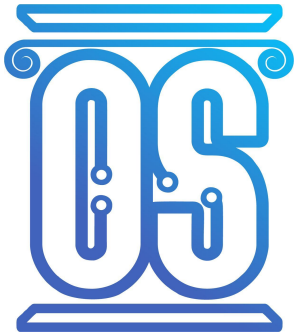


Mastering the Basics of DDD

Domain-Driven Design with Java

Otávio Santana, Founder
otavio@os.expert

Technology as strategic resource

Everything is around software!

“Every business is a software business” CMMI

“Every Company is a Data Company” CIO Network

“Every Company is a Software Company” Forbes

Software Development

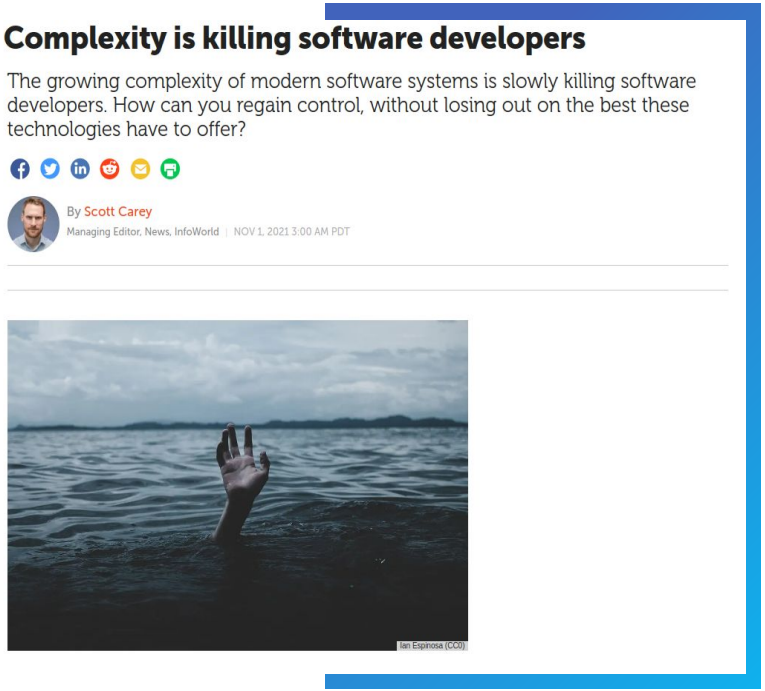
Failure?

- Hyper-Focused Planning And Design
- Unclear Or Undefined Client Expectations
- Unexpected Complexities
- Poor Collaboration Between The Product And Engineering Teams

The Complexity Paradox

The More Answers We Find, the More Questions We Have

- Developer experience is a market
- Trade-offs
- The hype effect

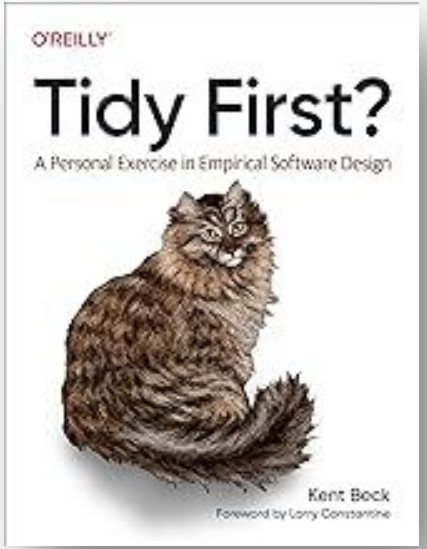



Otávio Santana
@otaviojava

Evolutionary Software?

No ready to change anything!

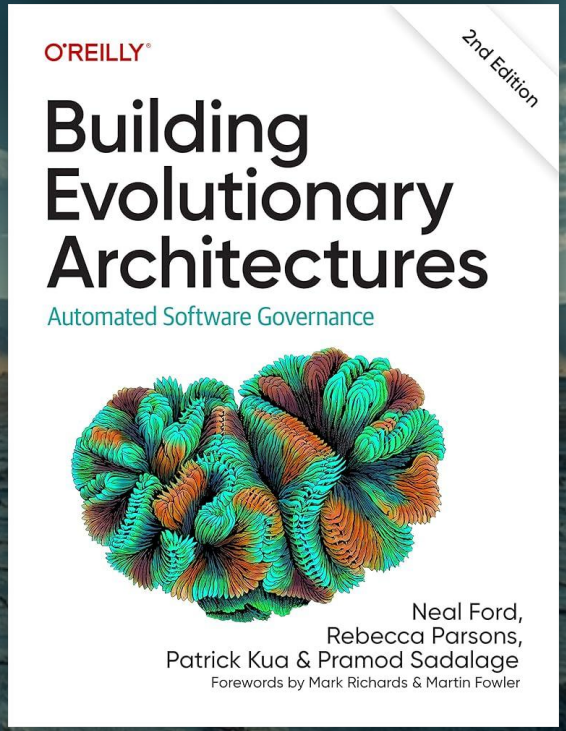
"To prepare to write their classic text **Structure Design**, Ed Yourdon and Larry Constantine examined programs to find out what made them so **expensive**. They noticed that the expensive programs all had one property in common: **changing one element required changing other elements**."



Software Erosion

Continuous development

"Also known as software rot, code rot, software decay, or software entropy, software rot is either a **slow deterioration** of software quality **over time** or its **diminishing responsiveness** that will eventually lead to software **becoming faulty**."



Software Engineering?

