Bill Sendewicz
September 7, 2021

Software Architecture vs. Code:
https://www.youtube.com/watch?v=9noJwoIivV8

Think and answer the following questions (and keep a digital document of this for your portfolio):

- Which types of diagrams are mentioned in the talk?

UML
"4+1" View
Layered Architecture diagram
Hexagon diagram
Onion diagram
C4 model: System Context, Containers, Components, Classes
Static model

- What is the standard for modeling software mentioned in the movie?

UML

- Why is nobody using it?

it is widely considered too complex and too low-level, and thus, nobody knows UML. Others say they don't need it because they use the Agile and/or test driven software development.

- What is the model-code gap and what is the problem with it?

Architecture models show one thing and source code shows another thing. The things that each shows are different. For example, architecture models show abstract things like components, but the programming language (source code) currently does not show these abstract concepts. Then there are things that the architecture models include, such as design decisions and constraints, that the source code cannot include. Architecture models are abstract and intensional whereas source code must be concrete and extensional to execute on machines. Architectural models are high-level and abstract and source code is very detailed and concrete.

- What do you personally think about layers for modeling software?

I think the C4 model has promise, where it's possible to zoom in or out depending on the level of abstraction needed and the particular level of detail the developer is coding.

- How do you think about Systems Modeling after watching this movie?

I think it's important, but I thought that before watching this presentation. But it reaffirmed my belief that architecture models should be constructed and agreed upon before any coding happens. However, I wasn't aware that there were architectural model diagrams that allowed a developer to zoom in or out to different levels of abstraction depending on the coding task, so that was valuable to learn.

- Anything else remarkable from the movie, we should share/discuss?

Simon Brown is advocating for the C4 model as the architectural standard. Is C4 a replacement for UML or just a framework for how UML diagrams should interrelate with one another? I didn't quite understand the relationship between the two.