



# EVENTSTORMING

---

## COLLABORATIVE LEARNING FOR COMPLEX DOMAINS

Paul Rayner  
 @thepaulrayner

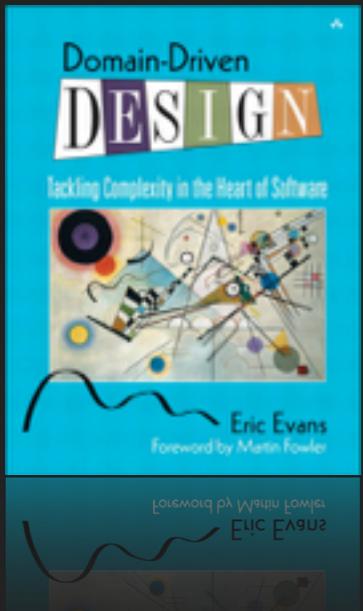


## THE LANDSCAPE

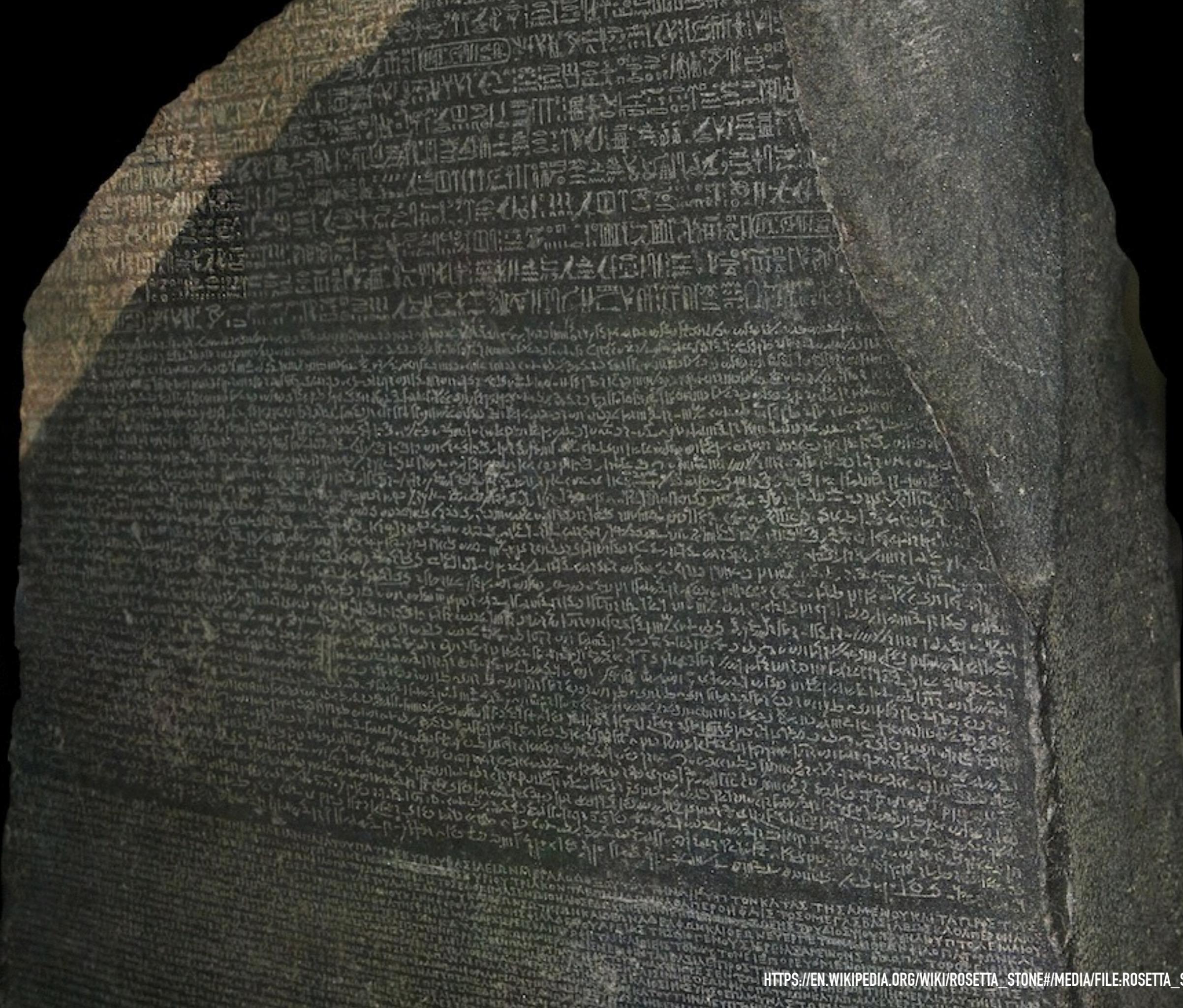
---

- ✓ Complex business rules & processes
- ✓ Knowledge silos
- ✓ Numerous diverse (& often legacy) system interactions

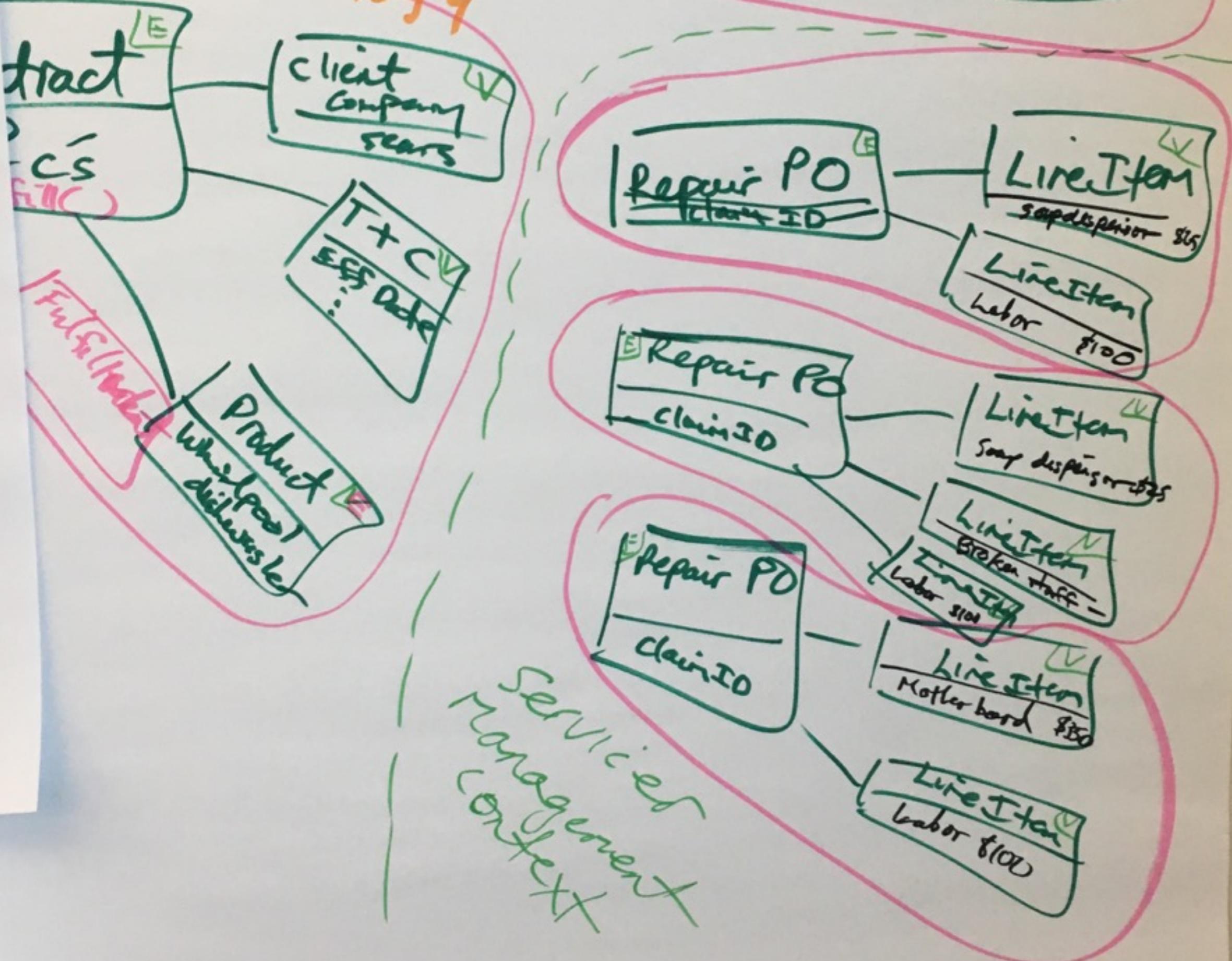
THE CRITICAL COMPLEXITY OF MOST  
SOFTWARE PROJECTS IS IN  
UNDERSTANDING THE DOMAIN  
ITSELF

