

Refocusing your domain (until a better title)

Monday 23 October 2017 16:25 (30 minutes)

This forum's subject is from 'Zero to Production'. But what exactly goes on during the 'zero' phase? What are the questions that you and your team ask yourselves before starting to code, and which are the first decisions you make?

'Apache or Nginx? Flask or Django? PostgreSQL or MongoDB? Containers or VMs?', are usual questions that are brought forward during the design phase. Albeit cool and exciting, these issues are the wrong things to consider ourselves with during the first stages of software development.

What is the domain your application operates on? What are your core entities and how do they interact with each other on a logical level? Which are the business rules that would persist whether you would be creating a Web application, a Command Line Interface or simple drawings on a whiteboard? Not only should these issues be the first for a developer to consider but their importance should be clearly reflected in the application's code.

In 2012, Robert C. Martin (Uncle Bob) proposed a layered software architecture pattern named 'The Clean Architecture' (<https://8thlight.com/blog/uncle-bob/2012/08/13/the-clean-architecture.html>). Built around the Dependency Inversion Principle (https://en.wikipedia.org/wiki/Dependency_inversion_principle), this pattern suggests a code structure that places the application domain (business entities & use cases) in its very core. It aims in achieving separation of concerns and produce software systems that are testable, adaptable and independent of external systems.

This talk will be a general presentation of the above architecture followed by a brief demonstration of the pattern (probably in Go).

Primary author: Mr TRIANTAFYLLIDIS, Nikolaos Petros (CERN)

Presenter: Mr TRIANTAFYLLIDIS, Nikolaos Petros (CERN)