

JAX / MAINZ / 06.05.2025

SOFTWARE- ARCHITEKTUR IM ENSEMBLE



JOSHUA TÖPFER
SENIOR CONSULTANT

**"Es braucht nicht
mehr Kommunikation.
Es braucht mehr
Kollaboration!"**

JOSHUA TÖPFER

Senior Consultant bei INNOQ

Joshua arbeitet seit mehr als 4 Jahren Vollzeit im Remote Ensemble. Er ist Maintainer von mob.sh und coacht Teams im Ensemble Programming.



A small, worn, light brown paper book titled "ARCHITEKTUR KONZEPT" lies on a wooden surface. The book is slightly open, showing its pages. The title is written in dark, hand-drawn capital letters. The background is a warm, wooden surface with a visible grain.

Architektur aus dem Elfenbeinturm

Worum geht es heute?

- Warum klassische Architekturarbeit oft nicht funktioniert
- Wie Ensemble Programming Nähe & Verständnis schafft
- Welche Rolle Architekt*innen heute einnehmen können
- Wie Entscheidungen im Team entstehen – kollaborativ
- Warum Dokumentation kein Ziel, sondern ein Werkzeug ist

Wie ich arbeite

- Remote-Team: 5 Devs & 1 PO
- „You Build It You Run It“ Team mit viel Ownership
- Keine dedizierte Architektenrolle
- Architekturarbeit passiert im Ensemble
- Verständnis, Entscheidungen & Code entstehen direkt im Flow



Wie wir gemeinsam Probleme lösen?

