Collaborative Modeling

for Better Architecture Decisions



Stefan Priebsch, thePHP.cc

Software Architecture Gathering



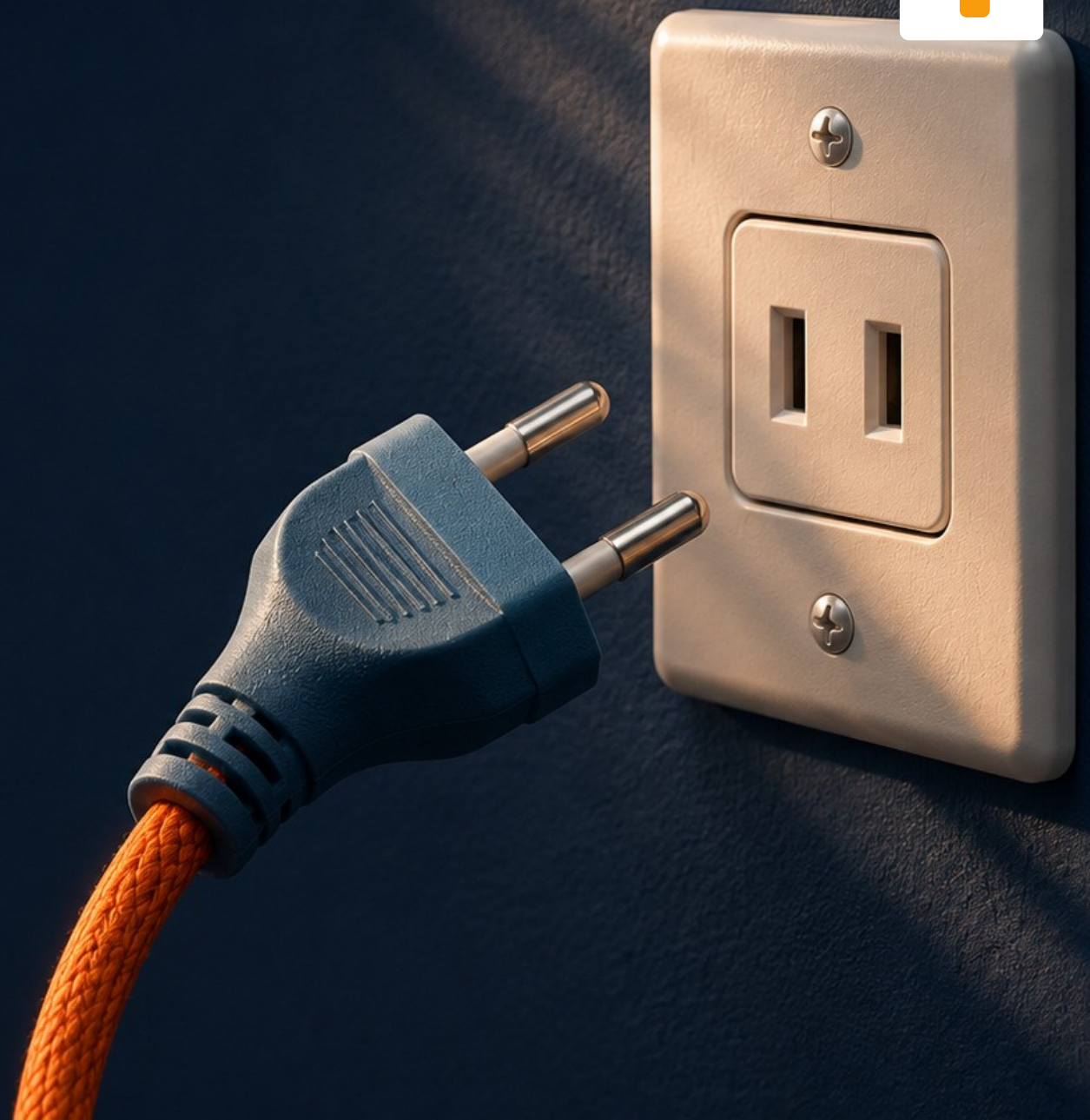
THE WALK TO THE STATION

Your front door.
My directions.



Misunderstanding often feels like understanding.

There is power in the room.



It fits.
Does it work?



