Functional Analysis – An Overview

By Robert J. Halligan, FIE Aust
Project Performance International

Functional analysis is generally regarded as an activity whereby functions are broken down into sub-functions, and relationships between sub-functions (e.g. sequence, concurrency, control flow, item flow, logical branching, looping, iteration, replication, etc.). Therefore, the breakdown (decomposition) of a function describes how that function is to be accomplished (performed).

Functional analysis has two major applications. The first is as a tool for CAPTURE and VALIDATION of requirements. That is, functional analysis is a technique used WITHIN requirements analysis. Anybody who has ever developed a use case, or written down how something is to be used for a particular use, has used functional analysis in this application. There are, of course, much more robust ways of using functional analysis as a requirements analysis tool than the traditional IT use case approach.

The second major application of functional analysis is as a design tool, a LOGICAL DESIGN tool. In this application, requirements level functions are broken down into solution level functions. Take the "Conduct the Olympic Games (a requirements level function)" system. When we define functions such as "design the stadium", "build the stadium", "obtain certificate of occupancy", "conduct event trial", etc., we are DECIDING UPON, we are CREATING, solution level functions. It is exactly the same with technology-based solutions. As soon as we decide that the function "design the stadium" is to be performed by the "stadium contractor" object, we have CREATED a functional requirement on that object. That is the design application of functional analysis.

When the "stadium contractor" object receives our requirements, and finds them unclear, incomplete, unverifiable, all the usual problems, the stadium contractor may perform a requirements analysis to capture missing requirements, validate what is already there, identify requirements issues to raise with us, raise those requirements issues with us, and facilitate their resolution. So in this sense, the design application of functional analysis was first, in creating some functional requirements on a solution object, which became the subject of the requirements analysis application of functional analysis, which lead to validated requirements on a solution level object, which became the subject of the design application of functional analysis. The relationship is recursive – the proverbial "which came first, the chicken or the egg?".

As a point of detail, the design application of functional analysis creates a design architecture, a LOGICAL design architecture. That logic is the logic of a PHYSICAL (STRUCTURAL) design architecture. Both are forms of design (noun); the activities that create them are both forms of design (verb). But having architecture is not enough; having selected architecture, we then need to add detail to a level of implementability, giving us a detailed design (noun).