

Requirements Engineering in the Solution Domain



Never tell people how to do things. Tell them what to do, and they will surprise you with their ingenuity. (George Smith Patton, general, 1885–1945)

6.1 What is the Solution Domain?

The solution domain is the domain in which engineers use their ingenuity to solve problems. The primary characteristic that differentiates the solution domain from the problem domain is that, invariably, requirements engineering in the solution domain starts with a given set of requirements. In the problem domain requirements engineering starts with a vague objective or wish list. The extent to which the input requirements for the solution domain are “well formed” depends upon the quality of the people within the customer organization that developed them. In an ideal world, all the requirements would be clearly articulated, individually testable requirements.

As indicated in Chapter 2, the solution is very rarely arrived at in a single step (see Figure 6.1).

At each level there is modelling and analysis done to first understand the input requirements and secondly to provide a sound basis for deriving the requirements for the next level down. The number of levels of design is dictated by the nature of the application domain and the degree of innovation involved in the development. No matter how many levels are necessary it is always vital to understand how much solution detail – the “how” – should be introduced at each step.

At every level in the solution domain, engineers must make decisions that move towards the final solution. Each of these decisions, by their very nature reduces the available design space; that is, they preclude certain design options, but it is impossible to make progress in the absence of decisions. Engineers are always very strongly tempted to go into too much detail too

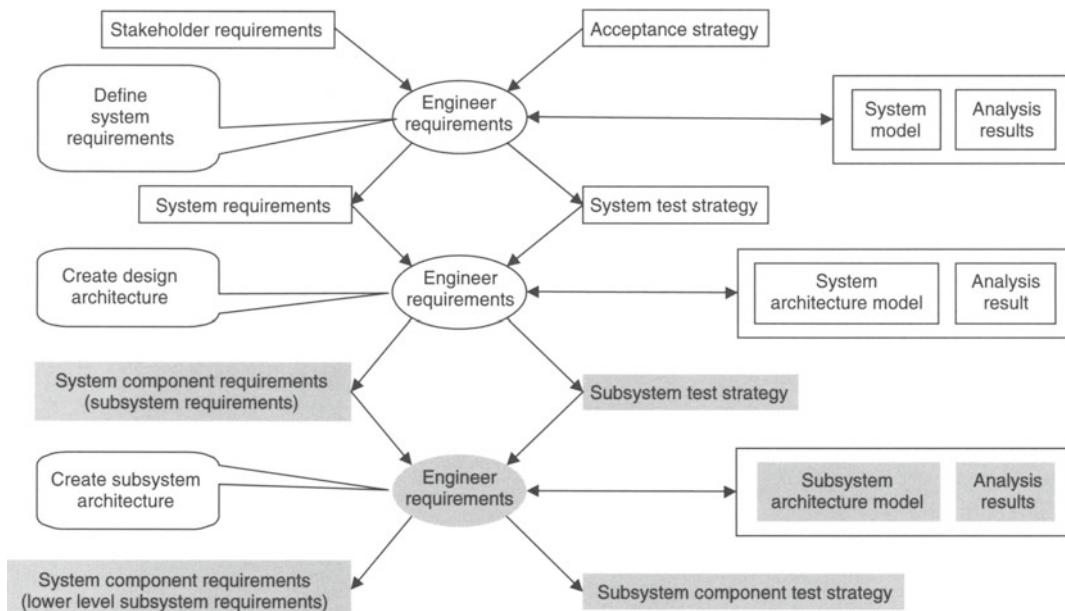


Figure 6.1 Possible instantiations of the generic process

soon. This temptation must be avoided, in order to allow creativity and ingenuity to work together to produce innovative solutions that could never be achieved in the presence of the constraints imposed by premature design decisions.

Typically the first level of system development in the solution domain is to transform the stakeholder requirements into a set of system requirements. These must define what the system must do in order to solve the problems posed by the stakeholder requirements. This first level is illustrated by the top instantiation of the generic process in Figure 6.1.

The issue of premature design detail is especially problematic at the first step. The system model indicated in Figure 6.1 must be created at a level of abstraction that enables the functionality of the system to be defined without going into unnecessary detail.

The next step after defining the system requirements is to create an architectural design as indicated by the second instantiation of the generic process in Figure 6.1. This must be expressed in terms of a set of components that interact to generate the emergent properties identified by the system requirements. The derived requirements from the architectural design process (Figure 6.1) define the requirements that the component suppliers must satisfy for each component.