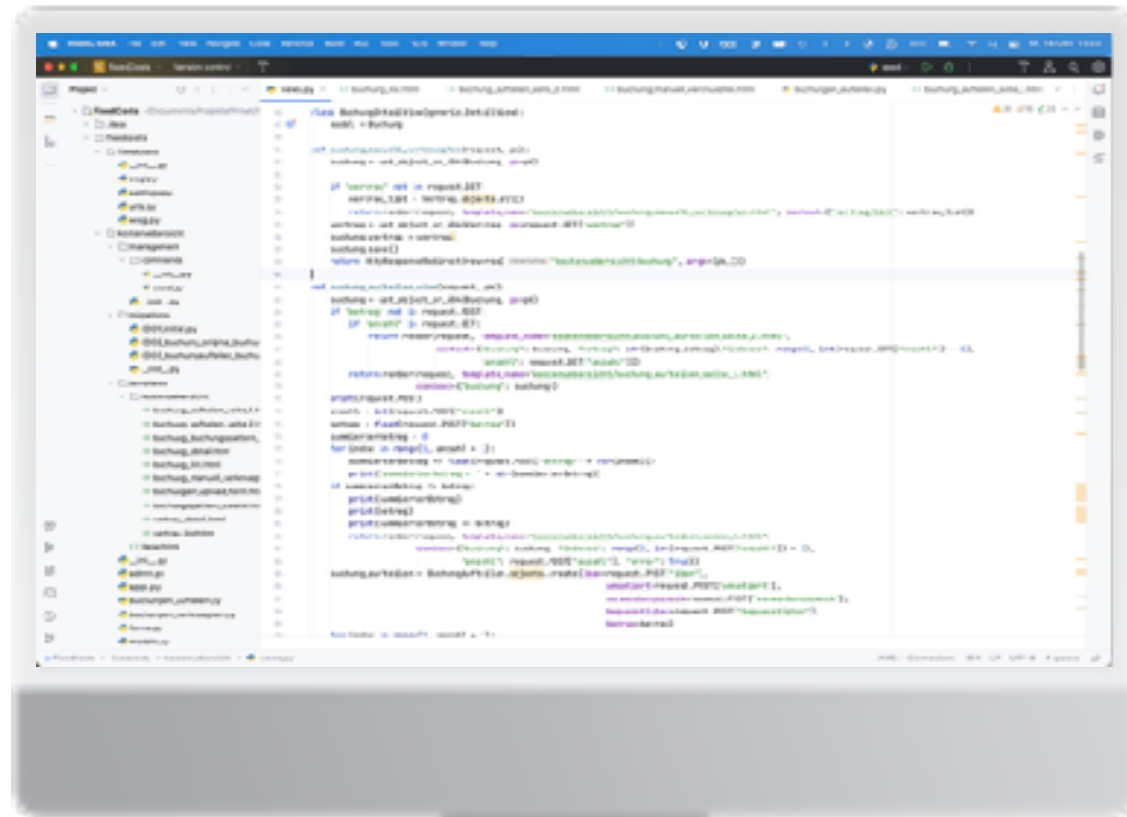
Im Ensemble

Wer hat schon mal Ensemble Programming gemacht?

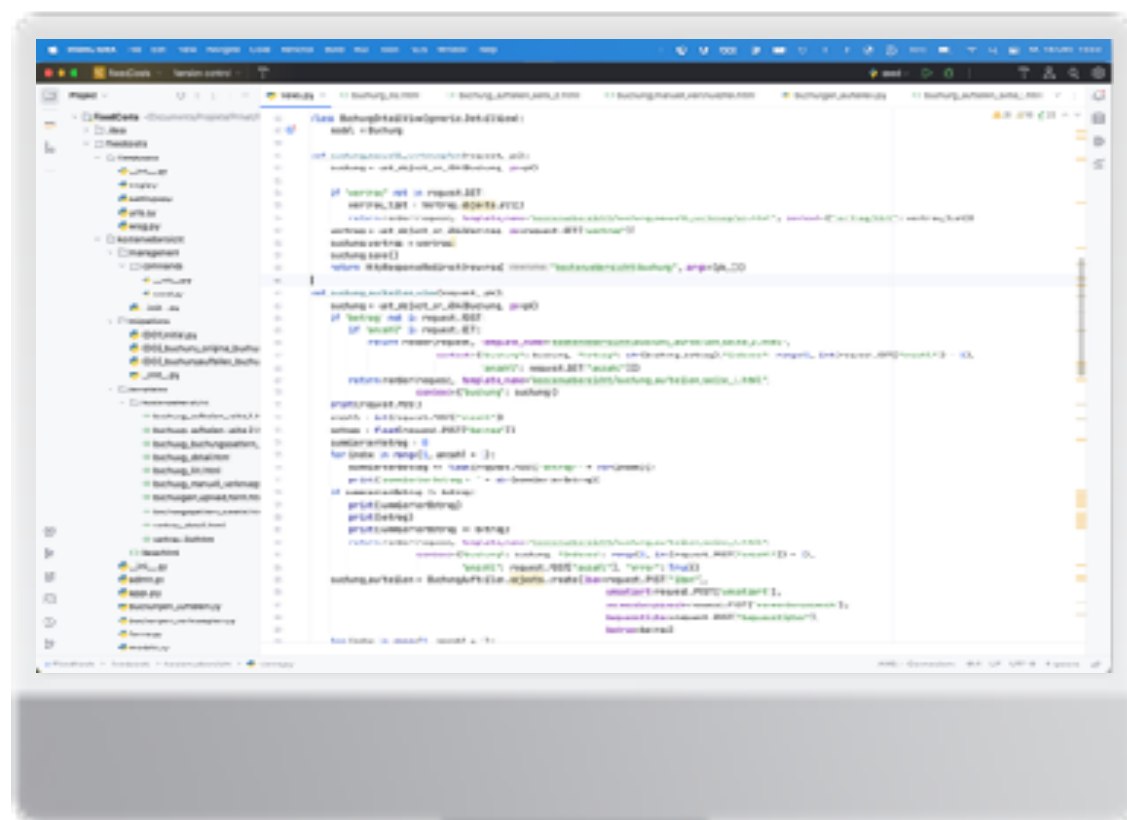


**All the brilliant minds working together
on the same thing, at the same time, in
the same space, and at the same
computer**

Woody Zuill



Typist



Typist



Typist



Typist



Typist



Kollaboration statt Kommunikation

- Vom Wissen aller profitieren
- Gemeinsames Problemverständnis schaffen
- Vermeidung des Sunk Cost Bias
- Starke Ownership & Akzeptanz von Lösungen

Was ist Software Architektur?