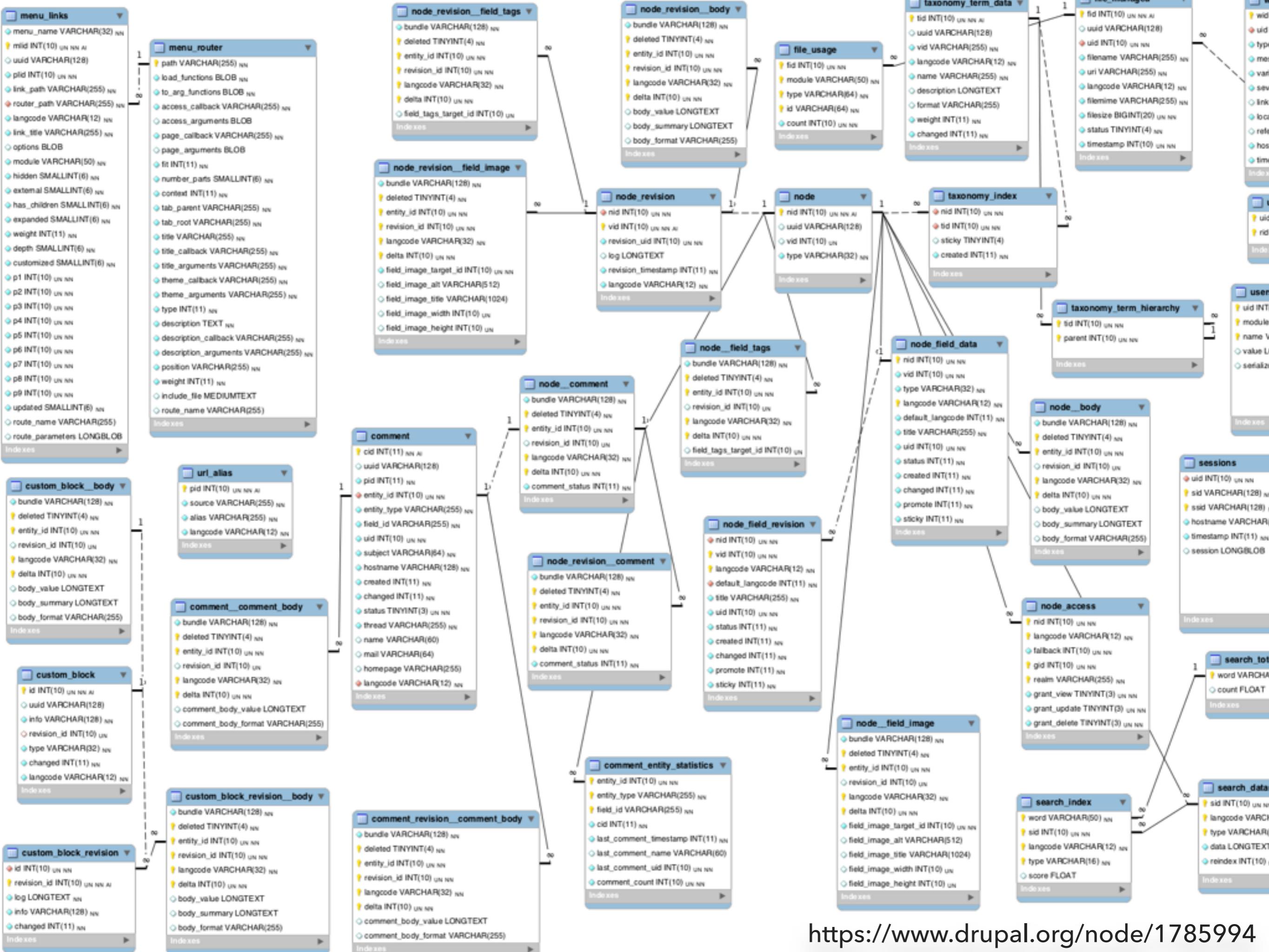


Eric Evans

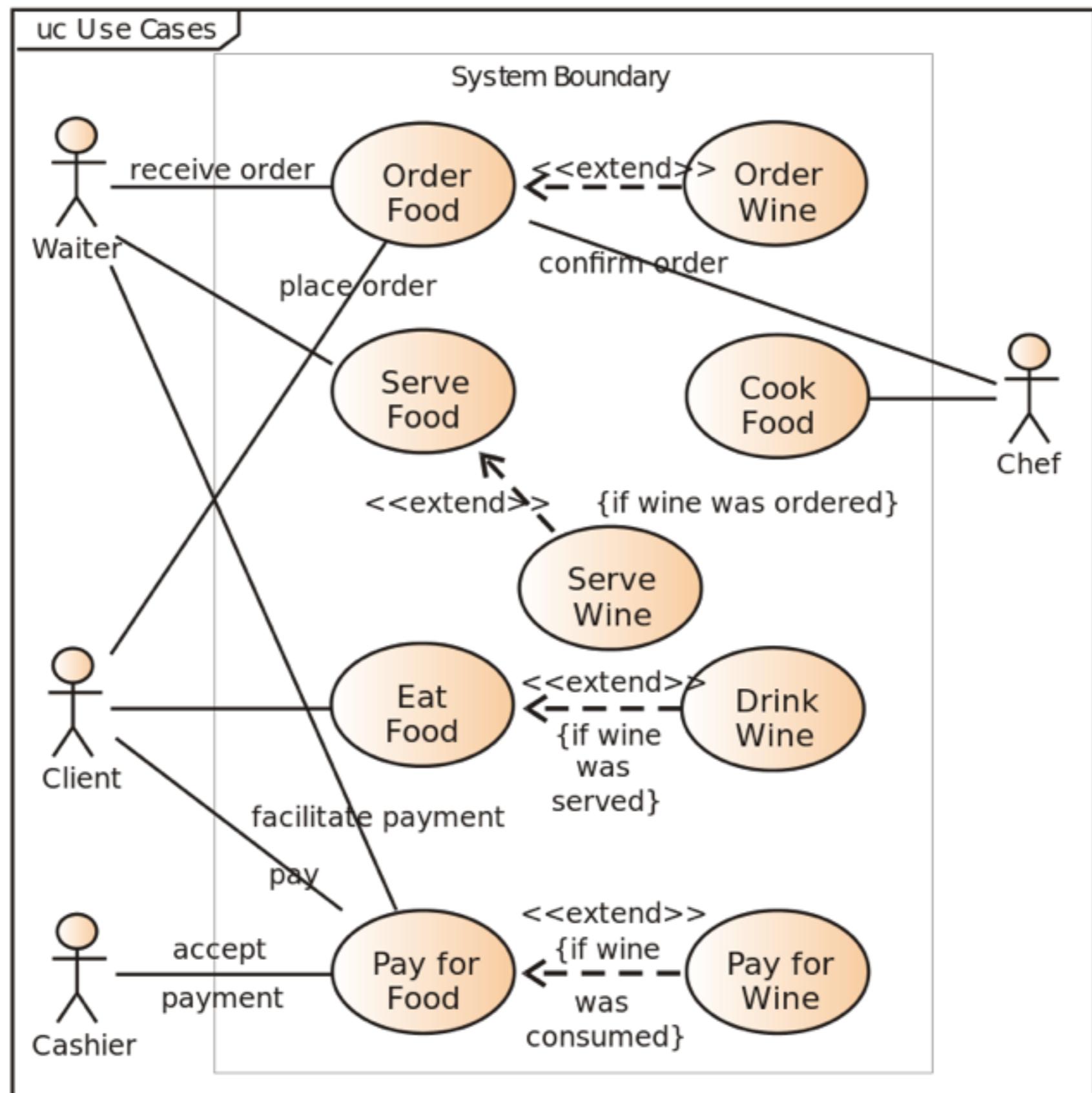


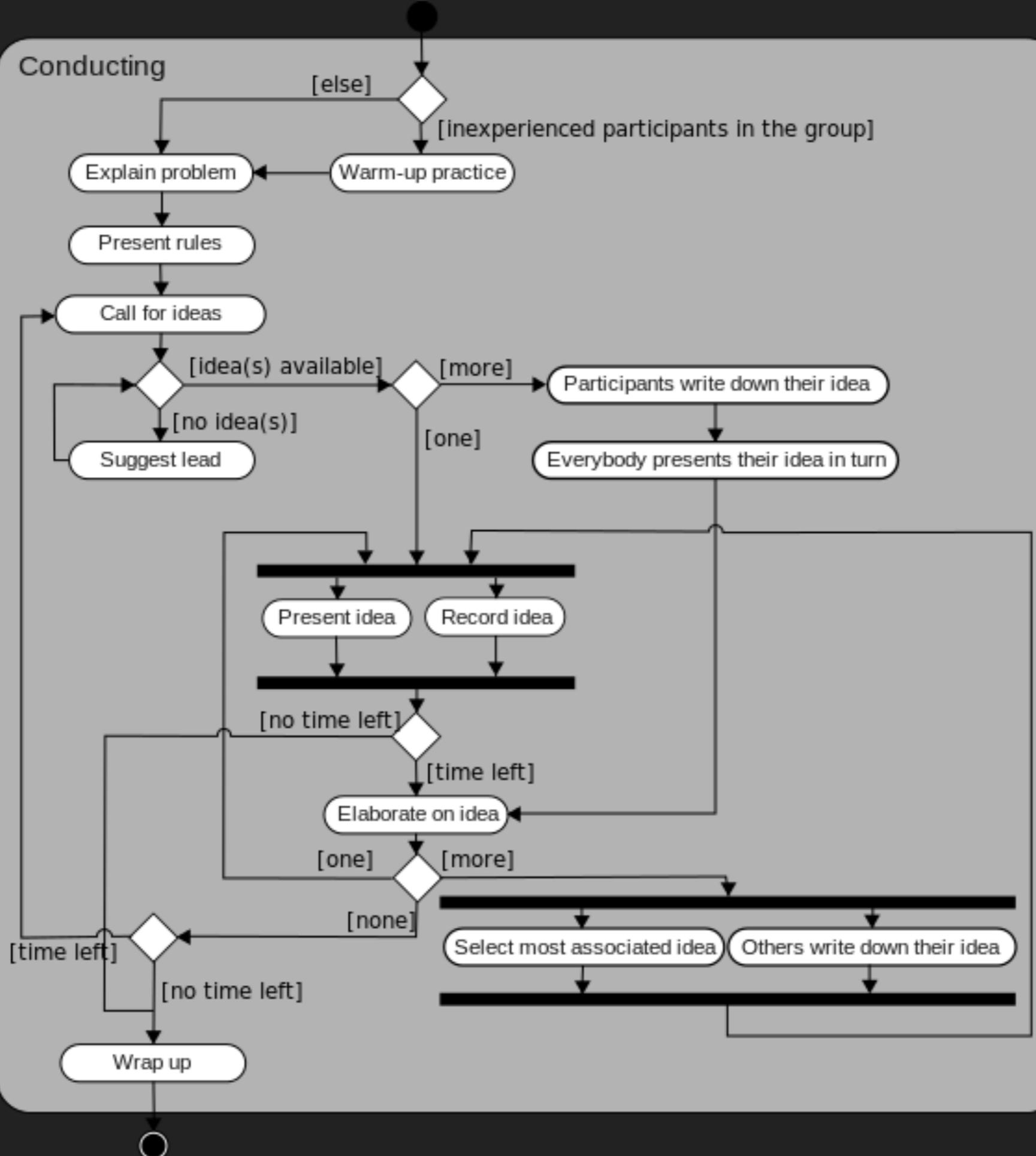
**HOW DO WE LEARN  
...FAST?**

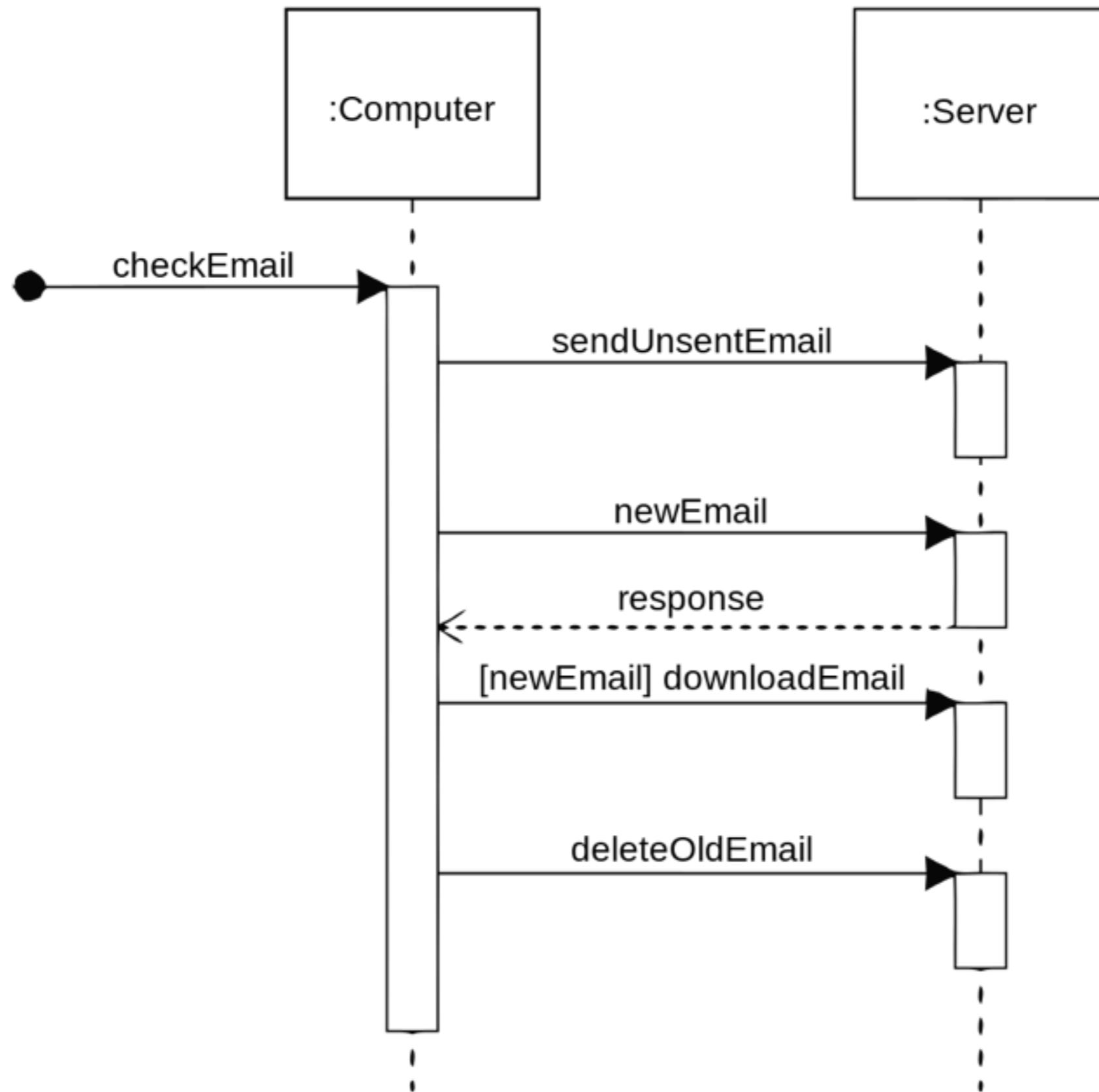




**BEHAVIOUR TRUMPS  
STRUCTURE**







**OPTIMIZE FOR  
SHARED LEARNING**

# EVENT STORMING

Alberto Brandolini



**PLACE EVENTS ON  
A TIMELINE**

ITEM  
ADDED TO  
CART

THIS IS A **DOMAIN EVENT**

- **ORANGE** STICKY NOTE
- VERB AT **PAST TENSE**
- RELEVANT FOR DOMAIN EXPERTS

## WHAT IS A DOMAIN EVENT?

---

Something that happened  
in the business  
that your domain experts  
care about



# SPECIES OF DOMAIN EVENTS

---

- **A step in a business process**

e.g. *customer order submitted, customer paid \$100, order completed, item shipped.*

- **Something that happens on a scheduled basis**

e.g. *nightly account reconciliation completed, end of month.*

- **Something meaningful that occurs as a result of something else happening**

e.g. *customer account locked after invalid password entered 3 times*

**VISUALIZE  
LEARNING  
OPPORTUNITIES**

WHAT IS  
THE  
PRICING  
STRATEGY?