A super inefficient way to build something

We waste time in meetings

Create complexity

Not ready for change

Premature legacy

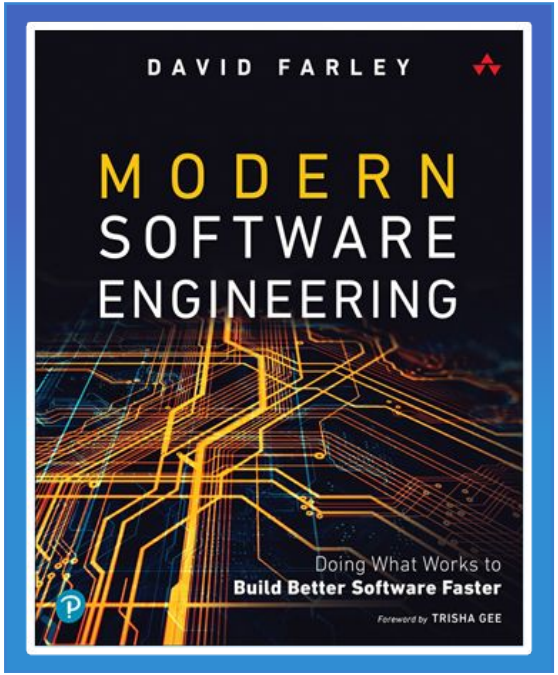
No achieve client goals


Otávio Santana
@otaviojava

Software Engineer

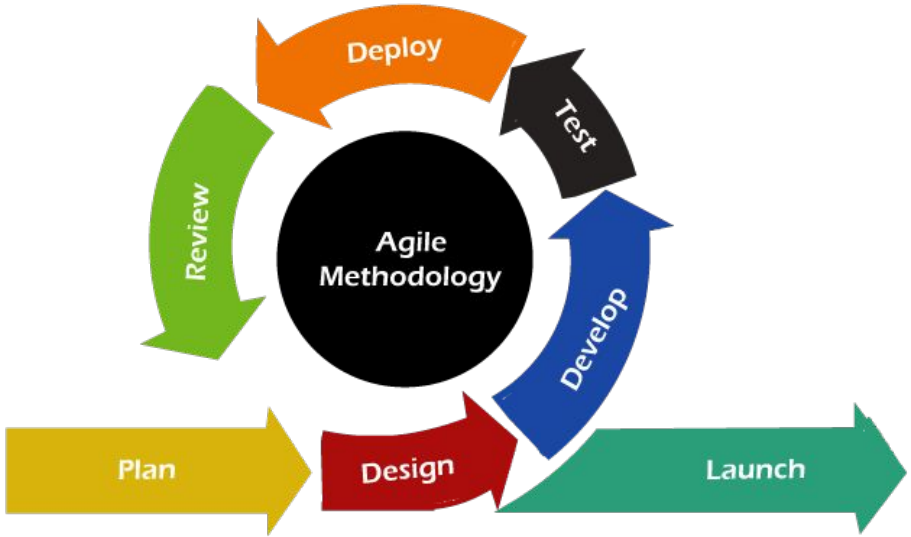
Definitions?

“Software engineering is the application of an empirical, scientific approach to **finding efficient, economical solutions** to practical problems in **software**.”



Agile

Agile is dead. Long live agility

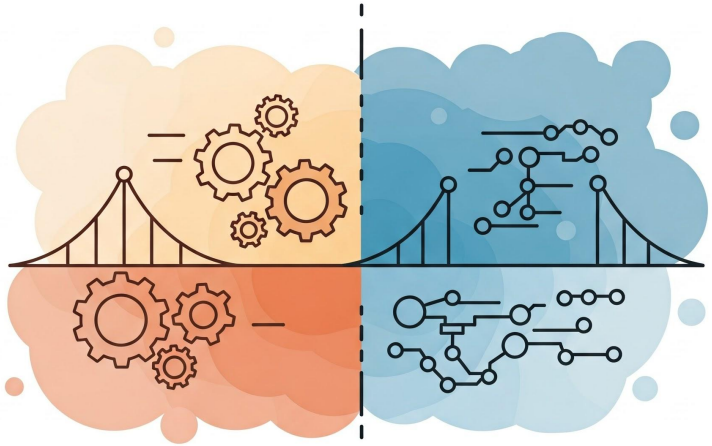


Digital.ai 2023

- 40% reported organizational resistance to change.
- 33% pointed to lack of leadership support.
- only ~31% of Agile projects are considered “successful”

Domain Driven Design

Designing software to meet business goals with less rework



Otávio Santana
@otaviojava

DDD?

The biggest misunderstanding in the software industry!

It is a framework!

Repository, Entity

Java

Microservices

Code only

Complex

Otávio Santana
@otaviojava

DDD!

Handling and focus to the Business

Domains

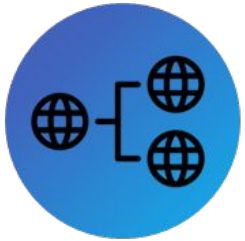
Transfer
Knowledge

Tactics

Strategy

Strategic DDD

The biggest mistake when implementing DDD



Domains and Subdomains

Context Mapping

Ubiquitous language

Bounded Contexts

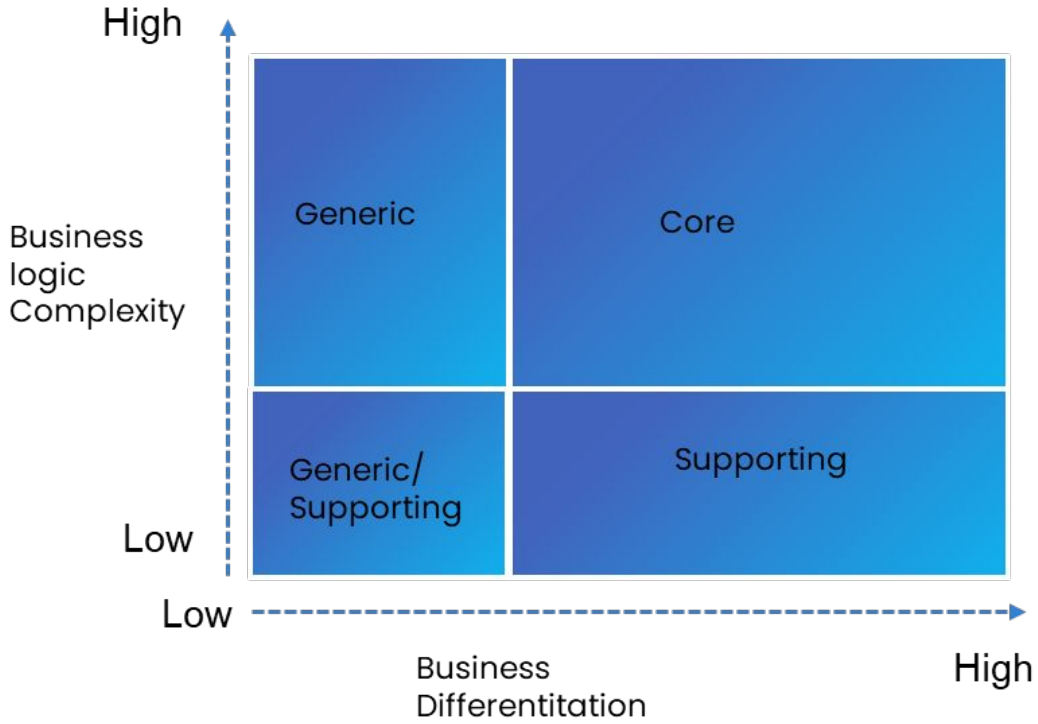
Domains

Central area of a company's operations

Subdomain Type	Role	Business Differentiation	Complexity of Business Logic
Core Subdomain	Unique to the company, defining its identity and competitive advantage	High	High
Generic Subdomain	Common across all companies, standard business activities	Low	High
Supporting Subdomain	Supports core business activities without providing direct competitive advantage	Medium	Varies