There’s electricity in the room

Location	Plug types	Voltage	Frequency
Germany	C / F	230 V	50 Hz
United States	A / B	120 V	60 Hz
Japan	A / B	100 V	50 / 60 Hz

Interface worked.
Understanding didn’t.



thePHP.cc

Mars Climate Orbiter

“We sent you the thrust data.”

GROUND SOFTWARE SENT

Pound-Force Seconds

NAVIGATION EXPECTED

Newton Seconds

The implementation mismatch escaped the checks.

LAUFENBURG BRIDGE

Same unit.
Different reference.



Laufenburg bridge

“We both measure height in metres.”

The same unit does not establish the same reference.

Laufenburg bridge

“We both measure height in metres.”

27 cm

Known difference between
height reference systems

The same unit does not establish the same reference.

Laufenburg bridge

“We both measure height in metres.”

27 cm

54 cm

Known difference between
height reference systems

The same unit does not establish the same reference.

Laufenburg bridge

“We both measure height in metres.”

27 cm

Known difference between height reference systems

54 cm

Result after correcting with the wrong sign

The same unit does not establish the same reference.



You force women to wear clothes?

Ferengi perspective in Star Trek Next Generation

“We should centralize identity.”

A university. A proposal. Everyone agrees.



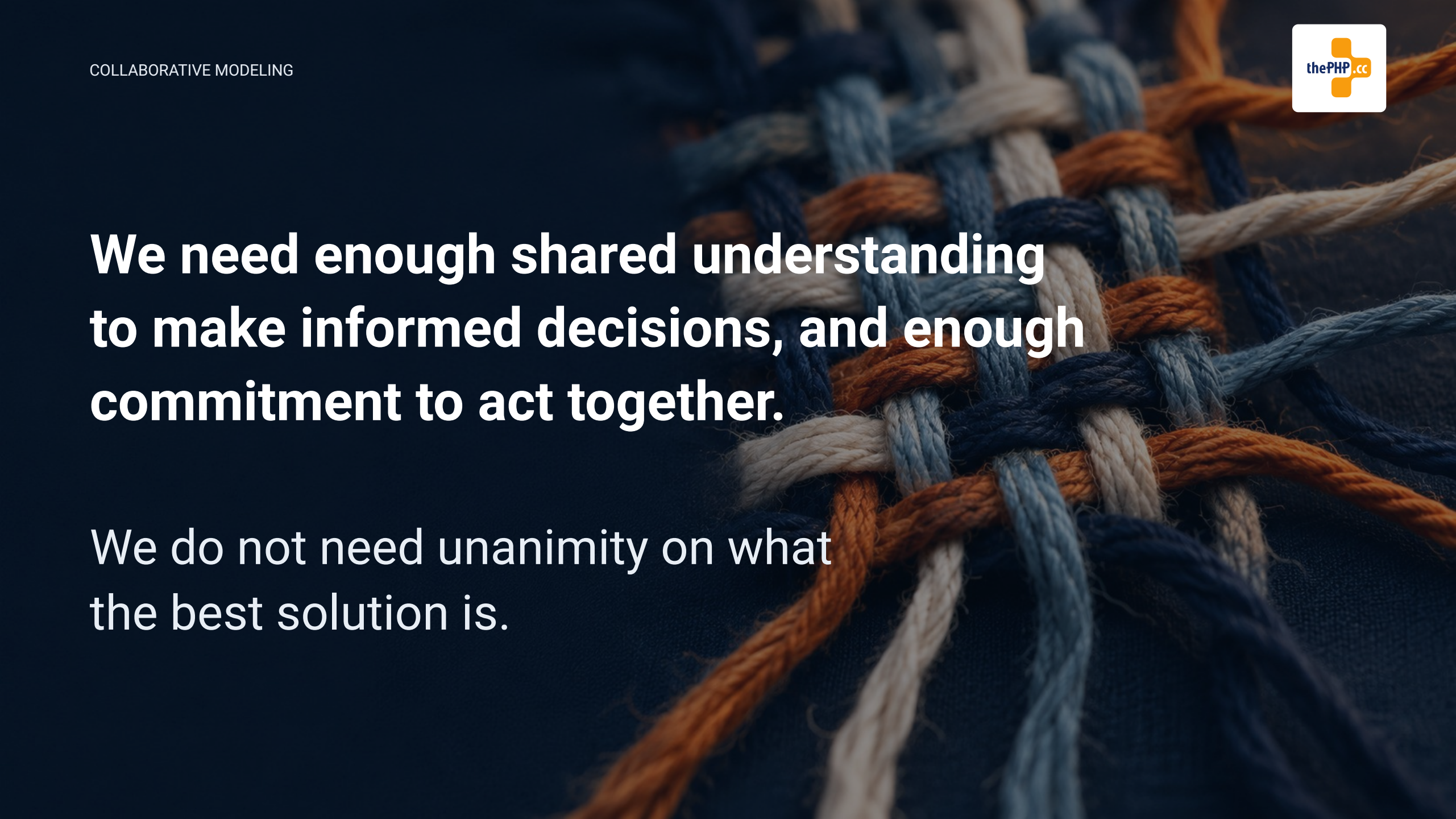
One sentence. Five interpretations.