## CAPTURE

- QUESTIONS
- RISKS / WARNINGS
- ASSUMPTIONS
- CONVERSATION POINTS

END OF  
MONTH

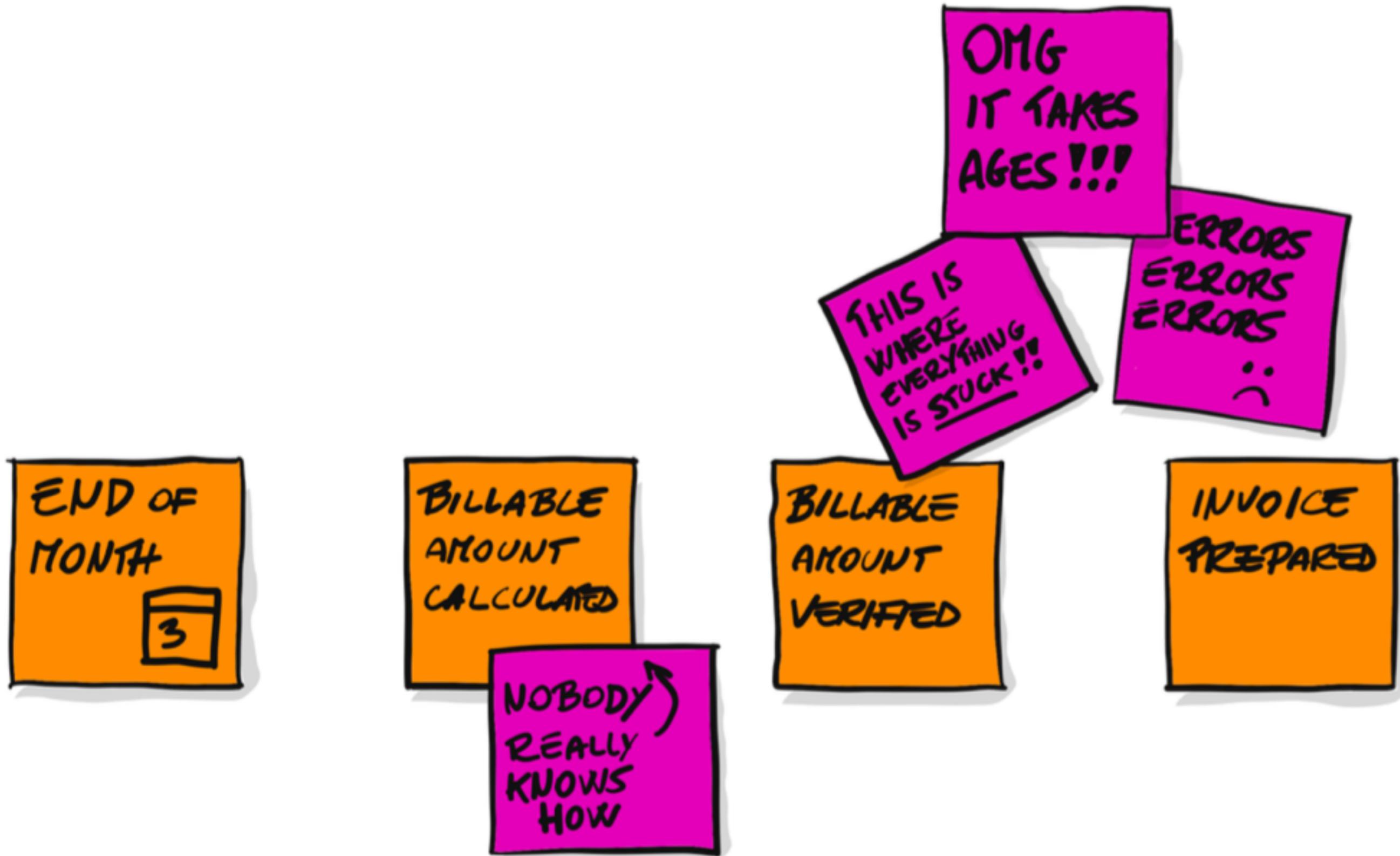
3

BILLABLE  
AMOUNT  
CALCULATED

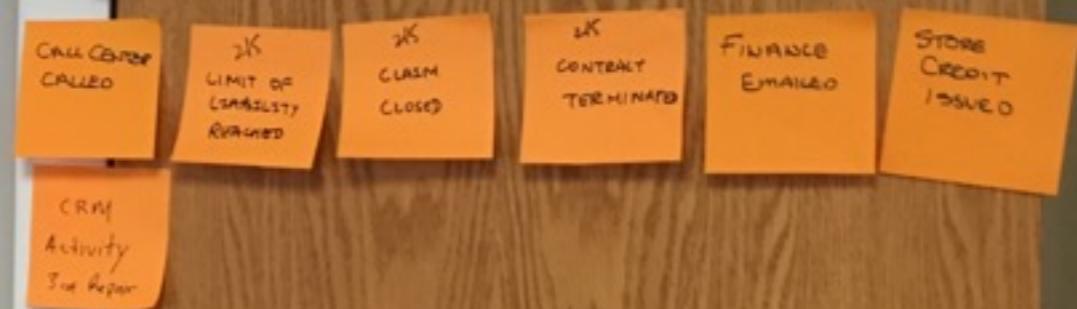
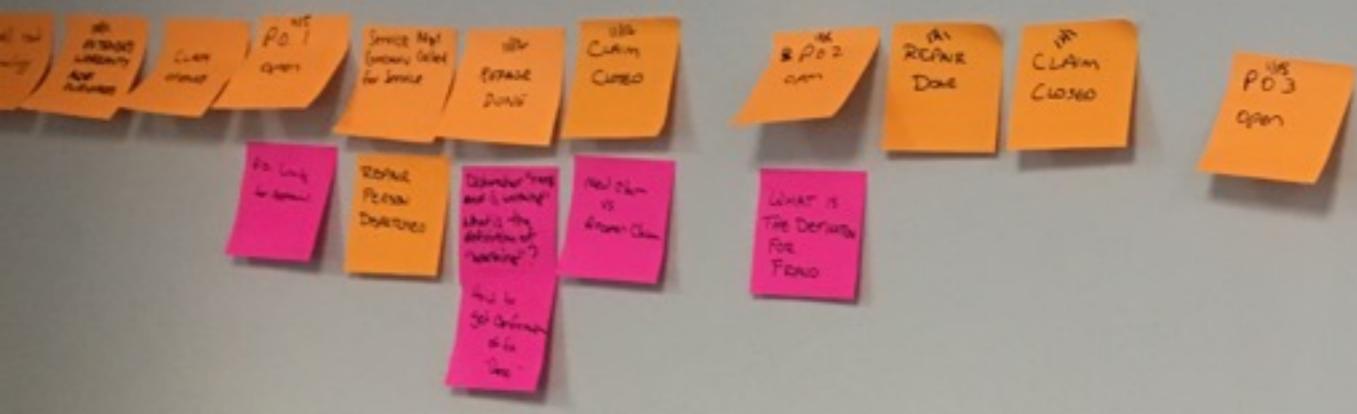
NOBODY  
REALLY  
KNOWS  
HOW

