

# Vortrag: Vom Big Ball of Mud zu DDD – ein Praxisleitfaden



Fachspezifischer Architekturentwurf  
Gastvortrag  
Freitag - 11.01.2019 - 13:30 Uhr - Raum 0501  
Christian Nockemann



Vom Big Ball of Mud zu DDD – ein Praxisleitfaden

„Domain-driven Design ist als Ansatz zur Modellierung von komplexer Software aus der modernen IT-Architektur nicht mehr wegzudenken. Bei der Konzeption neuer Software-Projekte hilft DDD in Form einer ubiquitären Sprache das Verständnis für die anstehenden Herausforderungen zu erkennen und zu lösen.“

Aber was ist mit bereits vorhandenen Software-Produkten? Auch schlecht strukturierte Legacy-Software („Big Ball of Mud“) kann mit der Einführung von DDD wieder in einen wartungsfähigen Zustand gebracht werden.

Dieser Vortrag beantwortet, anhand von Praxis-Erfahrungen aus Projekten in denen nachträglich DDD eingeführt wurde, folgende Fragen:

- Welche Vorteile bringt DDD überhaupt im Kontext von Legacy-Software?
- Wie erreicht man einen Zustand, aus dem heraus technische Änderungen überhaupt möglich sind?
- Was ist ein verteilter Monolith und wie kann er vermieden werden?
- Wie können Domänen identifiziert werden?
- Wie kann Kompatibilität in der Übergangsphase ermöglicht werden?
- Welcher Technologiestack eignet sich für die Umsetzung?

**viadee**<sup>®</sup>  
IT-Unternehmensberatung

**Technology  
Arts Sciences  
TH Köln**

Diesen **Freitag (11.01.2019)** bieten wir im FAE wieder einmal einen offenen Gastvortrag an. Diesmal besucht uns **Christian Nockemann von viadee** mit einem Vortrag zum Thema **"Vom Big Ball of Mud zu DDD - ein Praxisleitfaden"**. Alle Interessierten sind hierzu herzlich eingeladen, ab **13:30 Uhr in Raum 0501**.

## Über die Vortragenden:

*Christian Nockemann arbeitet seit 2009 als IT-Berater und Software-Architekt bei der viadee IT-Unternehmensberatung. Sein Fokus liegt auf dem Design und der Entwicklung von Java-basierten Enterprise-Anwendungen. Eine besondere Bedeutung gibt er dabei Qualitätskriterien des Softwareerstellungsprozesses wie bspw. der Anwendung des Domain-driven Designs, dem sinnvollen Einsatz von Entwurfsmustern und der Einhaltung von Clean-Code-Richtlinien.*

## Über das Thema:

*„Domain-driven Design ist als Ansatz zur Modellierung von komplexer Software aus der modernen IT-Architektur nicht mehr wegzudenken. Bei der Konzeption neuer Software-Projekte hilft DDD in Form einer ubiquitären Sprache das Verständnis für die anstehenden Herausforderungen zu erkennen und zu lösen.“*

*Aber was ist mit bereits vorhandenen Software-Produkten? Auch schlecht strukturierte Legacy-Software („Big Ball of Mud“) kann mit der Einführung von DDD wieder in einen wartungsfähigen Zustand gebracht werden.*

*Dieser Vortrag beantwortet, anhand von Praxis-Erfahrungen aus Projekten in denen nachträglich DDD eingeführt wurde, folgende Fragen:*

- *Welche Vorteile bringt DDD überhaupt im Kontext von Legacy-Software?*
- *Wie erreicht man einen Zustand, aus dem heraus technische Änderungen überhaupt möglich sind?*
- *Was ist ein verteilter Monolith und wie kann er vermieden werden?*
- *Wie können Domänen identifiziert werden?*
- *Wie kann Kompatibilität in der Übergangsphase ermöglicht werden?*
- *Welcher Technologiestack eignet sich für die Umsetzung?“*