Domains

The subdomains types



Ubiquitous Language

The core's communication

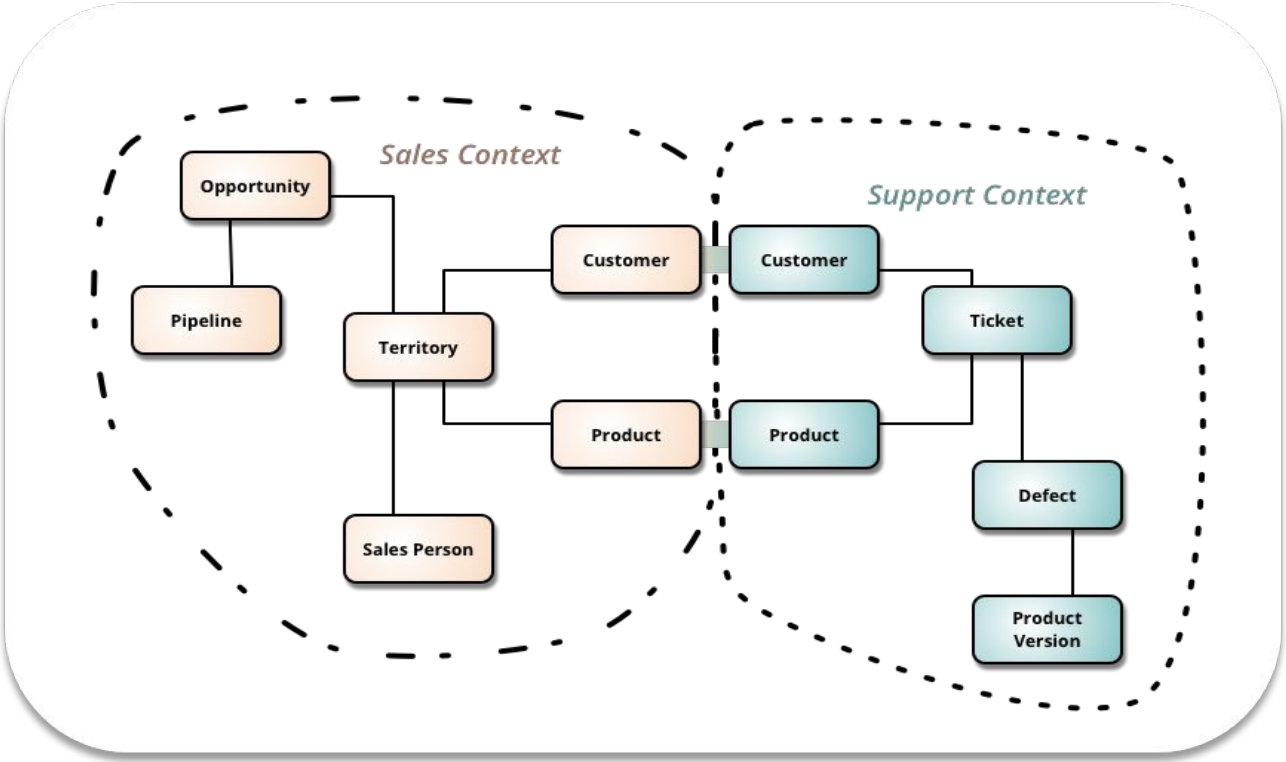
Define common terminology

The same word can vary from context

The common language between experts and the engineering team.

