


Modern Software Architecture (#1): Domain Driven Design

Modern Software Architecture with Domain Driven Design (DDD). Introduction on DDD and its relation to Microservices, Reactive Systems, BPM, Agile, CQRS and Event Sourcing.



Kai Niklas

Following

Oct 14, 2018 · 6 min read

🐦

🌐

📘

📺

🔖

👤

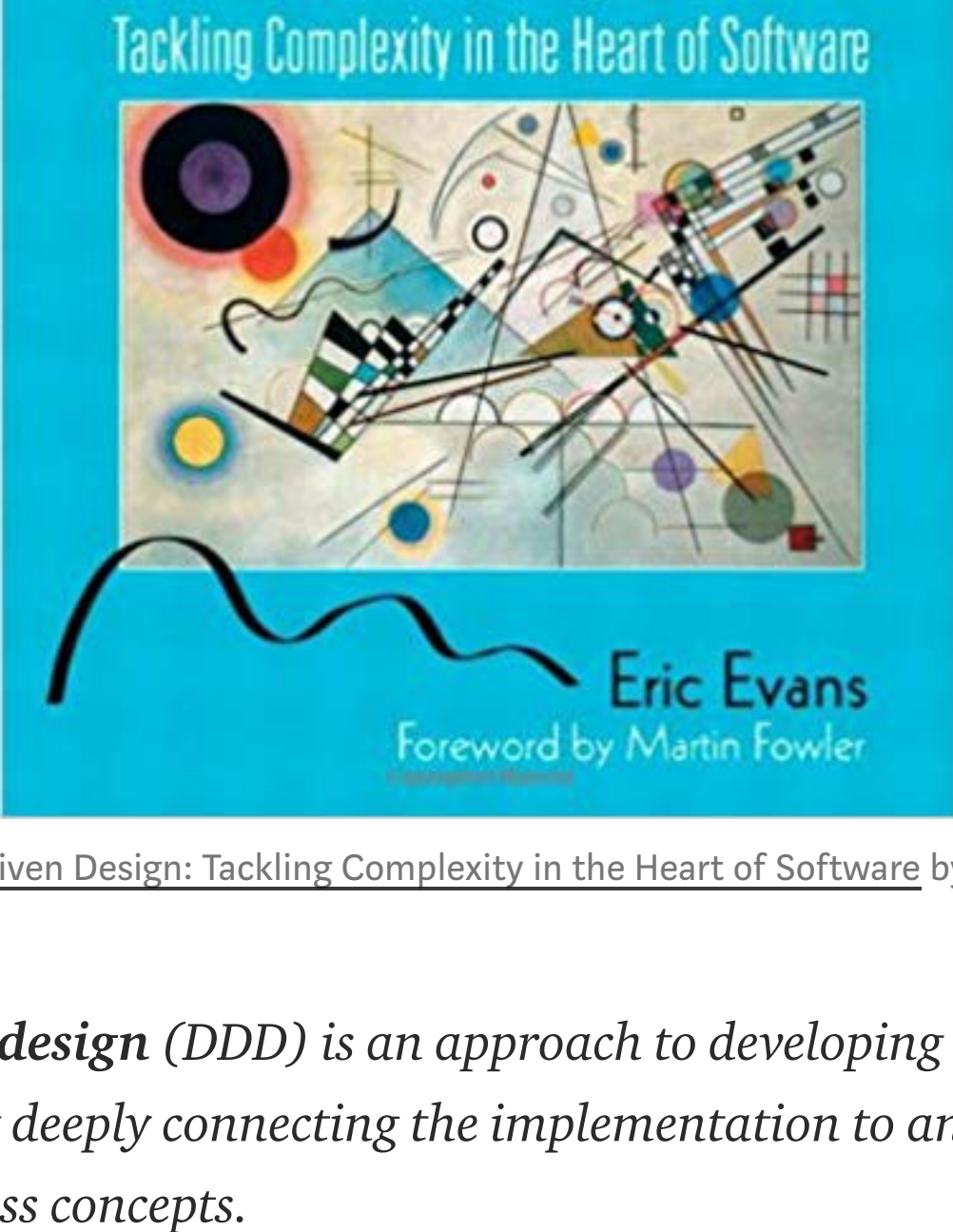
Welcome to the first issue on *Modern Software Architecture*. Regularly, I collect and curate interesting and insightful news and stories for a specific topic within the field of *Modern Software Architecture*. I want to share my experience as software and enterprise architect with you. This article is relevant for software developers, engineers and architects who want to up-skill themselves and learn from expert experiences.



Issue #1 on Modern Software Architecture focuses on Domain Driven Design

Quick Introduction to DDD

Domain driven design (DDD) is now around for ~15 years. The name was coined by [Eric Evans](#) in his book “[Domain-Driven Design: Tackling Complexity in the Heart of Software](#)”.