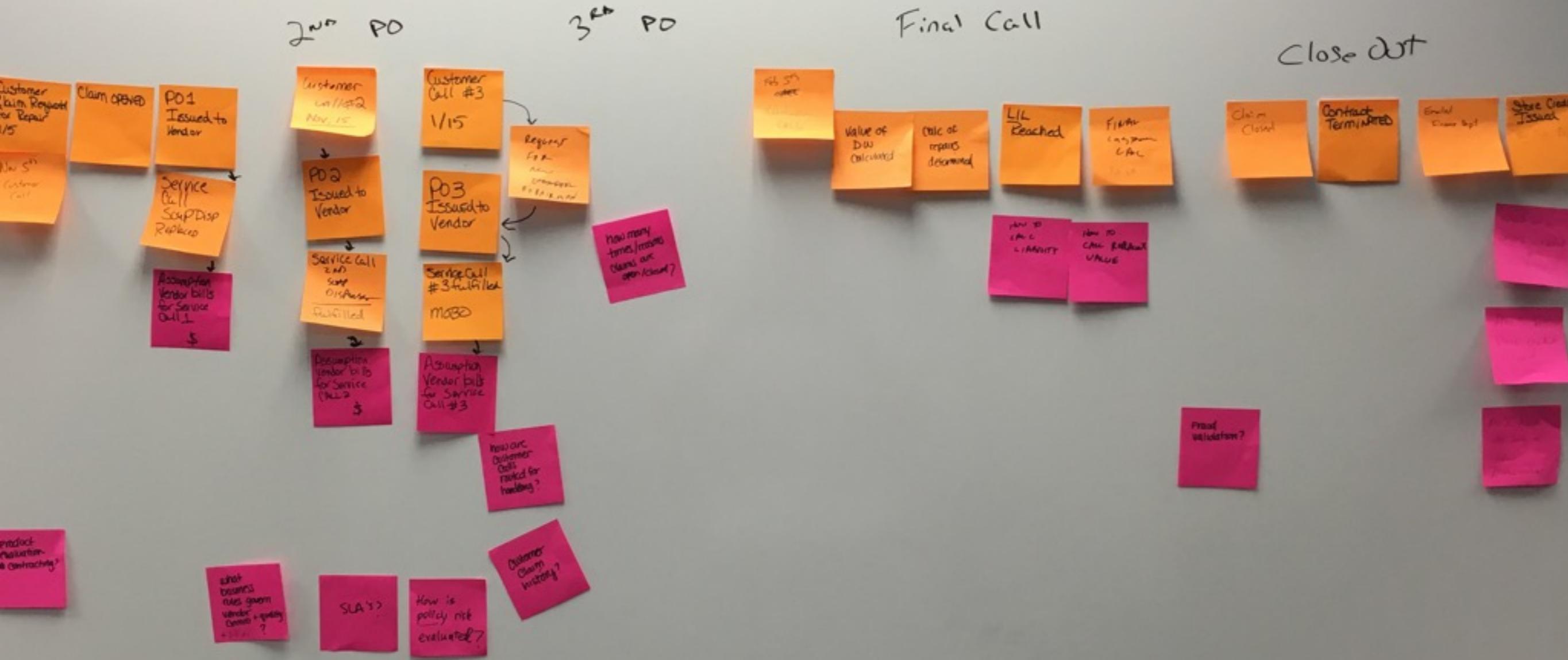
BILLABLE  
AMOUNT  
VERIFIED

INVOICE  
PREPARED



**EVENTS &  
QUESTIONS FIRST**





# EVENTSTORMING

## A LEGACY APPLICATION

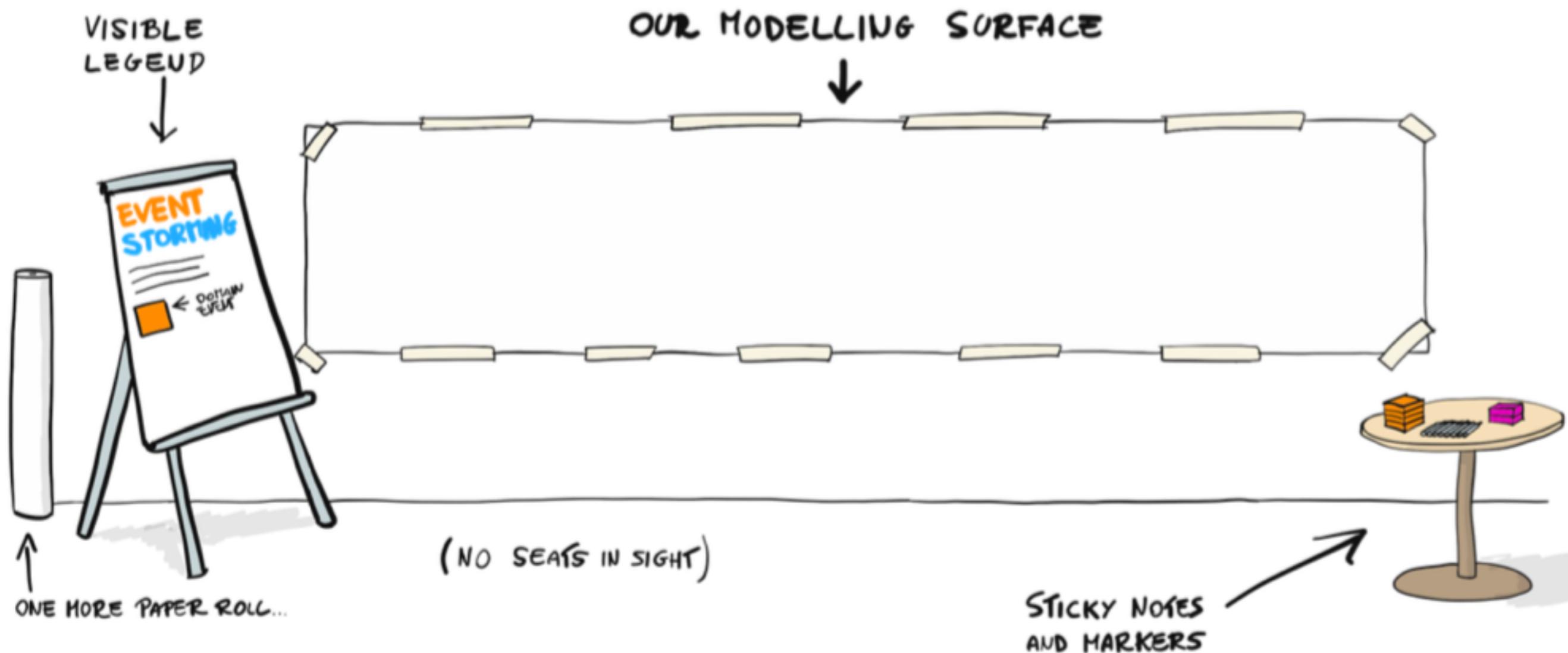


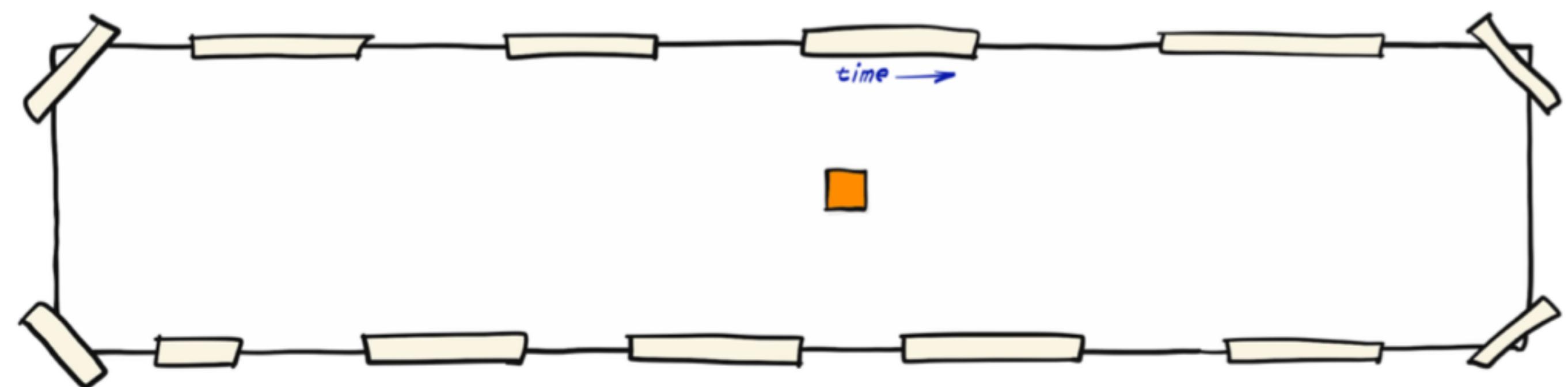


**PROVIDE UNLIMITED  
MODELING SPACE**









Once  
upon a  
time...

They lived  
happily ever  
after

More modeling space than you expect

**INVITE THE RIGHT  
PEOPLE**

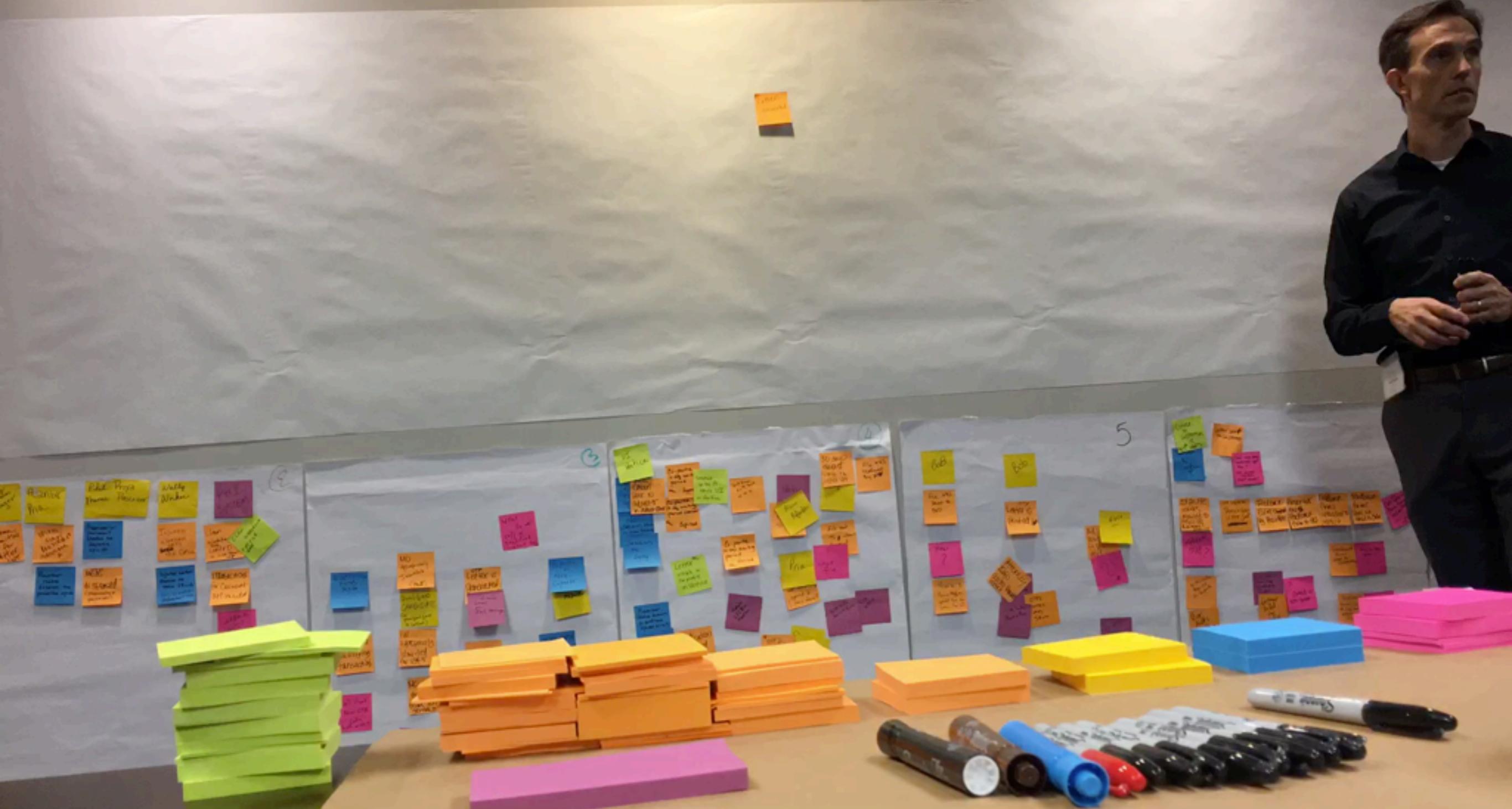
Development team members and business people gather in a room and *together* we create a visual map of the flow of events



