**Discuss the 4+1 model of software architecture**

According to Sommerville (2011), models of software architecture are helpful in understanding and comparing software, ensuring the validity of software system designs, and assessing the software architecture for reuse. Moreover the models can facilitate communication between system stakeholders as well as document the system architecture. Often software architecture is usually modelled using block diagrams, which show various sub-systems that make up a software system and the various relationships between them. One problem with these models is that the types of relationships between entities are not usually indicated. Another problem is that these models do not explain the properties of the entities themselves (Sommerville, 2011).

In order to alleviate the inherent problems of block diagrams of software architecture and to show more views or perspectives of the system, Philippe Krucheten invented the 4+1 model of software architecture, which is described in his original article published in 1995.

Krucheten (1995) proposes the use of 4 main views: the physical view, the logical view, the process view, and the development view as described here:

**The Physical Architecture.** This view identifies the various hardware elements of the system and takes the non-functional requirements of the system into account. Things like the scalability, throughput, availability, and reliability of the system are accounted for in this view.

**The Logical Architecture.** The terms of services provided to the user is described in this view. As such it models the functional requirements of the system. To do this, the logical view decomposes the system into objects of object classes and their logical relationships.

**The Process Architecture.** Using several abstract layers, this view shows how various sub-systems interact at run-time. This view takes into account some of the non-functional system requirements, like availability and performance.

**The Development Architecture.** This view partitions the system into layers or modules with their development in mind. That is, this view will parse the system into libraries or sub-systems that can be developed independently and later integrated into a whole system.

**The +1 view - a misnomer**

In addition to the 4 basic views described above, Krucheten (1995) also proposes **use cases** or **scenarios** as a 5th view that integrates the other 4 basic views. It is for this reason that Krucheten describes this model as a 4+1 view. Although this view may be redundant with part of the other views, the use cases or scenarios view describes the interactions between both objects and processes.

**References**

Kruchten, P. (1995). “Architectural blueprints--The “4+1” view model of software architecture.” IEEE Software 12(6), 42-50.

Sommerville, I. (2011). “Chapter 6 - Architectural Design. Lecture 1-2.” Software Engineering, Ed. 9. Retrieved from <http://my.uopeople.edu/mod/resource/view.php?id=57344>