Domain-Driven Design: Tackling Complexity in the Heart of Software by Eric Evans

Domain-driven design (DDD) is an approach to developing software for complex needs by deeply connecting the implementation to an evolving model of the core business concepts.

Its premise is:

- Place the project's primary focus on the core domain and domain logic
- Base complex designs on a model
- Initiate a creative collaboration between technical and domain experts to iteratively cut ever closer to the conceptual heart of the problem.

(Source: [dddcommunity.org](#))

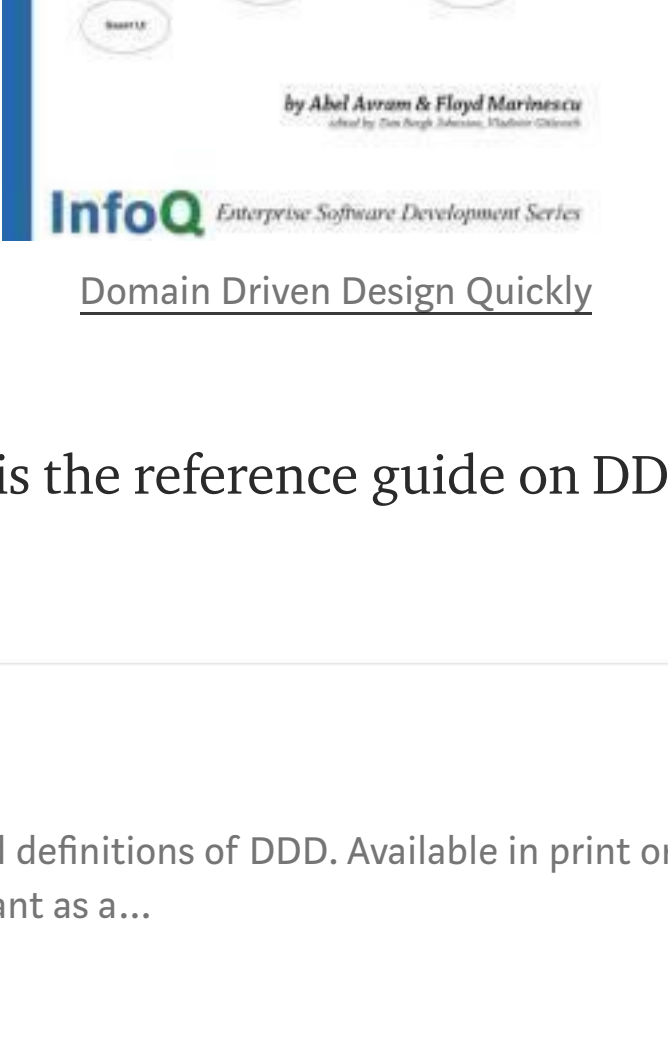
DDD is not a technology, rather it introduces terms, practices and principles to support decision making in complex business domains. If the complexity of the domain is not addressed early and properly within the application design, the best technology won't help to create a system which is maintainable, extendable, etc.

Further, DDD introduces an ubiquitous language which can improve collaboration of cross-functional teams. This language helps to bridge the famous “symmetry of ignorance” between people with different backgrounds and viewpoints, e.g., domain experts understand the practice and system designer understand the technology. (Further reading: [Symmetry of Ignorance, Social Creativity, and Meta-Design](#) by [Gerhard Fischer](#))

DDD takes some time to grasp, but it is worth the effort. In the following I give you some resources to better understand what DDD is all about, and in which contexts DDD is used today.

Get to know DDD — A shortcut

Many people state that the original book by Evans is cumbersome to read and understand. But most say, that it is a must read for every software developer or architect who want to build modern software systems. If you are searching for a shortcut, then probably this Minibook on DDD is your next read: “[Domain Driven Design Quickly](#)”, which you find on the InfoQ page for free download.



Domain Driven Design Quickly

Another great resource is the reference guide on DDD which can be found [here](#):