Students	One login across services
Administration	One authoritative person record
Research groups	Access for external collaborators
Operations	One identity service to run
Security	Consistent access revocation



Agreement ≠ understanding ≠ alignment

Agreement	We endorse the proposal.
Understanding	We can explain its consequences.
Alignment	Our commitments fit together.



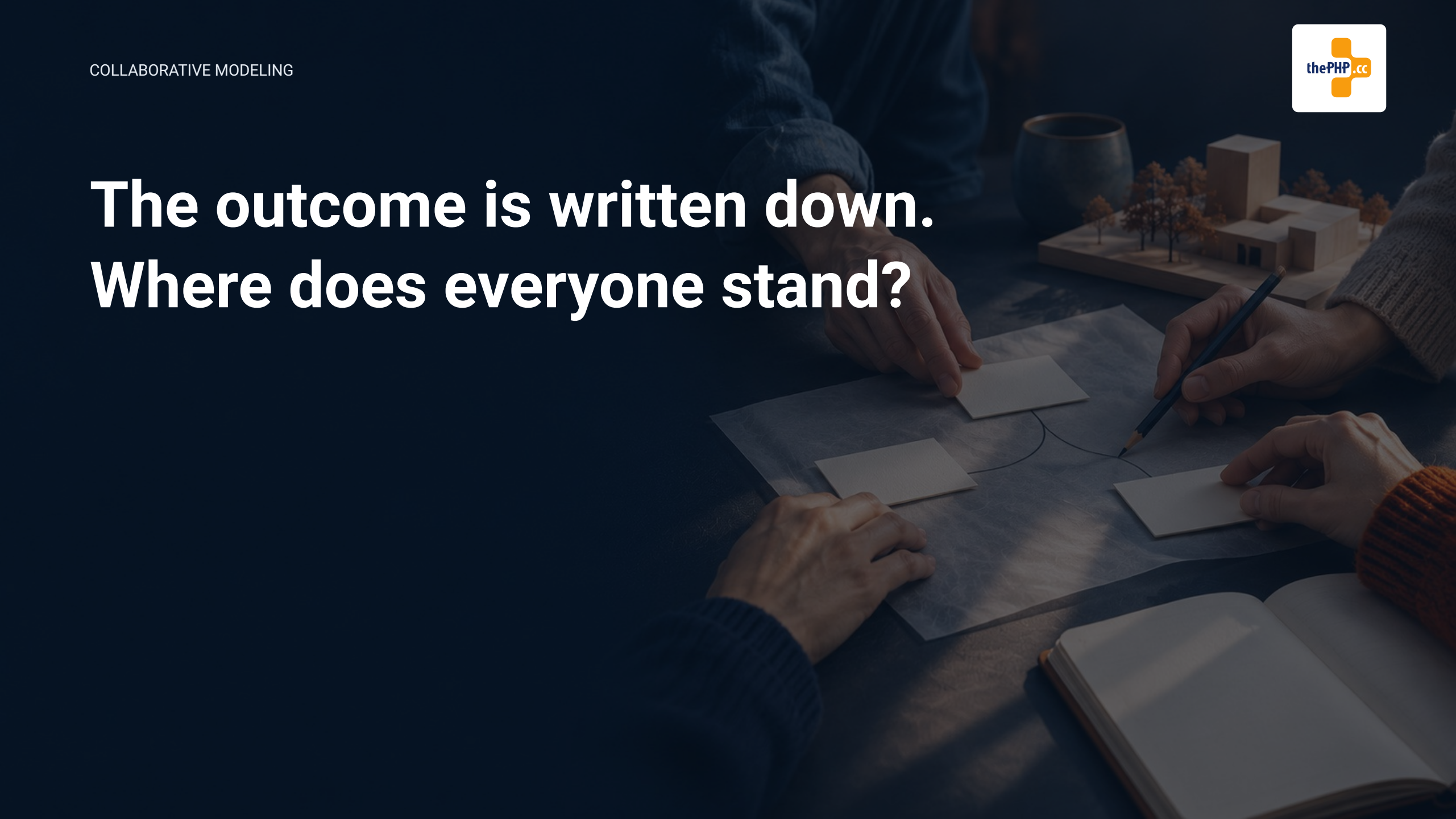
We need enough shared understanding to make informed decisions, and enough commitment to act together.