~~Software Architektur~~

~~ist die Strukturierung eines
Software Systems~~

Was ist mit Security?

ISO42010



Software Architektur

ist die Menge aller
wichtigen und schwer
änderbaren
Entscheidungen

Martin Fowler



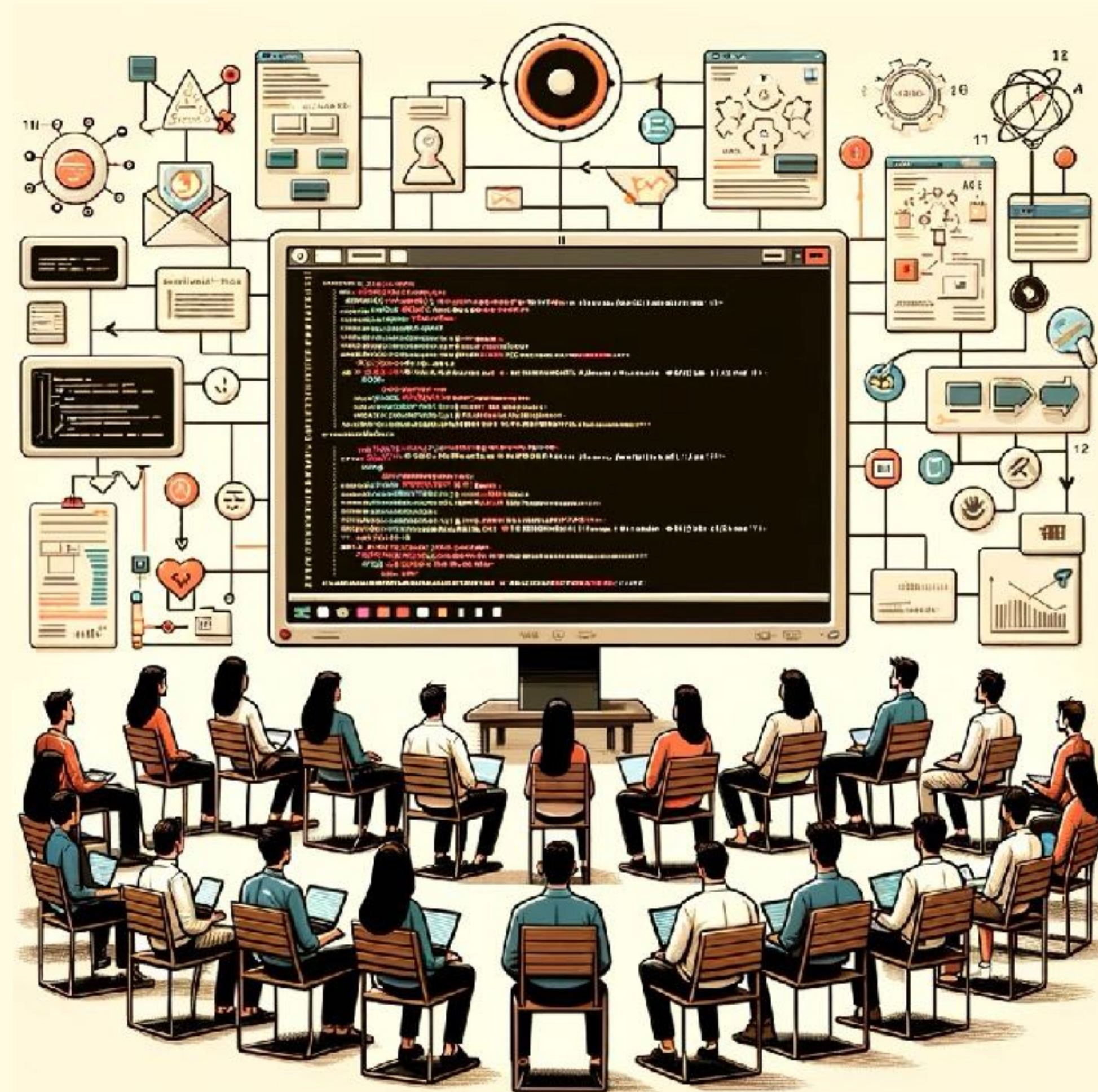
Die Softwarearchitekt*in

Traditionelle Sicht

- Karrierestufe
- Feste Zuordnung zu einer Person
- Verfügt idealerweise über tiefe System- und Technologiekenntnis
- Entscheidungsgewalt



