

# Foreword

How do you decide what a software system is supposed to do? And then, how do you communicate that decision between the various people affected? This book takes on this complicated problem. The problem is difficult because each participant has different needs. Project managers want to track progress. Programmers want to implement the system. Product managers want flexibility. Testers want to measure. Users want a useful system. Creating productive conflict between these perspectives, coming to a single collective decision everyone can support, and maintaining that balance for months or years are all difficult problems.

The solution Mike Cohn explores in this book, *User Stories Applied*, is superficially the same as previous attempts to solve this problem—requirements, use cases, and scenarios. What’s so complicated? You write down what you want to do and then you do it. The proliferation of solutions suggests that the problem is not as simple as it appears. The variation comes down to what you write down and when.

User stories start the process by writing down just two pieces of information: each goal to be satisfied by the system and the rough cost of satisfying that goal. This takes just a few sentences and gives you information other approaches don't. Using the principle of the “last responsible moment,” the team defers writing down most details of the features until just before implementation.

This simple time shift has two major effects. First, the team can easily begin implementing the “juiciest” features early in the development cycle while the other features are still vague. The automated tests specifying the details of each feature ensure that early features continue to run as specified as you add new features. Second, putting a price on features early encourages prioritizing from the beginning instead of a panicked abortion of scope at the end in order to meet a delivery date.