We do not need unanimity on what the best solution is.

Agreement is not evidence of shared understanding.



The outcome is written down. Where does everyone stand?

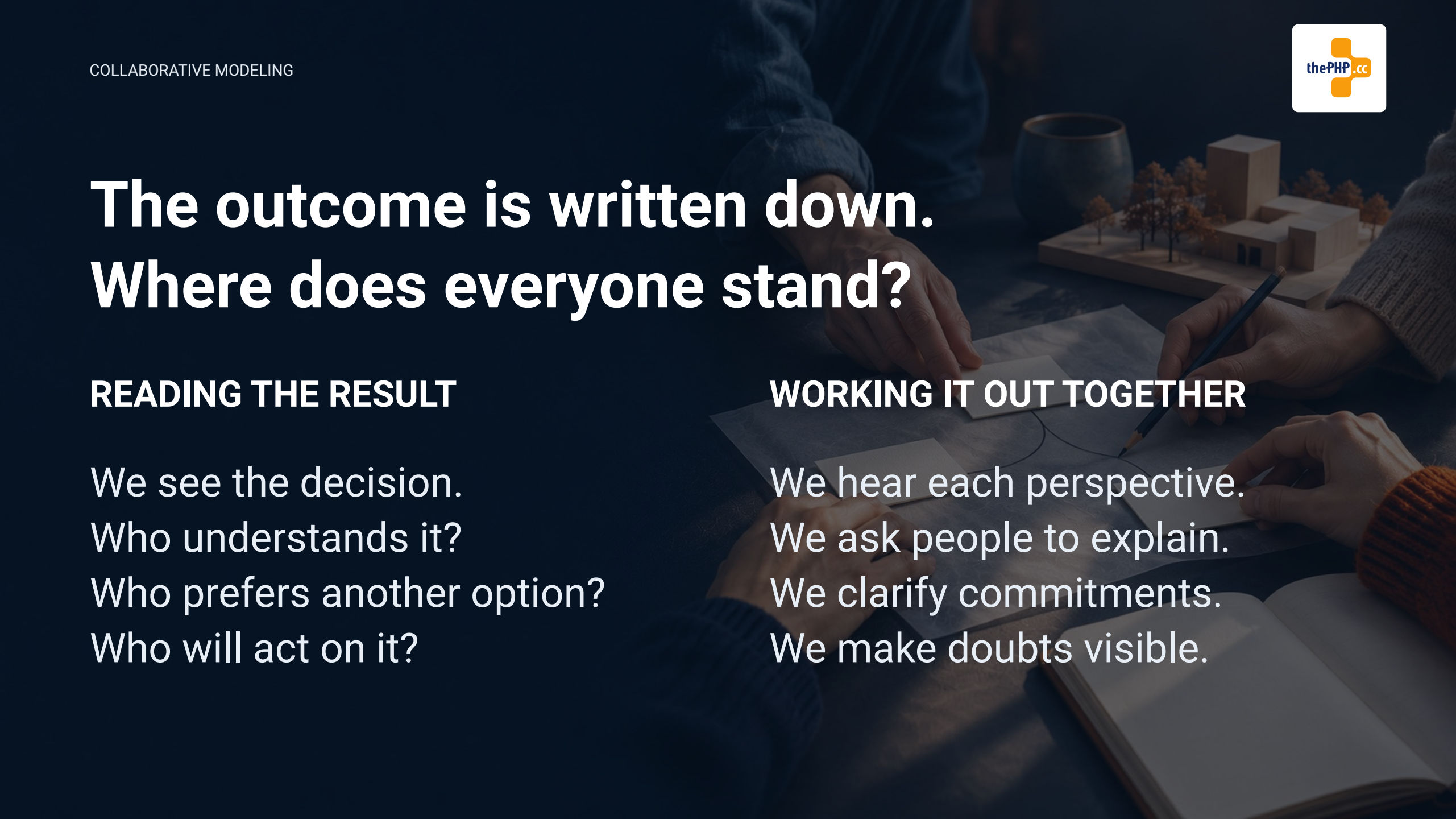


The outcome is written down. Where does everyone stand?

READING THE RESULT

We see the decision.
Who understands it?
Who prefers another option?
Who will act on it?





The outcome is written down. Where does everyone stand?

READING THE RESULT

We see the decision.
Who understands it?
Who prefers another option?
Who will act on it?

WORKING IT OUT TOGETHER

We hear each perspective.
We ask people to explain.
We clarify commitments.
We make doubts visible.

Understand together.
Then write it down.



When ambiguity becomes software

In the room

Question the assumption.
Change the decision.

In software

Find where it became behavior.
Change code and dependencies.
Verify the resulting system.

Even with agents, changing software costs more
than correcting a decision together before implementation.



Why apparent consensus happens

Premature convergence

We choose a solution before clarifying the problem and alternatives.

Why apparent consensus happens

Premature convergence

We choose a solution before clarifying the problem and alternatives.

False consensus

We use a term and assume we mean the same thing.

Why apparent consensus happens

Premature convergence

We choose a solution before clarifying the problem and alternatives.

False consensus

We use a term and assume we mean the same thing.

Authority bias

A senior voice proposes it.
Others keep their doubts to themselves.

thePHP.cc

Criteria for a good decision

Outcome	For whom, and by when?
Constraints	What must remain true?
Trade-offs	Which cost and risk do we accept?
Future change	What must remain feasible?
Evidence	What would make us reconsider?

Making sense of things. Together. Visibly.