Softwarearchitektur im Ensemble



Die Systemarchitekt*in

Moderne Sicht

- Hat einen Überblick über das gesamte System
- Kollaboriert mit den Teams
- Ist eher Sparringpartner*in oder Moderator*in
- Führt Entscheidungen herbei



... als Enabler*in

- Teams mit wenig Erfahrung benötigen Architekturwissen und -kompetenz
- Mitarbeit im Ensemble (Probleme verstehen & Vertrauen aufbauen)
- Fokus auf die Qualitätskriterien und deren Vermittlung
- Methoden für Entscheidungsprozesse und Dokumentation vermitteln
- Facilitation von Entscheidungsprozessen

... als Sparringpartner*in

- Fokus auf die Qualitätskriterien
- Überblick über andere Teams und mögliche Synergien
- Facilitation von Architekturentscheidungsprozessen
- Zusätzliche Außensicht und strategische Sicht

... verfasst Architekturrichtlinien

- Fokus auf die Makroarchitektur
- Geben Teams einen einfachen Rahmen und minimieren Absprachen
- Es kann Gründe geben von den Richtlinien abzuweichen (Golden Path)
- Richtlinien in enger Kollaboration mit den Team erstellen

Team als Architekt*in

Moderne Sicht

- Ist eine Rolle, die jeder annehmen kann
- Kollaboration statt Kommunikation
- Nah an den Problemen & Fokus auf Mikro Architektur
- Gesamte Expertise des Teams nutzen



Dokumentation

Dokumentation

- Zugänglichkeit und Auffindbarkeit
 - Zentraler Punkt (auch für Stakeholder*innen)
 - Logische Strukturierung (z.B. Arc42)
- Aktualität
- Klarheit und Verständlichkeit
 - Passende Sprache für die Zielgruppe
 - Nutzung von Diagrammen
- Vollständigkeit
 - Besonders wichtig die Architekturentscheidungen



Dokumentation

... als Selbstzweck

Wird nachträglich geschrieben

Eine Person schreibt

Kaum gelesen, oft veraltet

Viel Aufwand, wenig Wirkung

Dieses Bild ist durch KI erzeugt worden.

... als Werkzeug

Entsteht im Gespräch

Team erstellt gemeinsam

Wird genutzt & angepasst

Gerade genug, genau dort,
wo sie gebraucht wird

Dieses Bild ist durch KI erzeugt worden.

Gemeinsam dokumentieren

 **Gleichzeitig sprechen & schreiben**

Was gesagt wird, fließt direkt ins Dokument

 **Konkret statt vage**

Durch Diskussionen wird das Problem, aus verschiedenen Blickwinkeln klarer beschrieben.

 **Texte, die alle verstehen**

Gemeinsames Schreiben sorgt für Klarheit, noch bevor jemand fragt.

 **Nahtloser Übergang zur Umsetzung**

Dokumentation entsteht dort, wo auch Code entsteht

Docs-as-Code

- Pflege der Dokumentation nah am Code
- Dokumentieren ist kein Tool Bruch mehr
- Code und Dokumentation gemeinsam deployen
- docToolchain
 - Nutzt AsciiDoc
 - ReadMe für Devs
 - Export nach Confluence für Stakeholder

Architekturentscheidungen

Entscheidungen explizit machen

- Täglich ca. 20.000 Entscheidungen [1]
- Architekturentscheidungen verstecken sich oft im Detail
- Oft trifft sie jemand nebenbei & merkt es nicht mal
- Diskussionen im Ensemble sind ein Hinweis

Architecture Decision Records

- Dokumentation einer Architektur Entscheidung
- Soll Transparenz und Nachvollziehbarkeit schaffen
- Dient dem Wissenserhalt
- Oft durch eine Person vorgeschlagen dann wird entschieden

ADR

Titel: ...