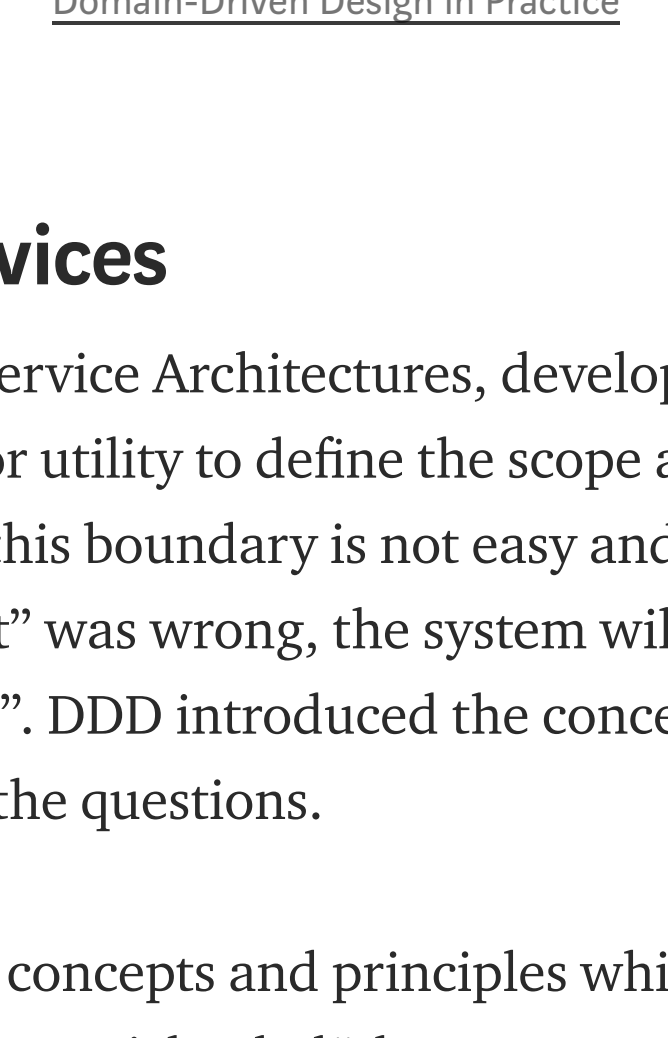
DDD Reference

A summary of the patterns and definitions of DDD. Available in print or as a free PDF. This document is meant as a...

domainlanguage.com

Practical experience on DDD

If you already know the basics, then you may ask: “How do I apply DDD in practice, and which challenges will I face?”. Questions like this and some solutions can be found in the recently released Minibook: “[Domain-Driven Design in Practice](#)”, which can be found on the InfoQ page for free download.

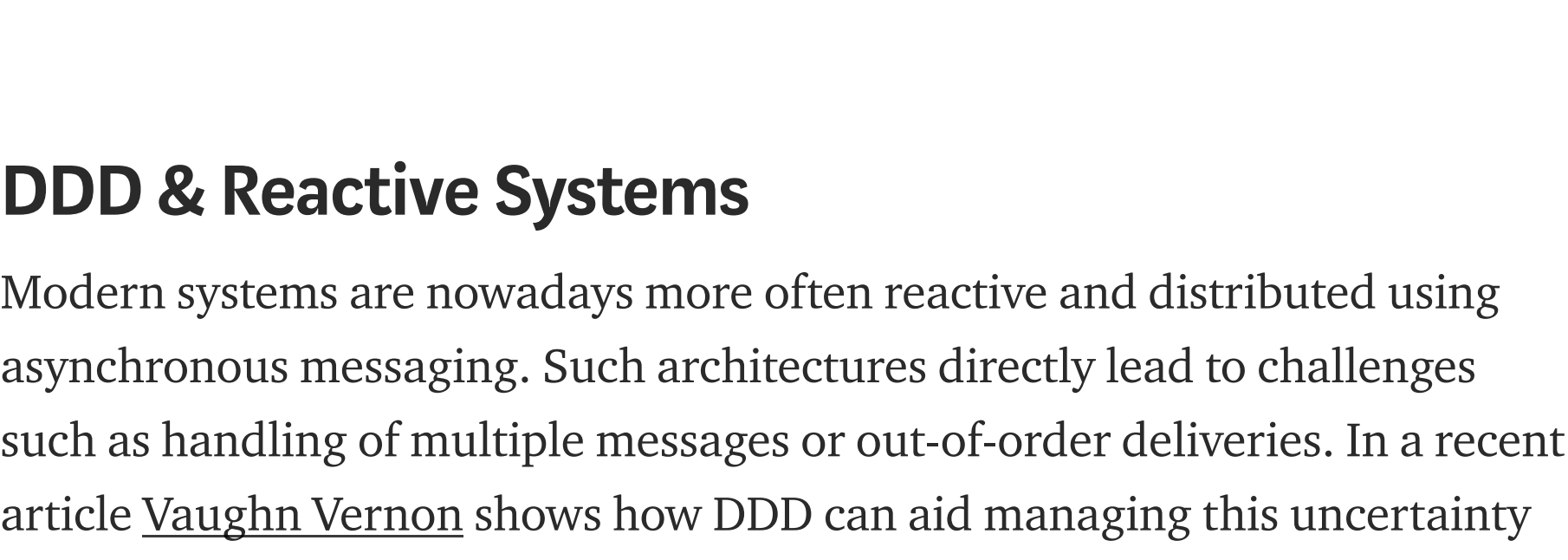


Domain-Driven Design in Practice

DDD & Microservices

In the context of Microservice Architectures, developers and architects searched for a method or utility to define the scope and size of each Microservice. Defining this boundary is not easy and if the fundamental decisions on how to “cut” was wrong, the system will rapidly evolve into the famous “[big ball of mud](#)”. DDD introduced the concept of *bounded contexts* which helps answering the questions.