EventStorming

Domain Storytelling

Architectural Roleplay

Example Mapping

User Story Mapping

Quality Attribute Scenarios

Event Modeling

Impact Mapping

Context Mapping

Bounded Context Canvas

Value Stream Mapping

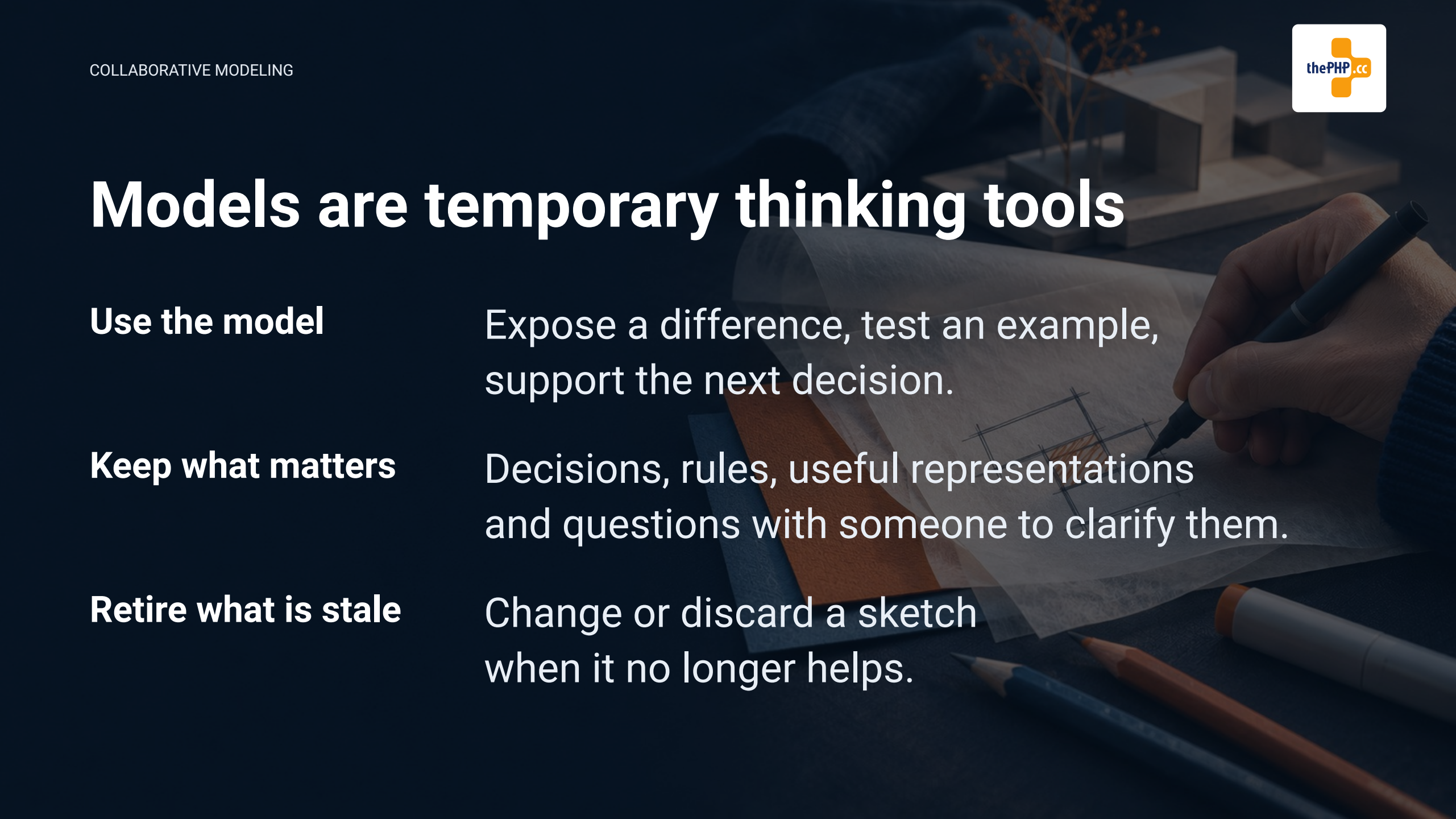
Customer Journey Mapping

Business Model Canvas

Wardley Mapping

CRC Cards

LEGO Serious Play



Models are temporary thinking tools

Use the model

Expose a difference, test an example, support the next decision.

Keep what matters

Decisions, rules, useful representations and questions with someone to clarify them.

Retire what is stale