Kontext: ...

Entscheidung: ...

Begründung: ...

Auswirkungen: ...

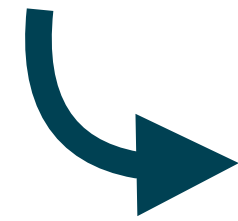
Alternativen: ...

Status: ...

Kollaborativ entscheiden mit ADRs



Problem verstehen



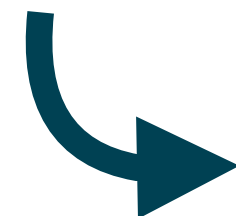
Kontext aufschreiben



Lösungen brainstormen



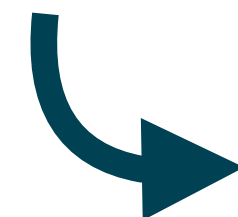
Vor- und Nachteile



Entscheidung treffen



Auswirkung dokumentieren



Direkt umsetzen

ADR

Titel: ...

Kontext: ...

Entscheidung: ...

Begründung: ...

Auswirkungen: ...

Alternativen: ...

Status: ...

Konsens

Vollständige Übereinstimmung

Zeitaufwendig

Kompromisse verwässern
Entscheidungen

Risiko das keine Entscheidung
getroffen wird

Konsent

Mittragen und kein Veto

Schnelle Entscheidungsfindung

Ermutigt neue Ideen auszuprobieren

Jeder muss ernsthafte Bedenken
äußern und Lösung mittragen können

So spät wie möglich entscheiden

- Entscheidungen reifen mit dem Problemverständnis
- Entscheidungen treffen wenn sie nötig sind
 - Direkt vor der Umsetzung
 - Nicht vorausgeplant
 - Keine Blockaden
- Schmerz muss spürbar sein
 - Fördert pragmatische Architekturen
 - Kann aber auch in technische Schulden umschlagen

Fazit

Als Systemarchitekt*in ist Nähe zu den Teams wichtig, um ...

- die Probleme der Teams besser zu verstehen
- Vertrauen zu den Teams aufzubauen
- als Sparringpartner*in wahrgenommen zu werden
- Architekturwissen zu vermitteln

Als Team ist die Kollaboration wichtig, um ...

- Architekturentscheidungen explizit zu machen
- vom Wissen aller zu profitieren
- ein gemeinsames Bild des Problems zu erhalten

Ressourcen

- Software Teaming: A Mob Programming, Whole-Team Approach von Woody Zuill & Kevin Meadows
- Code with the Wisdom of the Crowd: Get Better Together with Mob Programming von Mark Pearl
- remotemobprogramming.org