# **EVENTSTORMING**

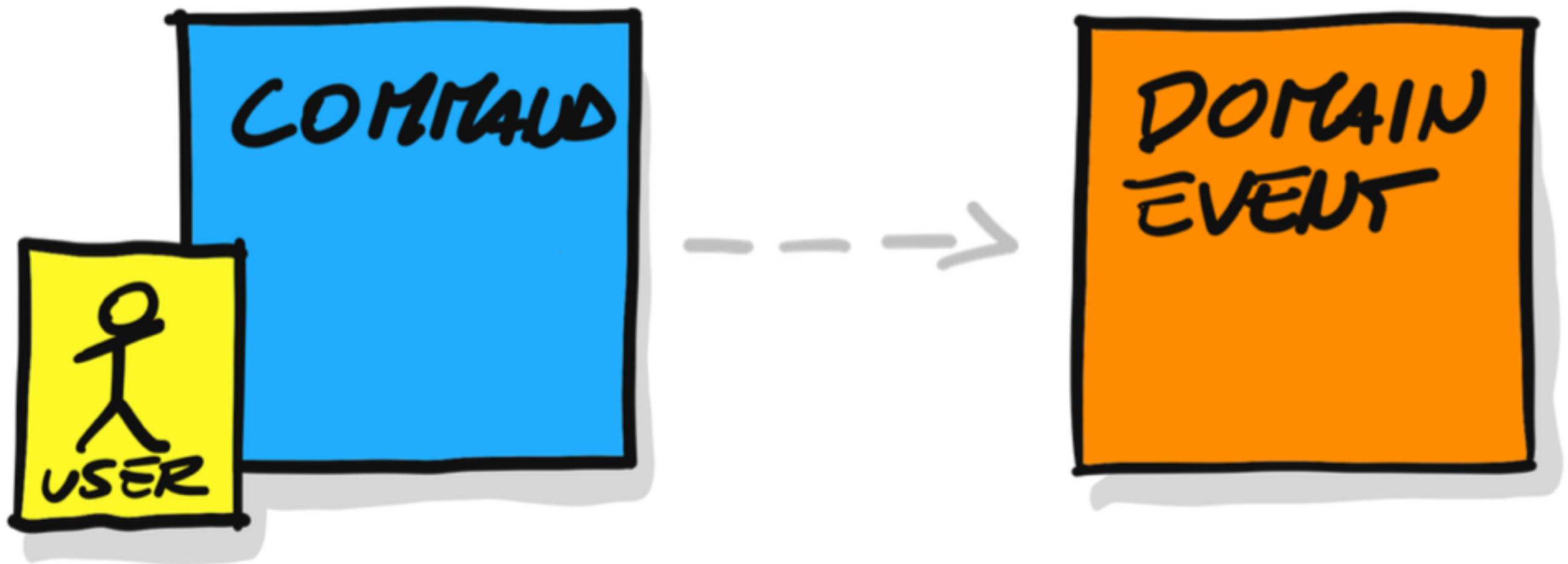
## **IN ACTION**



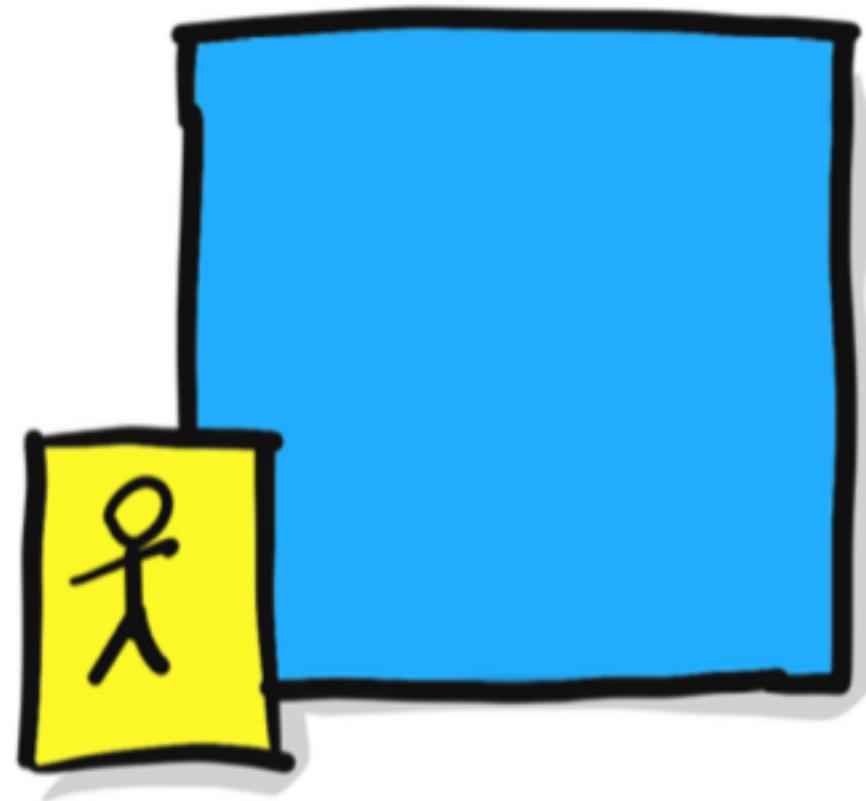




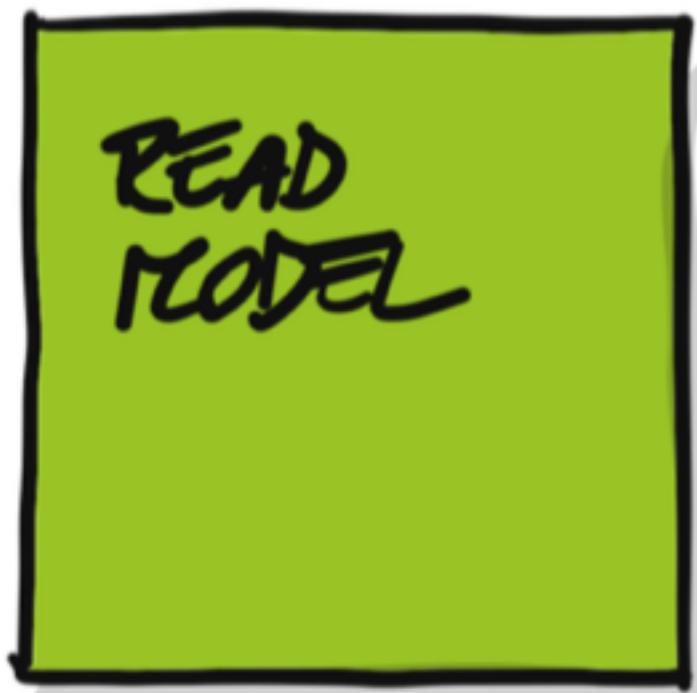
**VISUALIZE  
INTERACTIONS**



**USER INITIATED ACTION**



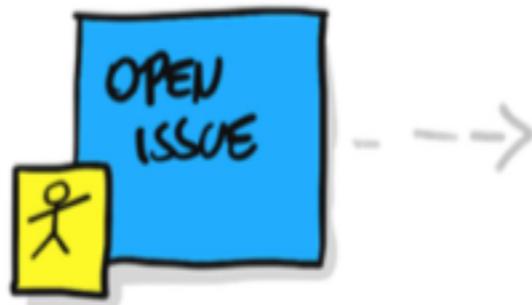
← THE DECISION



← THE DATA NEEDED  
IN ORDER TO  
MAKE THAT DECISION

# WHERE ARE DOMAIN EVENTS COMING FROM?

MAYBE AN ACTION  
STARTED BY A USER



**VISUALIZE  
BUSINESS RULES**

"WHENEVER"

THIS IS OUR  
**POLICY**  
SOMETIMES IT'S  
AUTOMATED,  
SOMETIMES  
PEOPLE JUST  
HAVE TO REMEMBER

**PROVIDE A  
VISIBLE LEGEND**

# EVENT STORMING

DOMAIN  
EVENT

- past tense
- "something that happened that the domain expert cares about"
- Immutable

?

- Assumption
- Don't know?