Change or discard a sketch when it no longer helps.

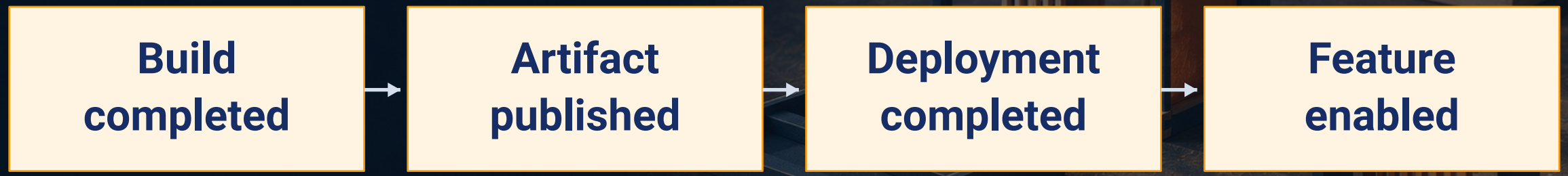
EventStorming

“The release is finished.”



EventStorming

“The release is finished.”







Domain Storytelling

“The sample has arrived.”

Field researcher	Collects the sample and its location
Reception	Records physical arrival
Analyst	Checks whether it can be analyzed

Who connects the sample to the collection site?


AUCTIONING A BICYCLE (ONLINE) - HAPPY PATH, PURE
SELLER
SETS
WITH
SHARE
REVENUE
BIDDER 1
BIDDER 2
ASSUMPTIONS
- BIKE IS CLEANED AND WORKS FINE
- BID INKREMENT
ANNOTATIONS
NO HIGHER BID
Stefan Priebsch