Bounded Contexts

Subset of the ubiquitous language




Otávio Santana
@otaviojava

Context Mapping

The relationships between these contexts

Shared Kernel

Customer-Supplier

Conformist

Open Host Service

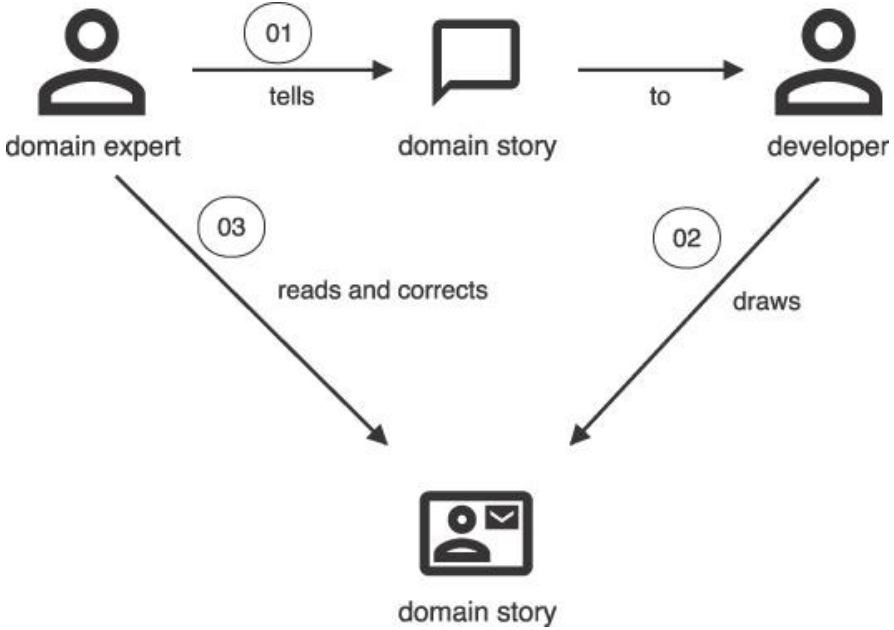
Anticorruption Layer

Published Language

Separate Ways

Domain Storytelling

A Collaborative, and Visual Way to Build DDD



Tactic DDD

The second step

Entity

Service

Repository

Aggregates

Value Object

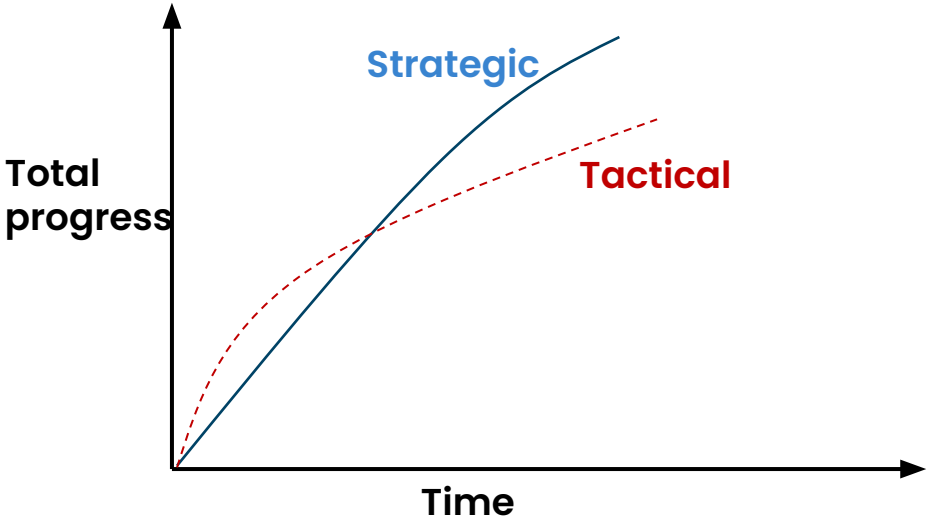
Factories

Events

Strategic and Tactical

Focus on tactics is the start of a huge mistake

Get something done
Working code isn't enough
Tactical tornado



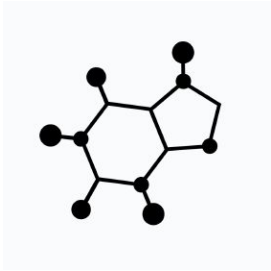
Governance

Automating processes and good practice rules

ArchUnit

PMD

checkstyle



Using Annotations

Getting architecture evidences

```
@Entity
public class CreditCard {

    @Identity
    private BigInteger id;

    private String number;

    private String name;

    private YearMonth expiry;
}
```

Otávio Santana
@otaviojava

Governance

Automating processes and good practice rules

```
@AnalyzeClasses(packages = "expert.os.examples")
public class IntegrationSampleTest {


    @ArchTest
    private ArchRule dddRules = JMoleculesDddRules.all();

    @ArchTest
    private ArchRule layering = JMoleculesArchitectureRules.ensureLayering();

}
```

PMD

Static source code analyzer

Bugs

Dead code

Security

Over Complicated expressions

Suboptimal code

Classes with high Cyclomatic

Complexity measurements.

Duplicate code

Otávio Santana
@otaviojava

Checkstyle

Create your style validations

Naming conventions of attributes and methods

The presence of mandatory headers

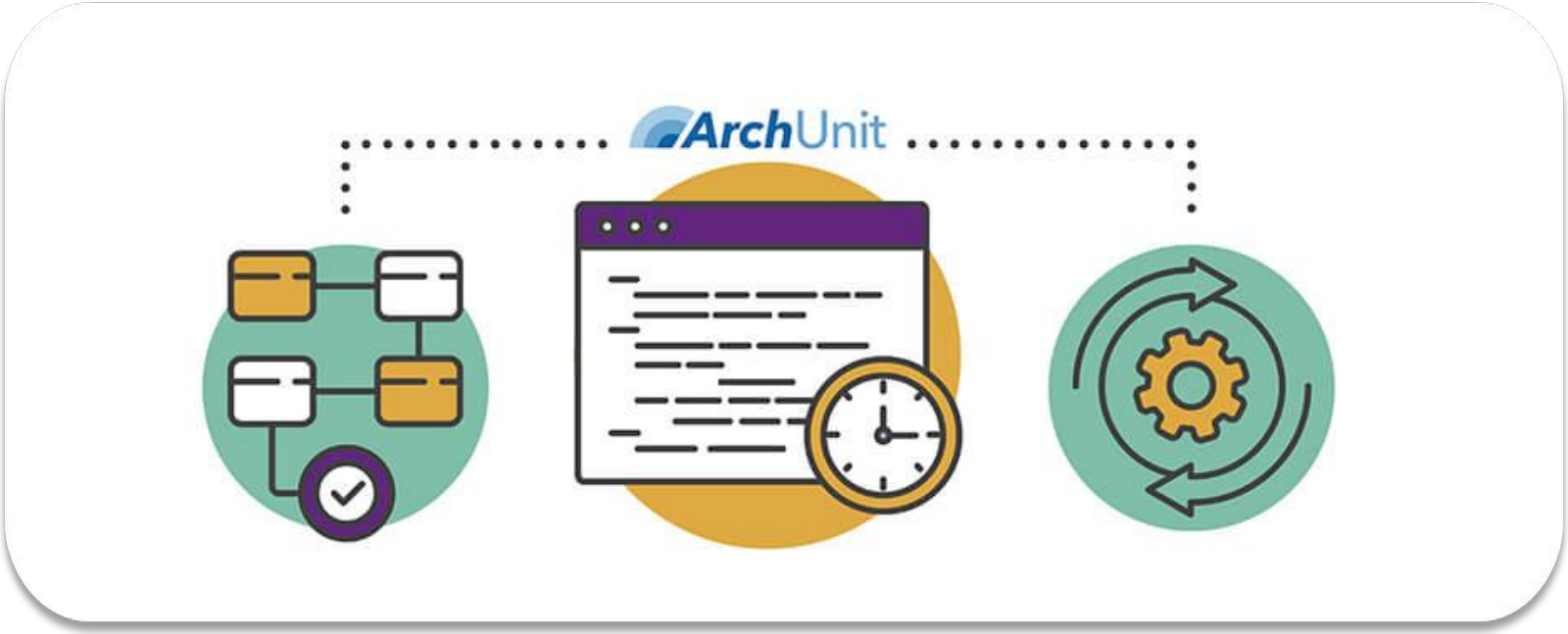
The practices of class construction

Multiple complexity measurements

Otávio Santana
@otaviojava

ArchUnit

Putting test at your Architecture and Design



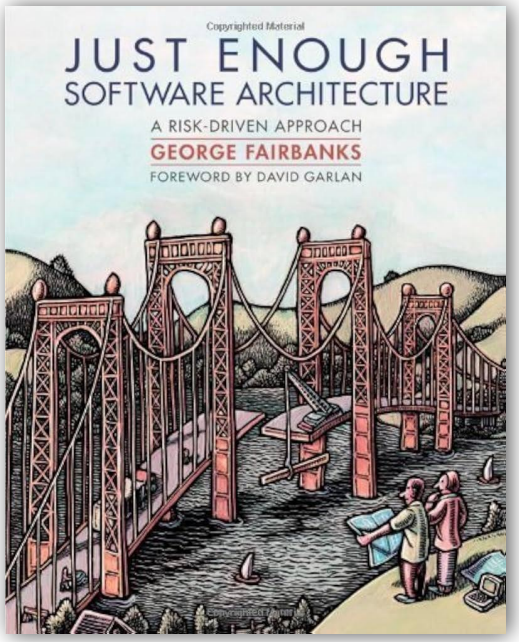
Otávio Santana
@otaviojava

xMolecules

Architectural abstractions in code.