Actor/  
persona

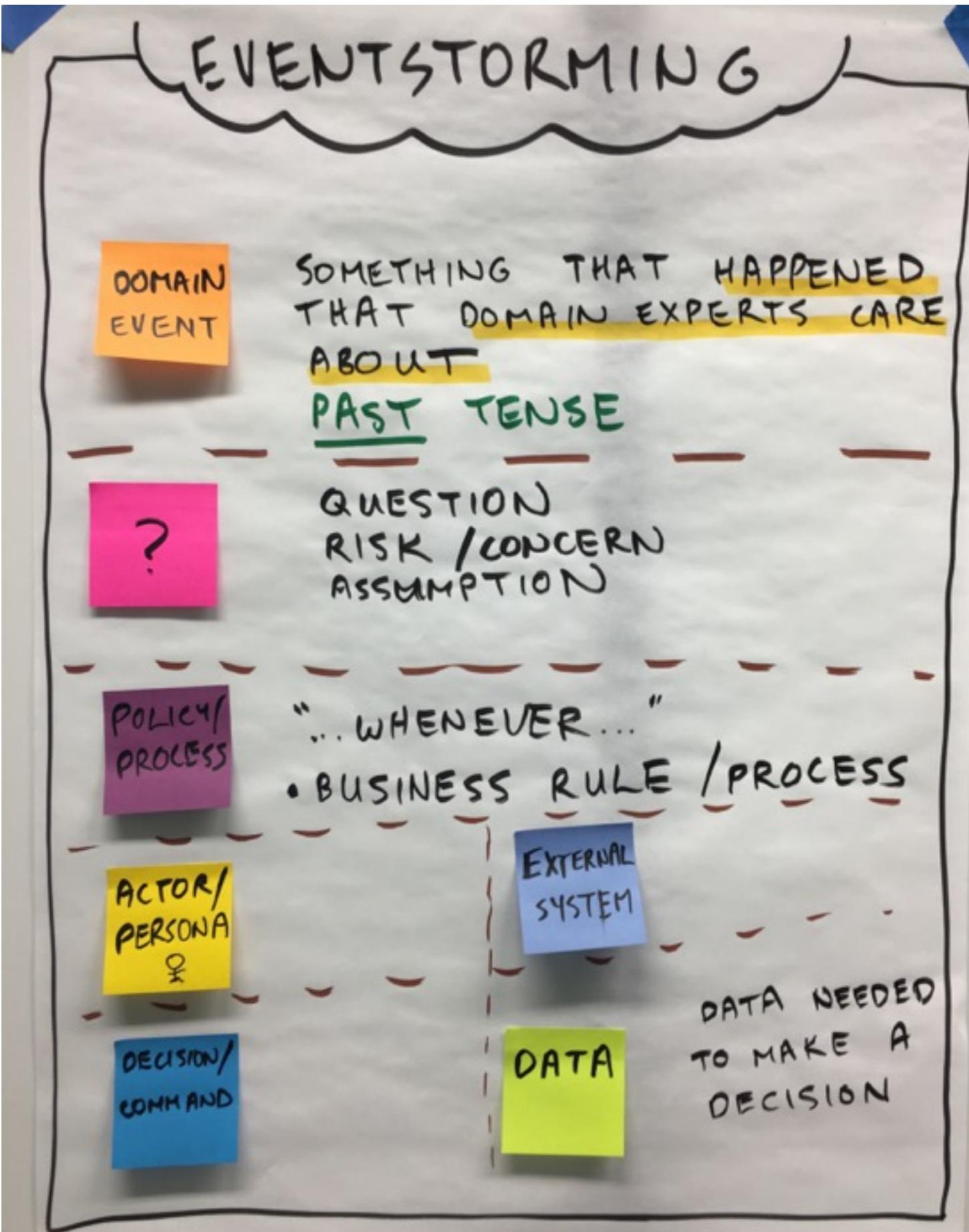
External  
System

Decision  
Command

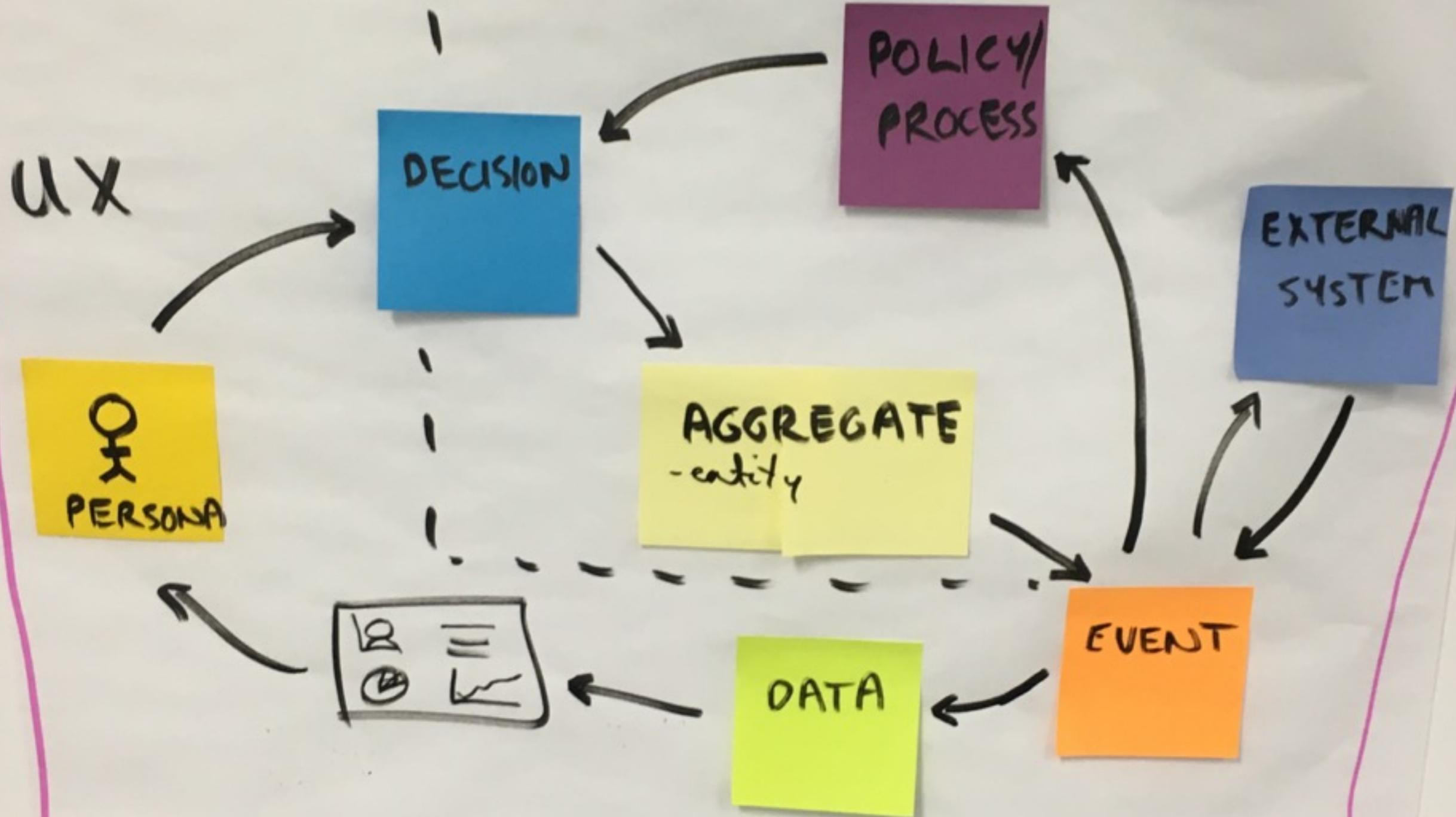
Read  
Model  
UI, file

Sources





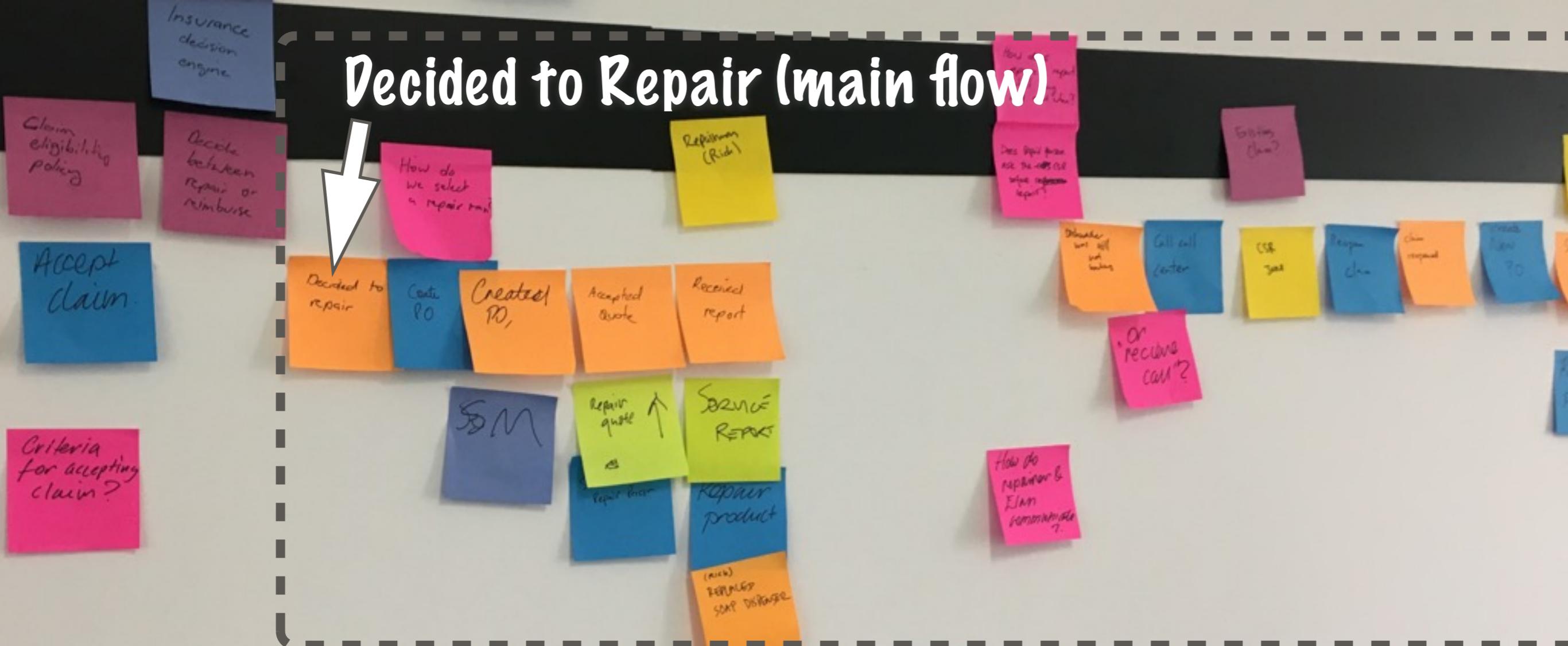




Adapted from [leanpub.com/introducing\\_eventstorming](https://leanpub.com/introducing_eventstorming)