Architectural Roleplay

The quickest way of making domain discoveries

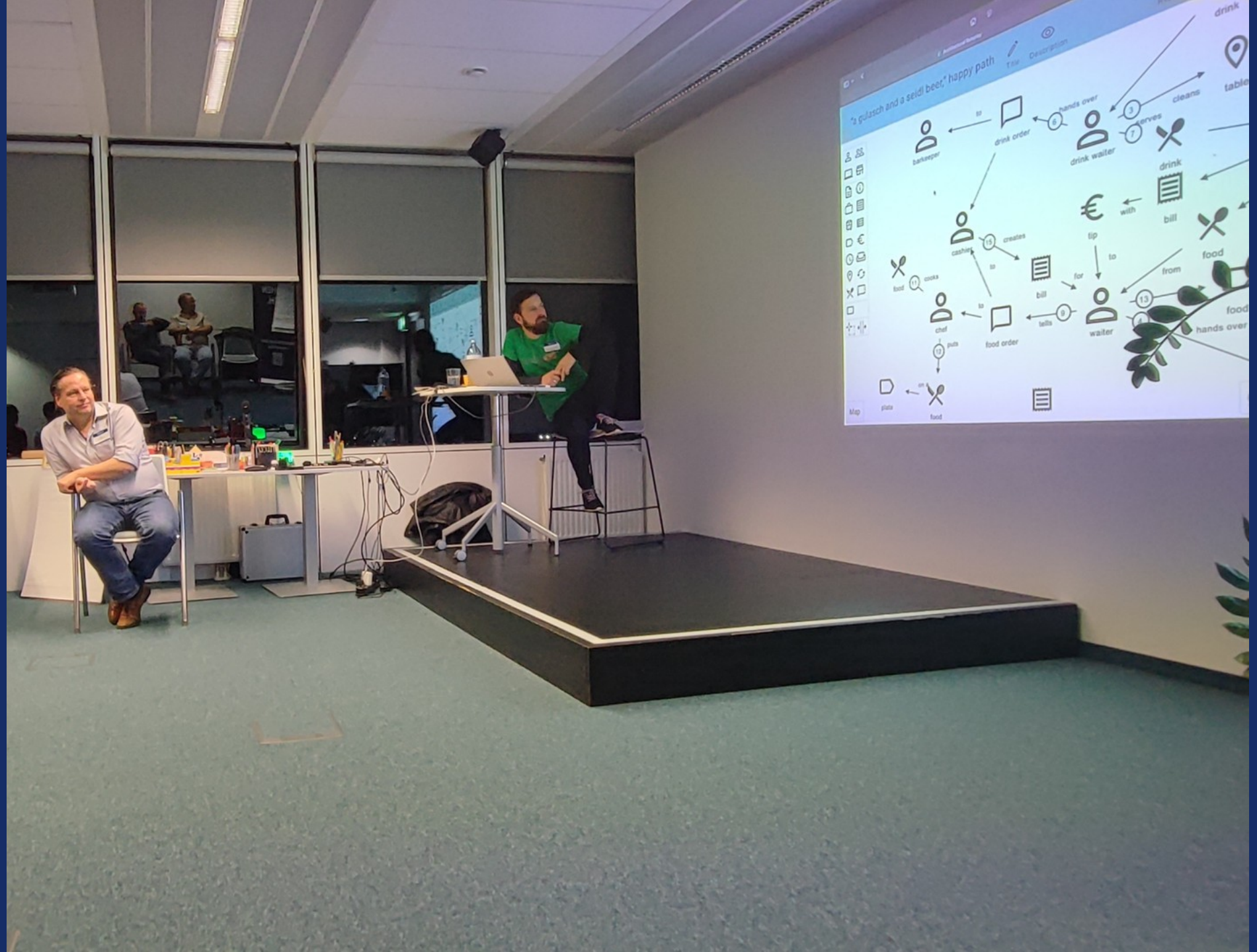
Enact the exchange. Capture what it reveals.

COLLABORATIVE MODELING



Travel expenses reimbursement

Live exercise · Architectural Roleplay


"a gulasch and a seidl beer," happy path
Title
Description
drink
table
cleans
serves
hands over
drink order
drink waiter
barkeeper
drink
bill
with
tip
cashier
creates
food
from
bill
for
waiter
food order
tells
chef
cooks
food
hands over
plate
food
Map

thePHP.cc

Architectural Roleplay



Explore the method

architectural-roleplay.com

The question determines the method

THE QUESTION

A USEFUL START

The question determines the method

THE QUESTION

One concrete case?

A USEFUL START

Domain Storytelling

Actors, activities and exchanged information

The question determines the method

THE QUESTION

One concrete case?