"There is always a leak between the language and the architecture you want to express. Language, like any other tool, shapes what you can do with it, and sometimes it doesn't allow you to express things the way you'd want from an architectural standpoint."

Architecturally Evident Applications – How to Bridge the Model-Code Gap?

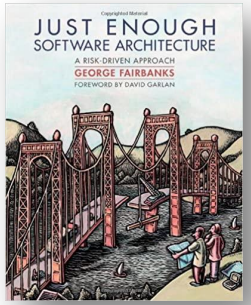


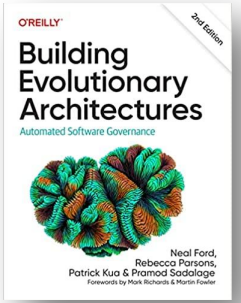
https://xmolecules.org/aea-paper.pdf

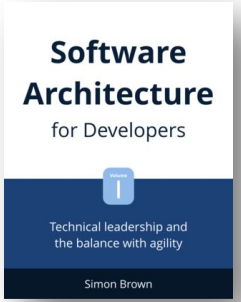
Otávio Santana
@otaviojava

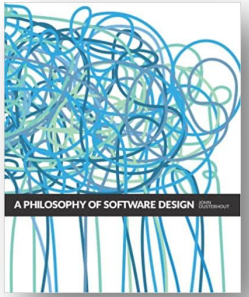
Software Architecture

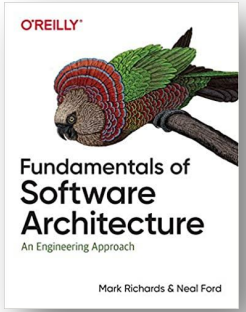
There is no space, the business is the goal!

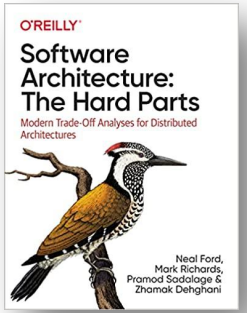


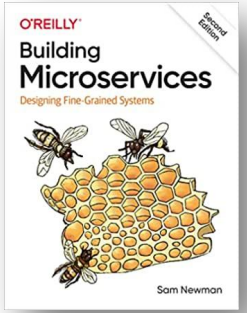








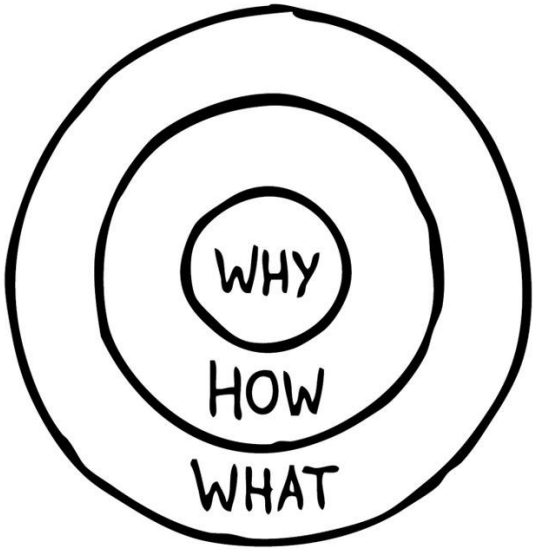




Otávio Santana
@otaviojava

Architecture

The laws

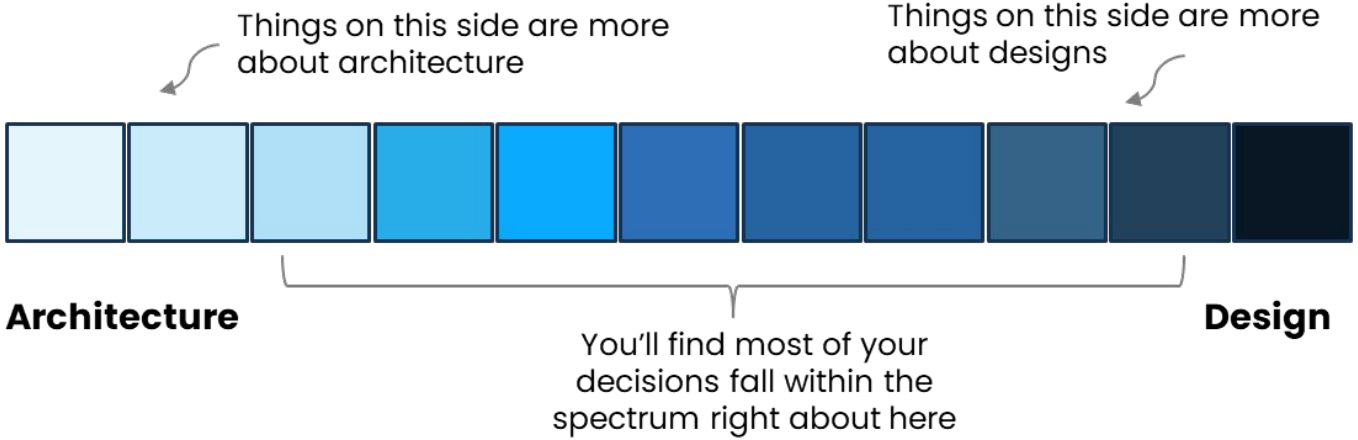


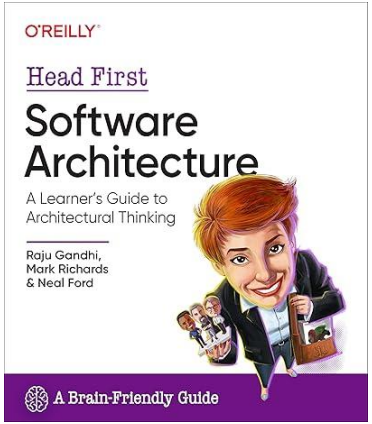
- Everything in software architecture is a trade-off
- Why is more important than how

