

Architecture, What is it and why do we need it?

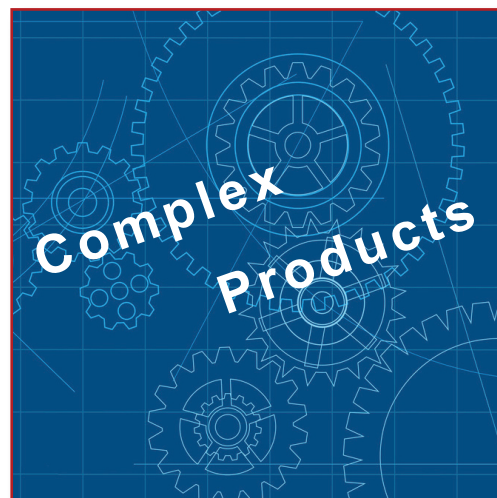
Architecture has been around for a long time. Ancient cultures used architecture, we use architecture and our future relatives will still use architecture. We can see the result of applying architecture all around us in buildings, city structure, landscape development, mechanical systems, IT implementations, and so on. What we see is the architecture the Architect used to design something so it could be engineered, build and fulfill its purpose.

I would like to quote John A. Zachman for a definition of architecture:

“Architecture is the set of descriptive representations relevant for describing a complex object (actually any object) such that an instance of the object can be created and such that the descriptive representations serve as the baseline for changing an object instance.

Assuming the descriptive representations are maintained consistent with the real world.”

The use of architecture goes beyond engineering and creation. The Total Cost of Ownership (TCO) of a complex product or system can be divided into two time periods: creating it and using it. If you realize that using a complex product or system requires adjusting it to the changing needs of the market, you'll see the value of architecture as defined by John A. Zachman, over the life cycle of any complex product or system.



Complex Products

If you need to change a complex product or system, or better *when* you need to change it, and you do not have a descriptive representation relevant for your object, you only have three choices:

1. **Take a risk** and change it anyway.
2. **Reverse engineer** before you change.
3. Don't change, **start all over** again.

Take a risk



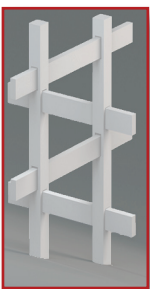
Although you don't know the architecture, you simply change your product or system anyway and see what will happen. This approach has proven to have a high risk that problems will arise. The more complex your instance is, the higher the risk. Don't forget the instance might interact with other objects that might be effected by the attempted change.

Reverse engineer



You start recreating the descriptive representations ('reverse engineer') from the existing implementation. You use the physical instance to learn its principles before you start changing it. This will take time and will cost money but it will ensure your change will be effective and without unexpected results.

Start all over



Don't change the instance to avoid problems. When a change is needed, you start building a new instance all over again. You might even create a descriptive representation of the new instance for future reference. This approach is certainly expensive and in most situations unrealistic.

Architecture helps you to engineer something. Can you imagine building a skyscraper without the descriptive representations explaining the components needed and their relationships? You need structural drawings, electrical schemas, a floor plan, etc. etc.. The local fire department will probably demand to see and approve them before you get a building permit. It's the same with a modern car, a bridge over a wide river, or any other complex object.

Architecture is not arbitrary and not negotiable if you want to build and (safely) use complex products.

The most complex

In my personal opinion, an Enterprise is the most complex object man ever created. And the demand for Enterprise Architecture is rising. Not to create your enterprise, it's already there. We need EA to be able to change your enterprise safely when needed. We change our IT systems to meet the ever changing demand of the customer. We try to meet the expectation of immediate availability of new services and products. To do this, we often use the option 'Take a Risk' (see above). Perhaps IT systems are capable of meeting the new requirements, but due to increasing integration and complexity it is more difficult to successfully manage the change. The integration of (IT) systems causes unexpected effects on services and products. This forces us to solve problems and spend time and money on a change that has already been executed. Time and money we want to invest in the change needed tomorrow.

What we need is a way to know in advance what the effect of our change will be, only then we can deploy our time and money in areas where we need it tomorrow. We need a predictable environment which we can

change quickly. Preferably 'first time right'. This is where EA is going to help. If we reverse engineer the parts of the enterprise we need to change, we can predict the effect of the change. We can even investigate all possible different solutions instead of the three of four we can think of without the insight knowledge that EA offers.

It's all there

We need to change our enterprise almost on a daily basis. We need to make our enterprises adaptable to the customer. If we don't, someone else will set up a new business offering exactly that service (a realistic instance of the third option above). Don't think this is impossible, because anyone can use the same IT enablers. It's more and more available in cloud services and if you do not have a large complex legacy (both IT and enterprise) you can easily meet the new requirements.

Established enterprises do have an advantage by means of business experience. It is all there in the legacy systems. All we need to do is find it, make it available and use it. EA enables you to do this. And if you start all over again with a new business setup, you're still not out of danger. How long does it take to build 'a new legacy', which puts you right where current enterprises are today?

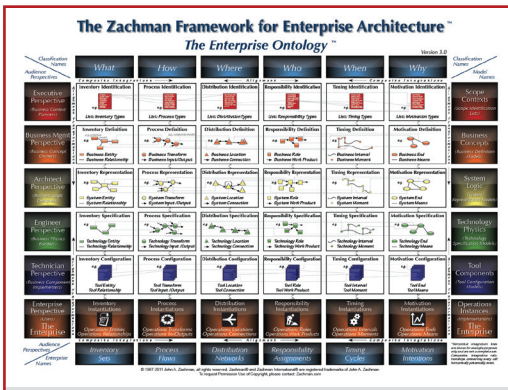
Again: Architecture is not arbitrary and not negotiable if you want to build and (safely) use complex products.

Jef Bergsma

Enterprise Integration Specialist

j.bergsma@jzz-3d.nl

Architecture - What and Why



JZZ-3d has a broad experience with developing business solutions and IT. We have an unparalleled understanding of the importance of IT within an enterprise. We're specialised in applying the Zachman 3.0 framework to solve current integration problems and prepare enterprises for the demanding future.

www.jzz-3d.nl