

Software Architecture: Leverage for System/Program Comprehension

Dewayne Perry
University of Texas, USA

Abstract

Building software systems is hard; evolving them is harder. Even with complete requirements, architecture and design documentation, comprehending a system, or even a part of a system, in such a way as to understand the implications of the changes we make is a difficult task. I believe that the separation of architecture from design helps us with the problem of system comprehension: identifying the fundamental system components and their interactions (and the constraints on those components and interactions) is the first step towards the creation of what Brooks calls “the conceptual integrity of design.”

Since the problem of component comprehension depends on the context in which the components are used, it is necessary to address this issue of system comprehension to provide that context. To this end, I will look at the role that architecture plays in comprehending and understanding software systems and explore different approaches to the rediscovery of a system’s architecture and how that architecture can be used as a tool for system and component comprehension.