

**Creep**—changes to requirements and project scope—is a main cause of project budget and schedule overruns. Conventional wisdom considers creep to be caused mainly by unclear **system and/or software requirements** *of the system/product to be created*. Since system/software requirements often describe the externally-observable functions that the system/product must perform in order to work as intended, they often also are called “Functional Requirements” or “Functional Specifications.”

However, there’s another type of requirements, **business requirements**, that often are understood inadequately, largely because developers and others tend to think that system/software requirements are *the* requirements. **Business requirements are what must be delivered to provide value**. Value comes from meeting business objectives. Business requirements are the REAL requirements, since value will be received only when the business requirements are met. There usually are *many possible ways* to accomplish the business requirements.

*One of those possible ways* of **how** presumably to accomplish the business requirements is the particular system/product to be created. A system/product only provides value if it actually satisfies the REAL, business requirements. A seldom-recognized but **major cause of creep is the failure of systems/products to meet the REAL, business requirements**, even when their system/software requirements are defined clearly and have been met.

Such failures usually are due to overlooking or inadequately defining the REAL, business requirements—typically because people are not sufficiently aware of the need to define business requirements and because they seldom know how to do it well. A contributing culprit is the common but mistaken conventional wisdom that business requirements are high-level and vague but decompose into detailed system/software requirements.

In fact, the difference between business and system/software requirements is not merely a level of detail. Rather, they are qualitatively different. Business requirements are **what** must be delivered to provide value, whereas system/software requirements are really high-level design of **how** the business requirements can be delivered. Both business and system/software requirements should be defined at both high- and detailed-levels.

The more one focuses on the system/software requirements, the less likely one is to understand adequately or even to be aware of the REAL, business requirements; and creep is inevitable. Conversely, *first defining business requirements appropriately in suitable detail reduces creep*, because then a system/product and its system/software requirements can be defined to actually meet the REAL, business requirements.

We are the first to acknowledge that often it is difficult to identify the REAL, business requirements. Not only is it much more inherently difficult than most people presume, the difficulty is aggravated when one has operated on the mistaken presumption that whatever system/product they intend to build must be what the user requires.

## **REAL, Business Requirements vs. Software Requirements**

2 of 2

To help overcome these difficulties, Go Pro Management, Inc. provides consulting and training assistance, including:

- Our two-day **Defining and Managing User Requirements** and our one-day version **Discovering Business Requirements** seminars teach how to use the powerful **Problem Pyramid™** tool and other techniques to discover and document the REAL, business requirements.
- Our one-day **21 Ways to Test that Business Requirements Are Right** (also presented with the title **Evaluating Business Requirements**) seminar shows a variety of methods which are much more powerful than conventional techniques to review accuracy and completeness of requirements that already have been defined. This content also is presented as the first day of our two-day **Testing Early in the Life Cycle** (also presented with the title **Testing that Requirements and Designs Are Right**) seminar, where the second day presents 15 ways to test that designs are right.
- Robin Goldsmith's recent Artech House book, **Discovering REAL Business Requirements for Software Project Success**, combines and elaborates upon the above seminars' concepts and techniques for both discovering and testing the adequacy of business requirements.
- Our one-day **Proactive User Acceptance Testing** seminar shows how to plan and carry out tests to demonstrate that the delivered system in fact meets the business requirements. Proactive planning of these tests not only increases user competence and confidence, it also helps detect missing, unclear, and incorrect requirements.