Process variants?

A USEFUL START

Domain Storytelling

Actors, activities and exchanged information

EventStorming

Events, dependencies and hotspots

The question determines the method

THE QUESTION	A USEFUL START
One concrete case?	**Domain Storytelling** Actors, activities and exchanged information
Process variants?	**EventStorming** Events, dependencies and hotspots
Quick discovery?	**Architectural Roleplay** An enacted scenario and its gaps

Another question, another method

THE QUESTION

A USEFUL START

Another question, another method

THE QUESTION

First useful delivery?

A USEFUL START

User Story Mapping

A complete useful increment

Another question, another method

THE QUESTION

First useful delivery?

Meaning of a rule?

A USEFUL START

User Story Mapping

A complete useful increment

Example Mapping

Rules, examples and open questions

Another question, another method

THE QUESTION

First useful delivery?

Meaning of a rule?

Your questions?

A USEFUL START

User Story Mapping

A complete useful increment

Example Mapping

Rules, examples and open questions

Your methods

It doesn’t need a name. **It needs to be useful!**

What these approaches share



What these approaches share

Concrete A case people can challenge →



What these approaches share

Concrete A case people can challenge

Collaborative Relevant knowledge in the conversation



What these approaches share

Concrete	A case people can challenge
Collaborative	Relevant knowledge in the conversation
Visible	Claims people can point to and change



What these approaches share

Concrete	A case people can challenge
Collaborative	Relevant knowledge in the conversation
Visible	Claims people can point to and change
Exploratory	Differences and open questions



What these approaches share

Concrete	A case people can challenge
Collaborative	Relevant knowledge in the conversation
Visible	Claims people can point to and change
Exploratory	Differences and open questions
Purposeful	Enough support for the next decision



I call myself method-fluid.

Select. Switch. Compose.



When the notation does not help, invent one.

Agree a legend. Check the meaning.





A small gap to software

A temporary access token expires at 18:00.

What happens at exactly 18:00?

Given	Expiry at 18:00
When	The clock reaches 18:00
Then	Reject the request



Current LLMs understand these formats.

They can interpret the graphical representations.

The architect’s job is to create shared understanding

Bring perspectives together

Relevant domain knowledge, technical judgment and risk assessment.

Make decisions explicit

Clarify assumptions, consequences and accepted trade-offs.

Share responsibility

People contribute knowledge and own commitments within their authority.



Don’t ask whether everyone agrees.

Find out whether everyone understands.

thePHP.cc

CoMoCamp 2027

The Collaborative Modeling Unconference



3–6 May 2027

Europahaus · Vienna

Bring your questions. Model together.

comocamp.org

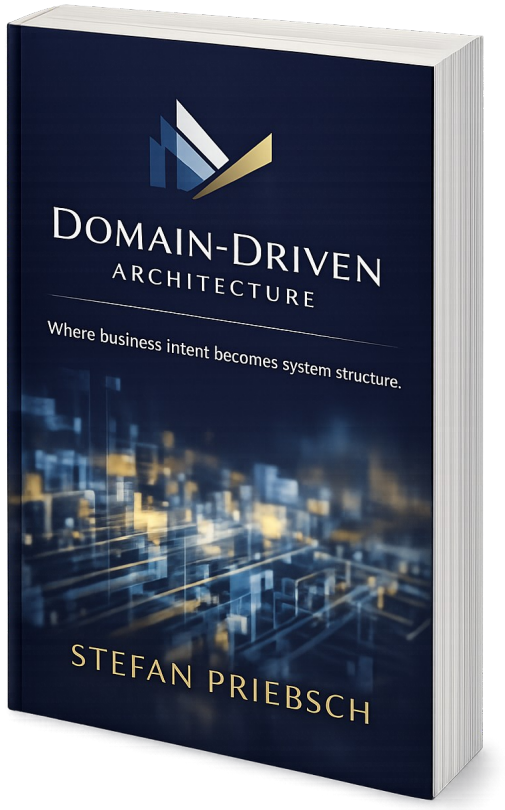
Stefan Priebsch

I’m Stefan, a Software Success Consultant.

I help organizations turn software investments into business results.

That starts with getting people to understand each other, because agreeing on a solution doesn’t mean we agree on the problem.





Domain-Driven Architecture

Follow the book

https://domain-driven-architecture.com