

Discover QCon Plus by InfoQ: A Virtual Conference for Senior Software Engineers and Architects (Nov 4-18).

Uncle Bob: Architecture is About Intent, not Frameworks

Lire ce contenu en [français](#)

Architecture is about intent, we have made it about frameworks and details, [Robert C. Martin](#), "Uncle Bob", [stated](#) earlier at this year's [DDD Exchange Day](#) in London.

Robert refers to an architecture model described in the book [Growing Object-Oriented Software Guided by Tests](#), (similar to the [Hexagonal architecture](#)), which describes an architecture with three zones with dependencies going only one way, from the volatile parts towards the more stable parts:

- A Domain Model with the core business rules, the most stable and important part for the business, not depending on anything else.
- Application Services for the use cases of the system which uses, and depends on, the domain model.
- External Details, database, user interface, network, etc., that are less relevant for the business model. The most volatile part and depending on the other two.

Robert notes that this model fails to describe what he regards as a key aspect; architecture is about intent, what the application does. He thinks we focus too much on details and frameworks and have made them the centre of our systems.

To deal with this shortage Robert goes back to a book by [Ivar Jacobson](#), "[Object-Oriented Software Engineering, A Use Case Driven Approach](#)" from 1992 where Ivar is defining a mechanism for application architecture based on small use cases without details.

Ivar introduces three types of objects, that fits naturally in the architecture model, and described by Robert as:

- Interactors that understands the use cases and has application specific business rules.
- Entities with the business rules, used by the Interactors.
- Boundary objects that transfer data between the external world and the Interactors.

A significant advantage with this model, Robert claims, is that it's a very testable model; it can be tested without any dependency on infrastructure by just sending and receiving data structures through the boundary objects.

Robert then moves to his Clean Architecture model, a variant of the already mentioned architectures. An essential aspect of all the three models is that they follow the Stable Dependencies Principle; "Do not depend on things that change or are likely to change". The models do this by letting the external volatile parts depend on the more stable parts e.g. the domain model, and not the other way around. This also makes it easier to change an implementation; the volatile part is depending on something that is stable, quoting Robert:

| *A good architecture allows volatile decisions to be easily changed*

Robert also refers to Jim Coplien and the DCI architecture which he thinks is similar to the other architectures.

Some criticism has been raised against some of these ideas that Robert has expressed in recent years, criticism that Robert also have responded to.

1

Please see <https://www.infoq.com> for the latest version of this information.