Definition of Software Architecture

Tao Xie, Professor
Computer Science @ Illinois
Learning Objectives

By the end of this video, you will be able to

• Explain what is the role of software architecture.
• Explain example definitions of software architecture.
• Explain key points emphasized when defining software architecture.
• Explain differences between prescriptive and descriptive architectures.
• List example benefits of software architecture.
Role of Software Design

Requirements

Implementation
Simple vs. Complicated Design

• “I conclude that there are two ways of constructing a software design: One way is to make it so simple that there are *obviously* no deficiencies and the other way is to make it so complicated that there are no *obvious* deficiencies.”

—Tony Hoare, 1980 Turing Award Winner (Hoare, 1981)
Definitions of Software Architecture

• “A software system’s architecture is the set of principal design decisions about the system.” (Taylor et al., 2009)

• “The software architecture of a program or computing system is the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them.” (Bass et al., 2003)

• A model of software architecture “consists of three components: elements, form, and rationale.” (Perry & Alexander, 1992)
  • Elements (what): Processing, data, or connecting elements
  • Form (how): Constraints (properties, relationships) on the elements
  • Rationale (why): System constraints, often derived from system requirements
Prescriptive vs. Descriptive Architecture

(Taylor et al., 2009)
Benefits of Software Architecture

(Garlan, 2000)
References


The End