Unsere Primer



www.innoq.com/primer

Kostenlos herunterladen



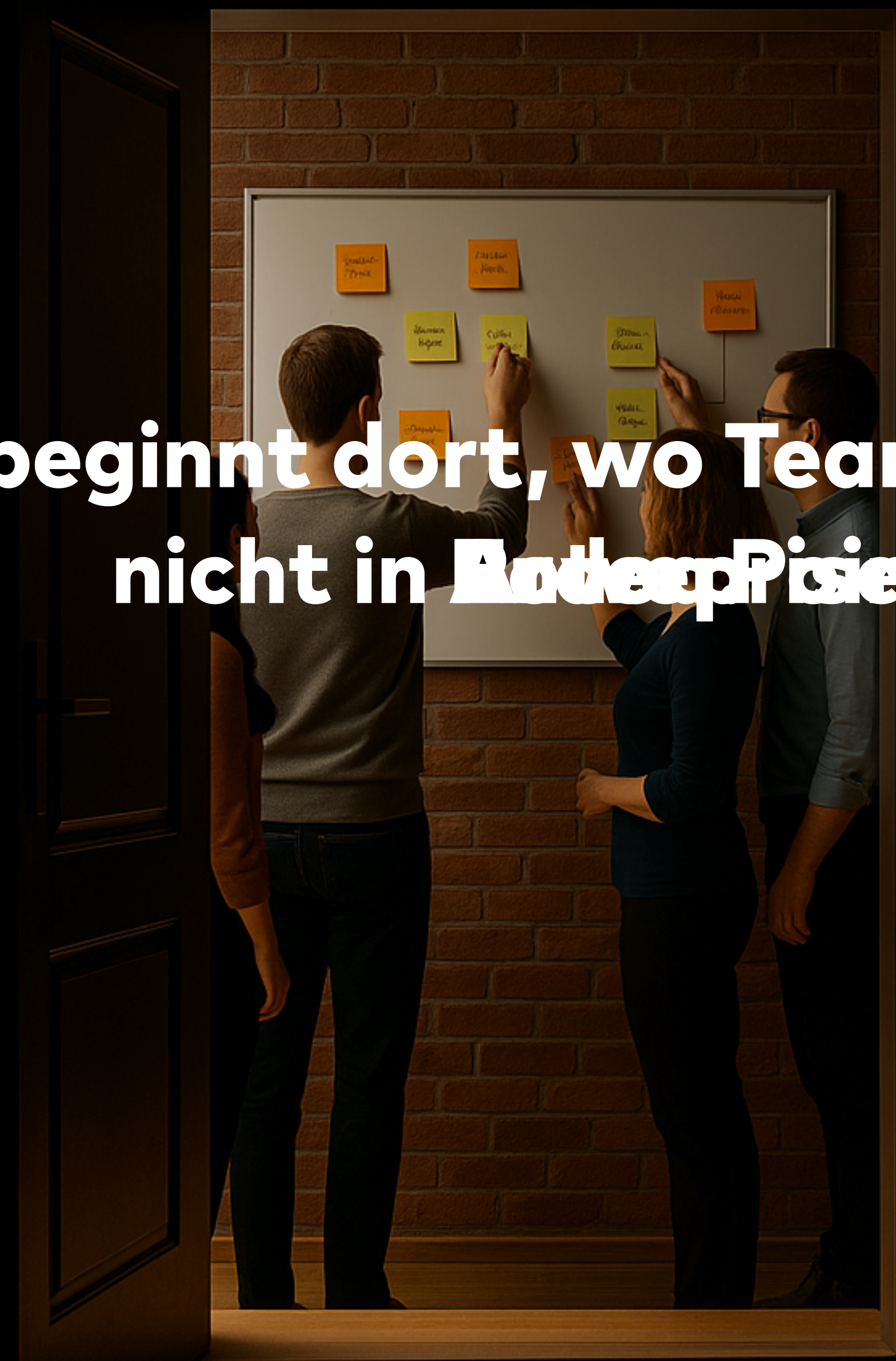
Unsere Primer



www.innoq.com/primer
Kostenlos herunterladen



**Architekturarbeit beginnt dort, wo Teams Probleme verstehen,
nicht in AutoCAD.**



SHOWCASE

KI-Agenten souverän integrieren – Europäische Unternehmens- systeme anbinden mit MCP

Mittwoch, 7. Mai, am INNOQ Stand



Robert Glaser
Head of Data & AI

INNOQ

LIGHTNING TALK

Backend- und UI-Plugins im Backstage Software Catalog

Donnerstag, 8. Mai, 14:45 Uhr,
am INNOQ Stand



Stefan Negele
Senior Consultant

INNOQ

Danke! Fragen?



Joshua Töpfer
joshua.toepfer@innoq.com
+49 172 3666344



innoQ Deutschland GmbH

Krischerstr. 100
40789 Monheim
+49 2173 3366-0

Ohlauer Str. 43
10999 Berlin

Ludwigstr. 180E
63067 Offenbach

Kreuzstr. 16
80331 München

Wendenstraße 130
20537 Hamburg

Spichernstraße 44
50672 Köln