**VISUALIZE  
ALTERNATE FLOWS**

# Decided to Repair (main flow)



# NATURAL TRANSITION TO EVENT SOURCING

BILLABLE  
AMOUNT  
CALCULATED

BILLABLE  
AMOUNT  
VERIFIED

INVOICE  
PREPARED

# EVENT SOURCING

---

BILLABLE  
AMOUNT  
CALCULATED

BILLABLE  
AMOUNT  
VERIFIED

INVOICE  
PREPARED

BillableAmountCalculated

BillableAmount
CalculationDate
AccountNumber
CustomerId

BillableAmountVerified

BillableAmount
OperatorId
VerificationDate
AccountNumber
CustomerId

InvoicePrepared

Invoiceld
TotalAmount
OperatorId
InvoiceDate
AccountNumber



# Aggregate state modeled as a sequence of domain events



# EVENT SOURCING

---

- ▶ Don't save the current state of objects
- ▶ Process events that lead to the current state
- ▶ TDD focused on command processing generating the correct events

BillableAmountCalculated
BillableAmount
CalculationDate
AccountNumber
CustomerId

BillableAmountVerified
BillableAmount
OperatorId
VerificationDate
AccountNumber
CustomerId

InvoicePrepared
Invoiceld
TotalAmount
OperatorId
InvoiceDate
AccountNumber



# **CLARIFYING CONTEXT BOUNDARIES**

# Service Management Context

EN  
IM  
~~aim~~  
~~pened~~  
and ways sorted?  
Claim  
Management  
Scrn

what  
action  
should  
be taken?

CSR

Service  
machin.

Repair  
PO  
Received.

Dispatch  
technician

Pick.

ASSESS  
Damage.

Repair

unable  
to repair.

Machine  
repaired

labour  
or  
part  
cost?

Cost  
incurred

Outcome  
coming  
back.

Outcome

Decision  
on  
claim  
share

CSR

CLOSE  
CLAIM

Repair  
PO  
~~date~~  
~~address~~  
~~description of fault~~

CREATED  
PURCHASE  
ORDER

Create  
Purchase  
order

What  
(lost)  
machin.  
liability  
reached

CSR  
♂

reimburse  
costs

EMAILED  
FINANCE DEPT  
FOR  
STATE  
CREDIT

FINANCE

Re  
Reimburse

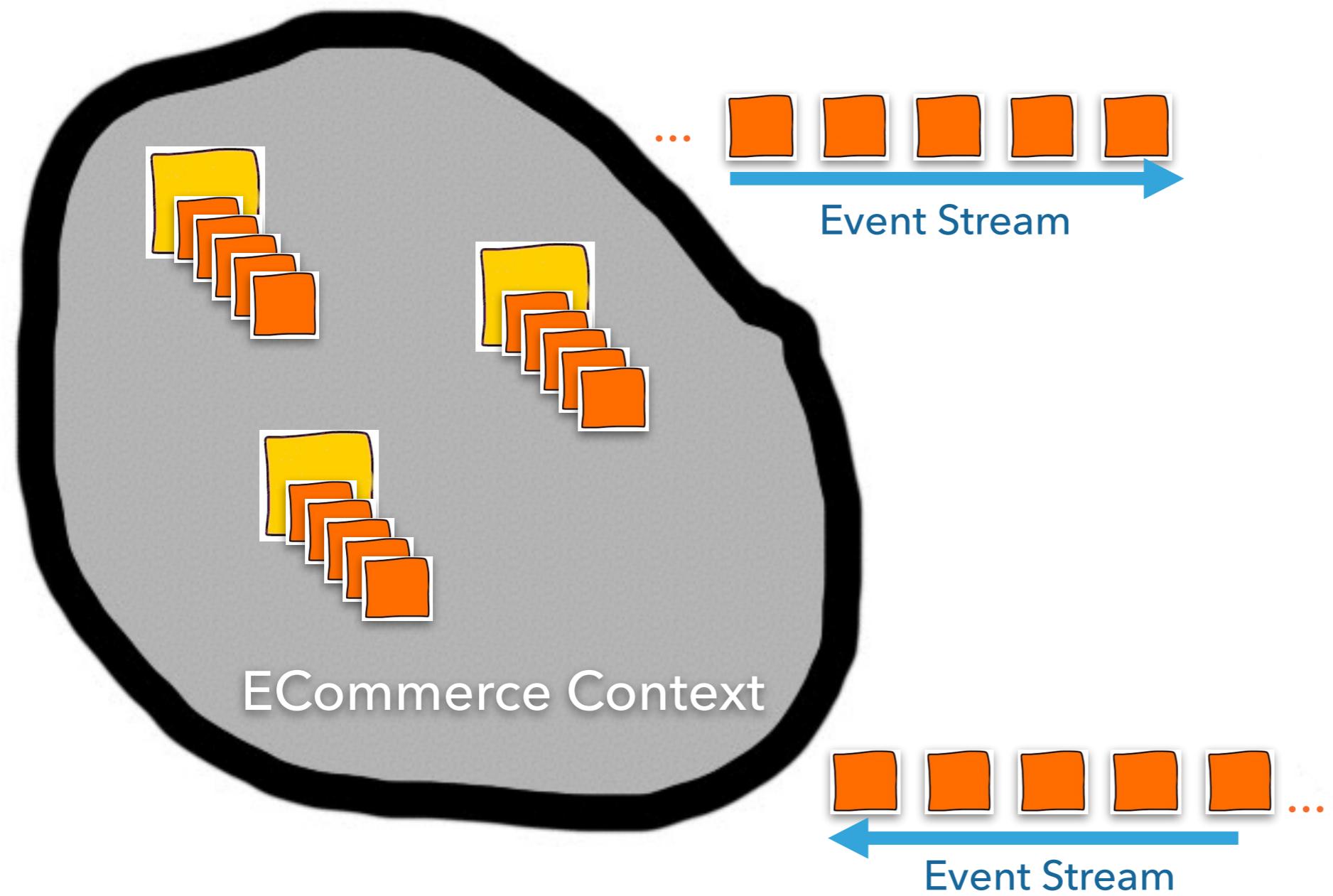
Voucher  
Value \$P

reimburse  
(for value  
of deliv.  
worker)

Close  
claim  
terminate  
contract

# Claims Context

**NATURAL TRANSITION TO  
DISTRIBUTED SYSTEMS**



# FACILITATION TIPS

## EVENTSTORMING FACILITATION TIP #1

---

Focus first on  
learning and  
understanding

## CLARIFYING QUESTIONS

---

- ▶ “What circumstances would cause ... to happen?”
- ▶ “What was the path that led us here?”
- ▶ “What is a good example of ...?”

# CLARIFYING QUESTIONS

---

- ▶ "What do you mean by ...?"
- ▶ "What might lead someone to do/need ...?"
- ▶ "What else might happen...?"

## EVENTSTORMING FACILITATION TIP #2

---

Anchor to a  
concrete business  
example

Clarify fuzzy  
concepts and  
language

nty  
#

ased  
aled  
ty  
Oct.

DISHWASHER  
FAILED  
#1-4

MFR  
WARR'TY  
EXPIRED

CSR  
(JANE)

how to  
ID the  
customer?

OPEN  
CLAIM

Customer  
called  
call center

Claim  
~~opened~~  
opened

Is  
claim  
always  
created?

CUSTOMER  
SEARCH  
- Policy #?  
- Name, postcode

Claim  
Management  
Scrn

Command is "Open Claim"

What should associated  
event be called?

Conversations  
and examples are  
primary

## EVENTSTORMING FACILITATION TIP #6

---

Post first,  
Ask questions  
later

## EVENTSTORMING FACILITATION TIP #4

---

# Timebox

"Everybody knows the problem: We need to be more innovative. Now we've got the solution: *Gamestorming*. This smart, fun, hands-on book will energize your brain and mobilize your creativity—and do it using stuff you already have in your supply closet!"

—Daniel H. Pink, author of *Drive* and *A Whole New Mind*

# Game storming

A Playbook for Innovators,  
Rulebreakers, and Changemakers



Dave Gray  
Sunni Brown  
James Macanufo

O'REILLY®

O'REILLY®

סםאי מכאנווּדוֹ  
סֻנְנִי בָּרוּן  
דָּבָר צַרְאָל

# POSSIBLE LIMITATIONS

---

- ▶ Simple business domains (e.g. CRUD) and/or systems
- ▶ Distributed teams
- ▶ Larger group/greater complexity = better facilitation
- ▶ ???

**WHY  
EVENTSTORMING?**



every one  
gets to  
understand

encourages  
complexity  
seeking

visualize  
business  
process

Imagery  
details

encourages  
asking  
questions

visual  
cues for  
different  
parts of the  
systems

over (idea)

faster than  
drawing

A  
GET DE BETTER  
UNDERSTANDING  
OF THE PROCESS

Exploration,  
communication

## WHY EVENTSTORMING?

---

Collaborative learning

*Cultivate mutual shared  
understanding*

*Uncover misunderstandings & missing  
concepts*

*Avoid rework*

## WHY EVENTSTORMING?

---

Model & explore complex domains

*Visual, tactile representation* of system behaviour & outcomes

Smoother transition to a rich domain model, design and coding

# QUESTIONS?

---

Paul Rayner  
 @thepaulrayner



# THANKS!

---

Paul Rayner  
 @thepaulrayner

