence in the accuracy of transactions processed in the application
Robustness	Ability to handle error and boundary conditions while running (internet connection goes down, power outage, hardware failure/replacement, and so on)
Scalability	Ability to handle a wide variety of system configuration sizes and requirements
Security	General security requirements (encryption levels over the Internet, hackerproofing, viruses, and so on)
Upgradeability	Ability to easily/quickly upgrade from a previous version of this application to a newer version on servers and clients (upgrade scripts versus manual upgrades)
Usability/ achievability	Level of training required for users to achieve their goals with the application. Usability requirements need to be treated as seriously as any other architectural issue (performance, availability, and so on). Many applications fail because they are hard to use and nonintuitive.