But DDD contains more concepts and principles which help to create robust and maintainable systems. [Michael Plöb](#) gave a presentation on how DDD principles can help to create good Microservices.



microXchg 2016 — Michael Plöb : Microservices Domain Driven Design, why and how?

DDD & Reactive Systems

Modern systems are nowadays more often reactive and distributed using asynchronous messaging. Such architectures directly lead to challenges such as handling of multiple messages or out-of-order deliveries. In a recent article [Vaughn Vernon](#) shows how DDD can aid managing this uncertainty through the use of good modeling.

Modeling Uncertainty with Reactive DDD

Vaughn Vernon has written several books on DDD and reactive messaging patterns, and has found that the nature of...

www.infoq.com

DDD & BPM

DDD is also helpful in the domain of business process modelling (BPM) to avoid BPM Monoliths which [Bernd Rücker](#) explains:

Avoiding the “BPM monolith” when using bounded contexts

For example in microservice architectures or Domain-Driven Design (DDD)

blog.bernd-ruecker.com

DDD & CQRS & Event Sourcing

Event sourcing is an approach to store data using deltas rather than their effects. The associated database is called event store and uses only the SQL statements INSERT and SELECT. Although event sourcing is not part of DDD, both concepts complement each other very well. (Source: [Golo Roden on thenativeweb.io](#))

The design pattern CQRS describes the separation of the write side from the read side of an application. CQRS and DDD do not necessarily belong together, but complement each other excellently, which is why their combination is often used in practice. (Source: [Golo Roden on thenativeweb.io](#))

the native web | DDD & co., part 1: What's wrong with CRUD

This is one of the largest issues in everyday software development: the domain language and the technical language...

www.thenativeweb.io

DDD & Agile

When it comes to modelling, the first reaction is often: “Modeling does not fit into agile, because it is [big up front design](#) (BUFD) which is an [anti-pattern](#) in agile.”. For modeling in the context of DDD I disagree. Concepts from DDD even support the agile way very well. There are many agile thought leaders who support and push DDD, e.g., [Martin Fowler on his webpage](#) or Kent Beck on the back-cover of Evans on DDD. And if you want to get things right, sometime you need to think about it a bit longer. Don't just rush into it, especially if you apply sophisticated concepts such as CQRS & Event Sourcing.

[Michael Douglass](#) article explains it very well how DDD and Agile relate to each other:

Domain Driven Design & Agile — They Work Together!

I recently found myself tweeting a full, 140 character message. But, after it gained the attention of a few people, I...

codeburst.io

Too many definitions? Read the FAQ!

With DDD a lot of terms and concepts are introduced. Especially, if you dive deeper into DDD with CQRS and Event Sourcing. To understand how all these concept fit together, the following FAQ is a great resource.

CQRS, Event Sourcing and DDD FAQ

The ideas around CQRS are fascinating, but different enough from the way we do conventional development that some...

www.cqrs.nu

Want to read more?

Subscribe to my [newsletter on modern software architecture](#). Grow your skills and become a better software architect.

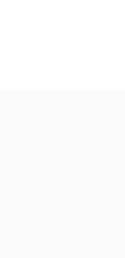
Software Architecture

Domain Driven Design

Software Engineering

Agile

Microservices

 195 claps

🐦


🌐

📘

📺

🔖

👤




WRITTEN BY

Kai Niklas

Software Architect | Software Engineer | Open-Minded | Paraglider | PhD | Innovation Principal Consultant | https://bettersoftwarearchitect.com

Following



Modern Software Architecture

Hand-picked posts on Modern Software Architecture.

Following

Write the first response

More From Medium

More from Kai Niklas

CI/CD with Angular 6 & Firebase & Bitbucket Pipelines

Kai Niklas in Quick Code
Dec 14, 2018 · 7 min read

244

📖

Related reads

Modular Monoliths— A Gateway to Microservices

Natalie Conklin in Design and Tech.Co
Mar 31, 2019 · 4 min read ★

687

📖

Related reads

(Micro)services with Event Notifications

Cengiz Han in hepsiburadatech
Feb 25, 2019 · 6 min read

1.2K

📖

Discover Medium

Welcome to a place where words matter. On Medium, smart voices and original ideas take center stage - with no ads in sight. Watch

Make Medium yours

Follow all the topics you care about, and we'll deliver the best stories for you to your homepage and inbox. Explore

Explore your membership

Thank you for being a member of Medium. You get unlimited access to insightful stories from amazing thinkers and storytellers. Browse

Medium

About Help Legal