

# Literatur

## Bücher

### Grundlagen

- **Andrew Hunt, David Thomas (1999): The Pragmatic Programmer: from journeyman to master, Addison-Wesley**
  - Guter Überblick über Methoden, Strategien und Herausforderungen in der Softwareentwicklung. Sehr praktisch orientiert, gut strukturiert und einfach verständlich.
- **Kent Beck (2023): Tidy First?: A Personal Exercise in Empirical Software Design, O'Reilly**
- **David Farley (2022): Modern Software Engineering: Doing What Works to Build Better Software Faster, Addison-Wesley**
  - Sehr gute Übersicht über moderne Softwareentwicklungsmethoden. Solide Grundlage für architektonische Arbeiten und Entscheidungen.

### Architektur

- **Eric Evans (2003): Domain-Driven Design: Tackling Complexity in the Heart of Software, Addison-Wesley Professional**
  - Standardwerk für Software-Architektur. Sehr gute Grundlage für die Entwicklung von grösseren Anwendungen.
- **Vaughn Vernon (2013): Implementing Domain-Driven Design, Addison-Wesley**
- **Martin Kleppmann (2017): Designing Data-Intensive Applications, The big ideas behind reliable, scalable and maintainable systems, O'Reilly**
- **Steve Freeman, Nat Price (2009): Growing Object-Oriented Software Guided by Tests, Addison-Wesley**
  - Captures and describes in great detail the London School of TDD
- **Robert C. Martin (2018): Clean Architecture: A Craftmans Guide to Software Structure and Design, Prentice Hall**

### Deployment

- **Jez Humble, David Farley (2010): Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation, Addison-Wesley Signature Series (Fowler)**
  - Wichtiges Grundlagenwerk für eine heute weit verbreitete Entwicklungsmethode: Continuous Delivery

### Integration

- **Gregor Hohpe, Bobby Wolf (2004): Enterprise Integration Patterns, Pearson Education**
  - Great, in-depth catalogue of patterns for software system integration

### Programmiersprachen

- **Douglas Crockford (2008): JavaScript: The Good Parts, O'Reilly**
- **Douglas Crockford (2018): How JavaScript Works, virgule solidus**
- **Brian W.Kernighan, Dennis M. Ritchie (1988): The C Programming Language, Person Addison-Wesley**

### Standartwerke

- **Martin Fowler (2018): Refactoring: Improving the Design of Existing Code, Addison-Wesley Professional**
- **Erich Gamma et al. Design Patterns (1994): Elements of Reusable Object-Oriented Software, Addison-Wesley Professional**

- Classic on design patterns that's still relevant today.

## **Verteilte Systeme**

M. van Steen and A.S. Tanenbaum (2023): Distributed Systems, Version 4.01., distributed-systems.net

## **Blogs and Collections**

- Awesome Event-Driven Architecture Awesome
- C4 Model Introduction
- Gut dokumentiert: Architecture Decision Records
- enterpriseintegrationpatterns.com
- martinowler.com
- continuousdelivery.com/

## **Articles**

### **Domain Driven Design**

- <https://martinfowler.com/bliki/DomainDrivenDesign.html>
- <https://martinfowler.com/bliki/UbiquitousLanguage.html>
- <https://martinfowler.com/bliki/BoundedContext.html>
- [https://martinfowler.com/bliki/DDD\\_Aggregate.html](https://martinfowler.com/bliki/DDD_Aggregate.html)
- <https://martinfowler.com/bliki/EvansClassification.html>
- <https://martinfowler.com/bliki/ValueObject.html>

## **Podcasts**

- SE Radio 637: Steve Smith on Software Quality
- Netzpodcast
- Developer Voices
- Cloudcast
- DevOps Paradox
- localfirst.fm
- The Engineering Room
- Better Offline
- Thoughtworks Technology Podcast
- Add Dot