Otávio Santana
@otaviojava

Architecture vs Design

The boundaries to DDD

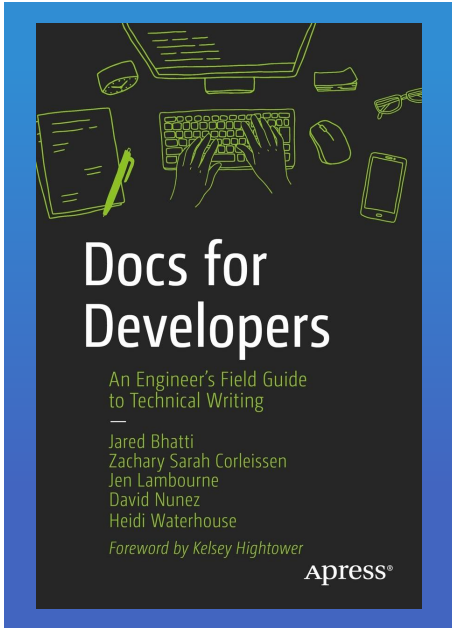




Documentation

The reader is the user

- **Code**
- **Changelog**
- **README**
- **API**



Otávio Santana
@otaviojava

Documentation

Architectural perspective

C4-model

Architecture's map

Tech-radar

Technologies's view

ADR

Don't repeat the error

Communication

A clear direction

Otávio Santana

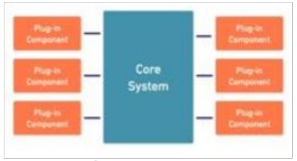
@otaviojava

The Architecture styles

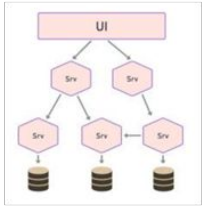
There is a world beyond microservices and Hexagon model

CQRS Architecture

Layered (n-tier) Architecture

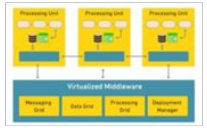


Microkernel Architecture

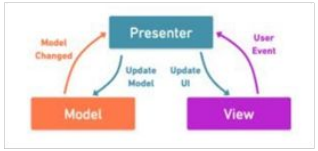


Micro service Architecture

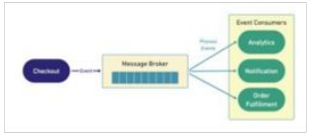
Orchestration Architecture



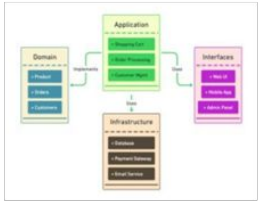
Spaced- Based Architecture



MVP Architecture



Event-Driven Architecture

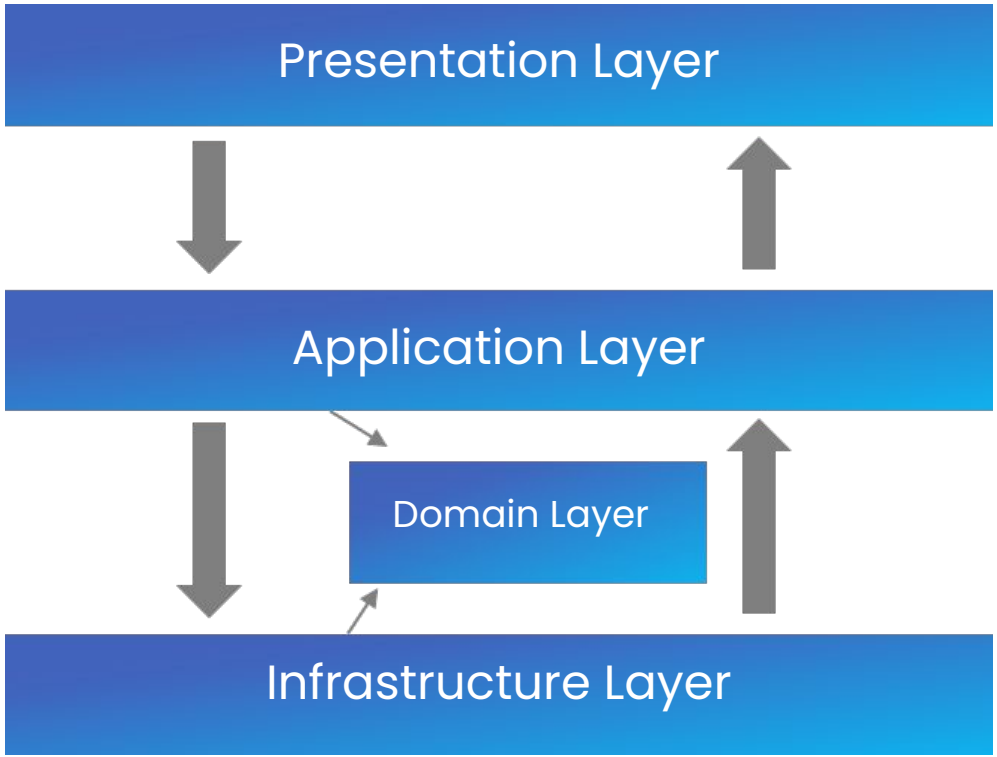


DDD Architecture

Otávio Santana
@otaviojava

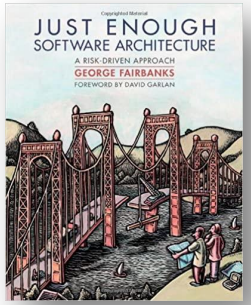
Architecture DDD

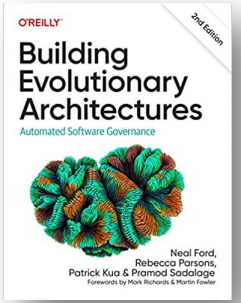
Using Layers to abstraction

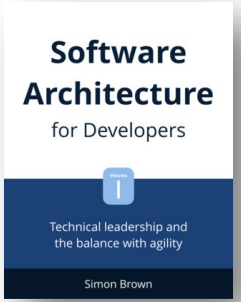


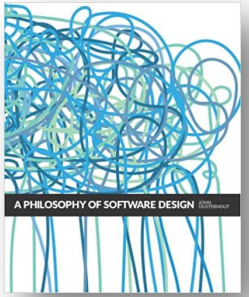
Software Architecture

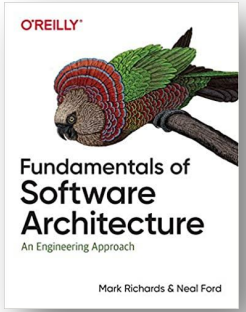
There is no space, the business is the goal!

