

## Use Case

The Use Case tool is key to defining functional requirements, but not for capturing the non-functional requirements. Discuss the statement and show how this situation can be resolved.

Since the Use Case tool is for defining specific inputs, outputs, and behavior of an information system, it would be a misuse of the tool to expand it to non-functional requirements. A use case is something very specific, an action requiring a response. These are scenarios users describe about their jobs, called “user stories” in the extreme programming methodology (Extreme Programming, 1999), which the system analyst converts into transactions for the system to perform. The implementation of functional requirements may be tested by having testers run through the many use cases.

A non-functional requirement is something needing quantification. For instance, system performance and up-time are both non-functional requirements. “The system must be up 99.9 percent of the time,” is a non-functional requirement. Such requirements require a very different approach for documentation and testing. For instance, the performance of a system would need to be measured with comprehensive software that can run benchmarks and load testing. There appears to be a need in Software Engineering for a more effective way to document and measure non-functional requirements, which products like Eclipse TPTP, Debian Linux Development Server, TestNG, and Junit are attempting to fill (Eide, 2008).

Extreme Programming, *User Stories*, 1999, Retrieved on Feb 2, 2008 from:  
<http://www.extremeprogramming.org/rules/userstories.html>

Petter L. H. Eide, *Quantification and Traceability of Requirements*, Norwegian University of Science and Technology, Fall 2005. Retrieved online Feb 2, 2008 from:  
<http://www.idi.ntnu.no/grupper/su/fordypningsprosjekt-2005/eide-fordyp05.pdf>