

1. *What are the important similarities and differences between an "architectural style" and a "programming paradigm" (like functional programming, logic programming, dataflow, etc.)?*

I think the similarity between an architectural style and a programming paradigm is that they both classify problems based on their features and aim at structuring solutions to them, where the difference is the level of where that problem and solution exist. So, an architectural style will show the highest level of abstraction of the system design where it focuses on layers and modules and how they are communicating together. However, a programming paradigm focus on data structure and algorithm of the computation in which it shows how a given program or programming language can be organized. That is, an architectural style takes care of the outer part of the system while programming paradigm takes care of the inside part of it. I also think constraints are another similarity between the two where each architectural style and programming paradigm has its own constraints to be considered while applying it.

2. *How do you think "architectural style" affects software quality?*

I believe architectural style improves software quality because of the different software quality attributes it positively affects like reusability, interoperability, and maintainability. So having an architectural style that defines the module of a software makes it easier to reuse this module in another software as well as coupling one software into another. Also, the experience developers get while applying such style will be transferred as well when reusing it in another system. I also believe that having an architectural style makes it easier to understand the whole system which accordingly improves maintainability by reducing the effort required to locate and fix an error.