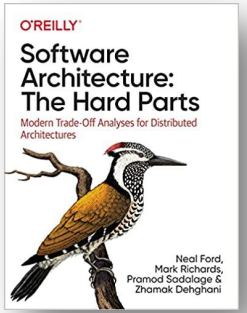


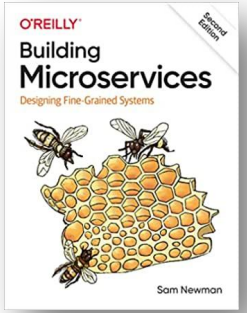












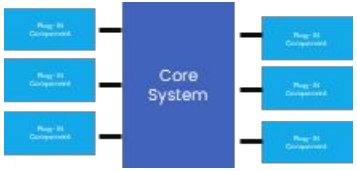
Otávio Santana
@otaviojava

The Architecture styles

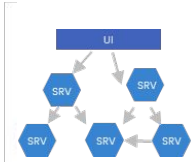
There is a world beyond microservices and Hexagon model

CQRS Architecture

Layered (n-tier) Architecture

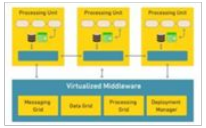


Microkernel Architecture

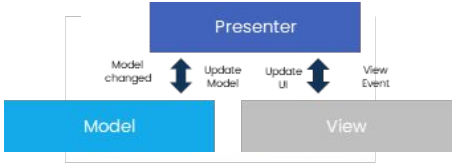


Micro service Architecture

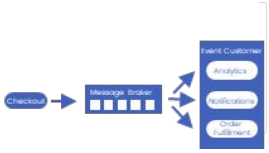
Orchestration Architecture



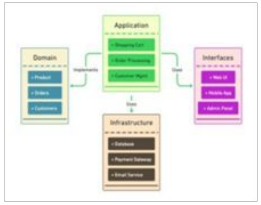
Spaced- Based Architecture



MVP Architecture



Event-Driven Architecture

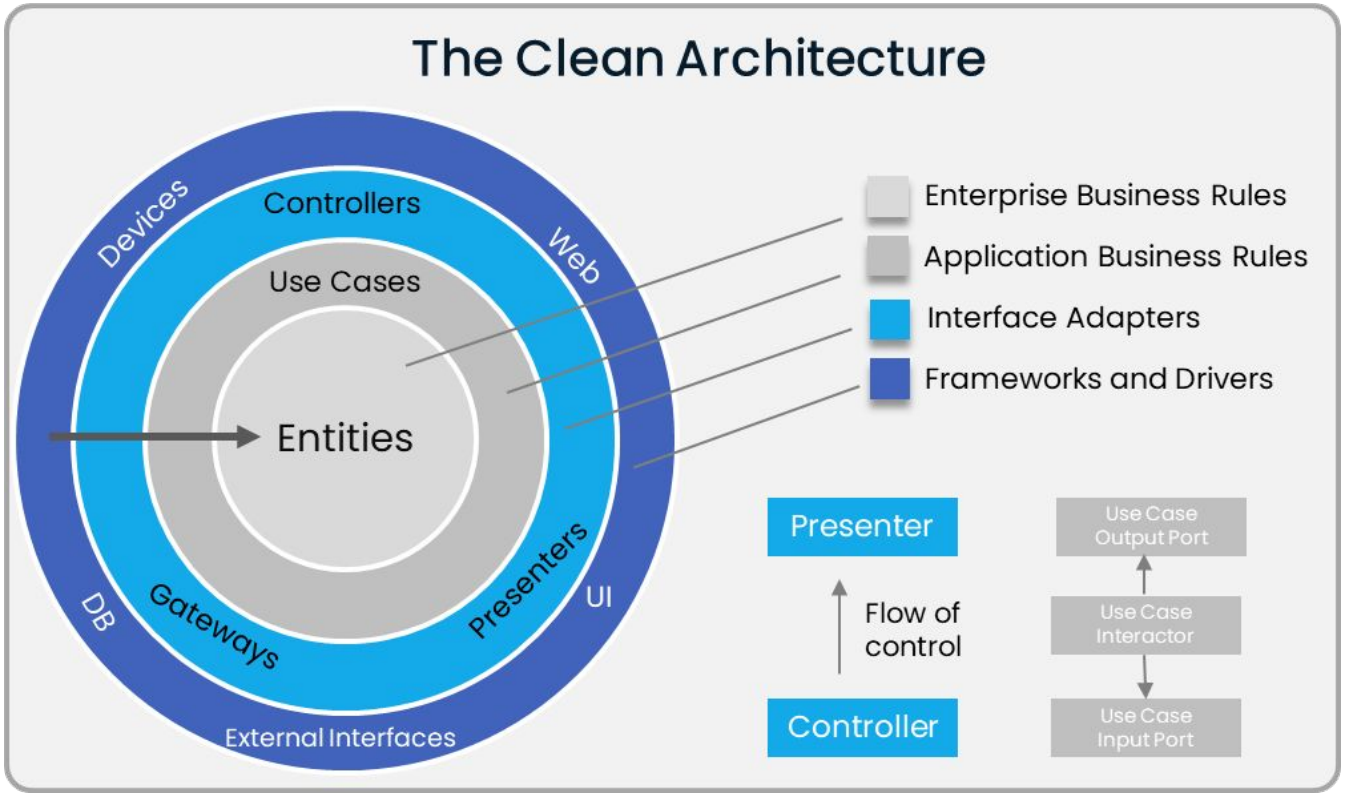


DDD Architecture

Otávio Santana
@otaviojava

Clean Architecture

How to Combine it with DDD?



