Twitter Thread by Vladik Khononov

Vladik Khononov
@vladikk

What do you see in the picture? A piece of cardboard? Some junk? No! — It's a model!

Thread on models and bounded contexts 1/9

#DDDesign #BoundedContext

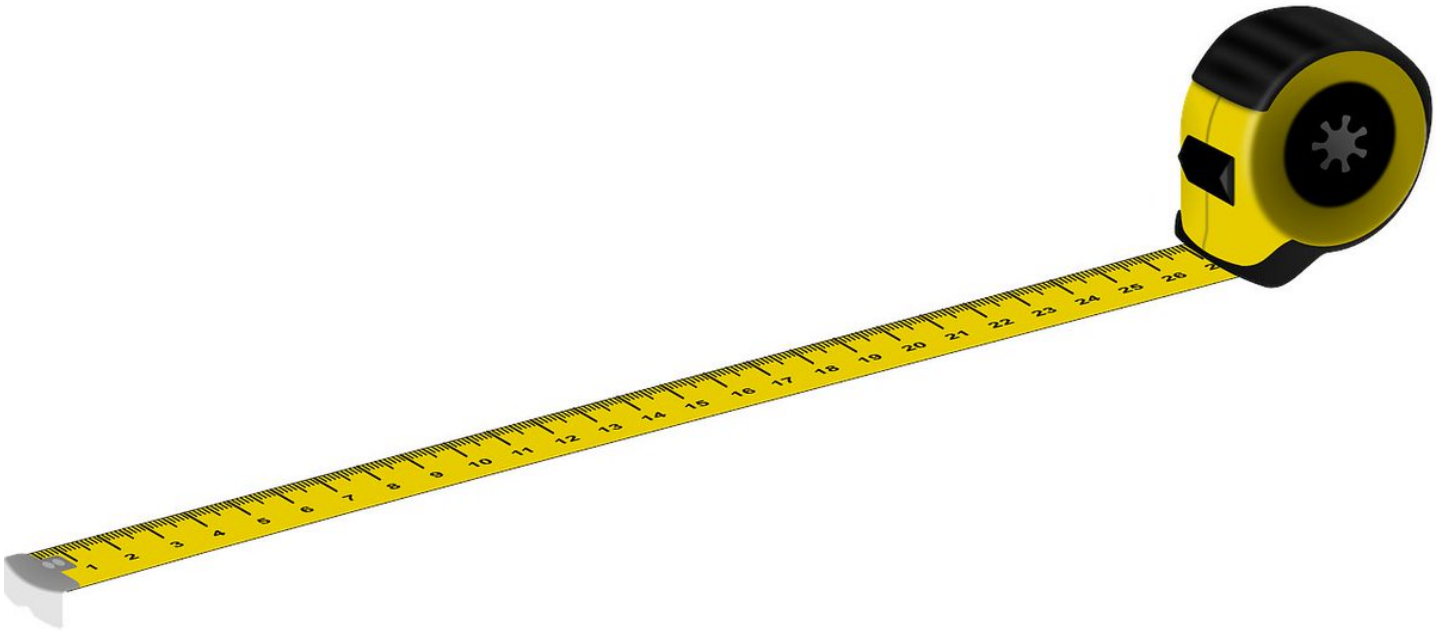
It’s a model of the Siemens KG86NAI31L fridge. The cardboard doesn’t look anything like the fridge? — That’s true but not important. A model is not a copy of a real-world entity but a construct supposed to solve a problem. 2/9

Hence, the correct question to ask is: what problem does this model solve? In our apartment, we have a weird entry to the kitchen. The cardboard’s is in the size of the fridge’s width and depth, and the problem it solves is checking whether it can make it through the door. 3/9

As the saying goes, all models are wrong, but some are useful. This model is wrong in many different ways. Nevertheless, it is useful. It helped us to decide whether to buy the fridge or look for a smaller one. 4/9

Building an exact, 3d cardboard copy of the fridge would definitely be a fun project! But would it make the solution any better or more precise? No. If the 2d model fits, a 3d model will fit as well. 5/9

But what about the fridge’s height? What if the base fits, but it’s too tall for that door? Would that justify building a 3d cardboard fridge? — Nope. A simple tape measure does the job. What is a tape measure in this case? — Another simple model. 6/9



So we have two models of the same fridge to check whether it can make it through our weird kitchen door. That piece of cardboard and a tape measure reflect the DDD’s approach to modeling business domains. 7/9

A domain model should omit the extraneous information irrelevant to its task. Also, no need to design a complex jack-of-all-trades model if multiple, much simpler models can effectively solve each problem individually. 8/9

Finally, a model is only valid for the specific problem it is supposed to solve: its bounded context. 9/9