Otávio Santana
@otaviojava

Refactoring

Be ready for change

Better Readability

Reduced Complexity

Improved Maintainability

Reduced Technical Debt

Enhance Performance

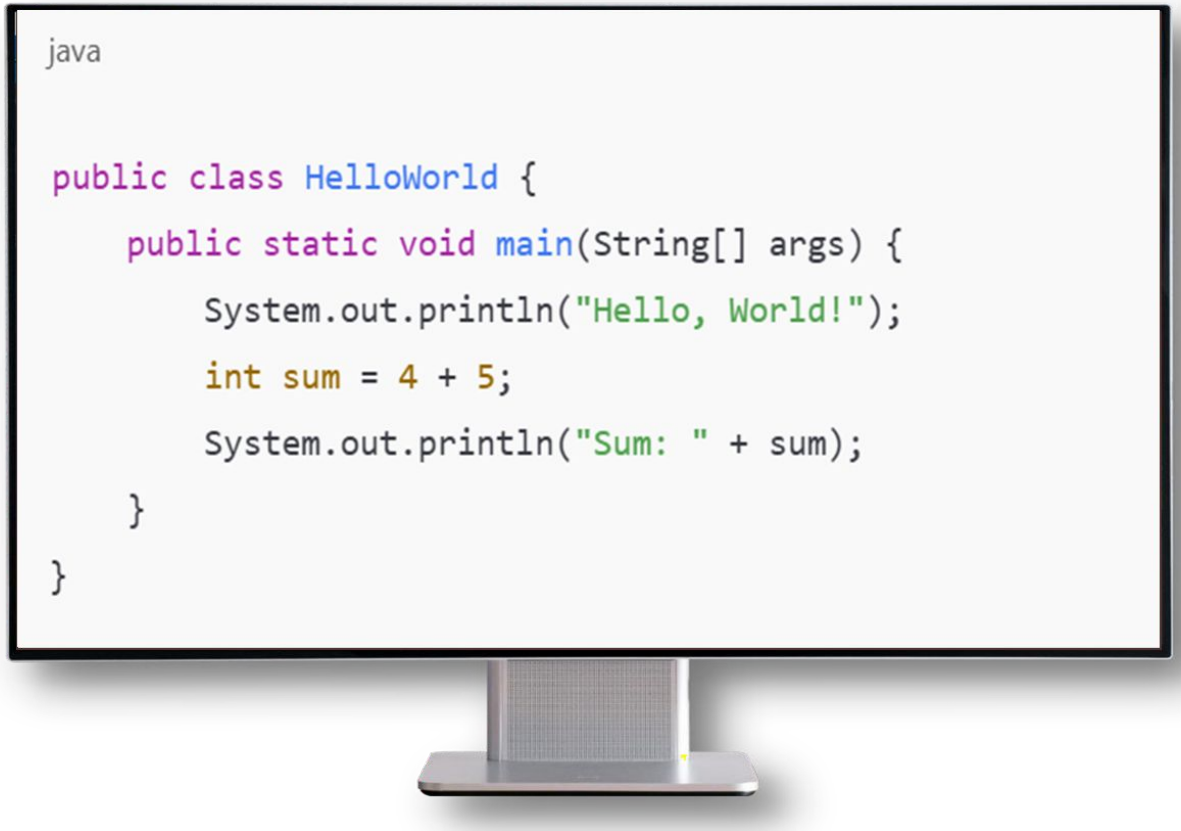
Effortless Extensibility

Fight against Software Erosion

Otávio Santana
@otaviojava

Show me the code

Let’s show the code structure




Otávio Santana
@otaviojava

Software Engineering

The principle of the ultimate sophistication Engineer

- Documentation
- Persistence
- Code Design

- Software Architecture
- Leadership
- Test

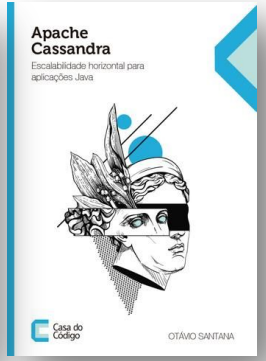
Otávio Santana
@otaviojava

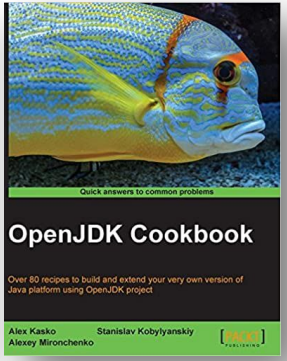
Expert

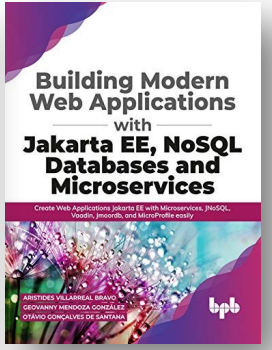
Otavio Santana

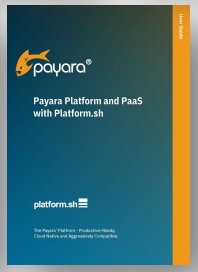
Software Engineer & Architect

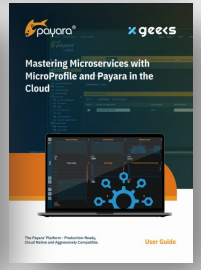
- [x] Java Champion, Oracle ACE
- [x] JCP-EC-EG-EGL
- [x] Apache and Eclipse Committer
- [x] Jakarta EE and MicroProfile
- [x] Duke Choice Award
- [x] JCP Award
- [x] Book and blog writer

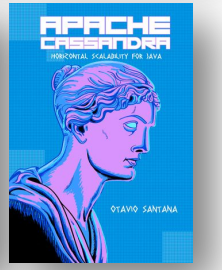










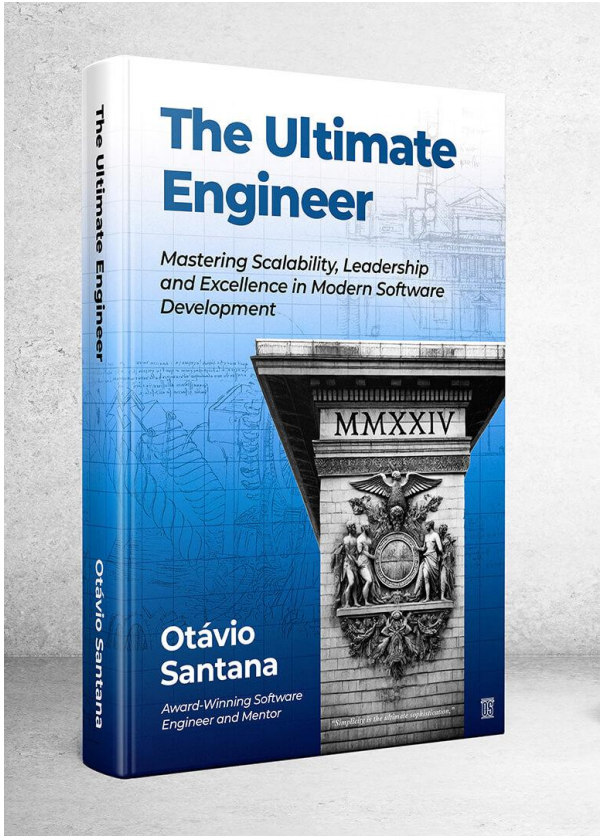


Otávio Santana
@otaviojava

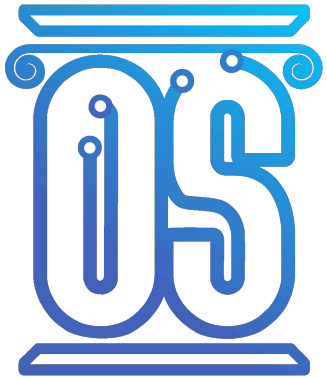
Ultimate Engineer

A practical guide to becoming a high-impact software engineer and architect

https://os.expert/ebook-the-ultimate-engineer/



Otávio Santana
@otaviojava



Thank you!

Otávio Santana

OS Expert Founder, Software Engineer & Architect

otavio@os.expert

https://twitter.com/otaviojava

https://www.linkedin.com/in/otaviojava/

https://www.youtube.com/@otaviojava