

From Days to Hours: How EventStorming Transformed Our Domain Modeling Process

Vadzim Prudnikau



HUMAN MADE
NO AI USED

This talk is NOT about...

- How to learn EventStorming in 40 minutes

Shortly about me

I run training in EventStorming, Architecture, DDD, and TDD



Vadzim Prudnikau

 [vadim-prudnikov](https://www.linkedin.com/in/vadim-prudnikov)



Co-owner and instructor at  trainitek

Hands-on architect at  APOTEK 1

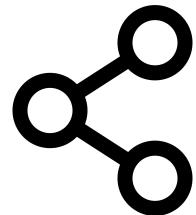
Today, my Goal is...

- To show **how** EventStorming can be **useful for developers**
- To **inspire you to use** EventStorming when you are back at work

What will we address today?

- **Why** it makes sense to consider EventStorming
- **How** to run EventStorming
- **Additional questions** about EventStorming
- [If we have time] Two **interesting** real-life **stories**

Let Me give You some Context



How many of you agree that...

- *“Many people work on projects they actually don’t understand”?*
(Sergey Baranov)
- *“Systems often contain wrong developers’ assumptions about functions and dependencies rather than knowledge of domain experts”?*
(Alberto Brandolini)

Did You experience the following?

- One Architect spends a lot of time on gathering requirements from different people and then assumes that developers will quickly understand what they will implement.

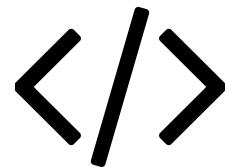


... and that architect disappears :)

What helps?

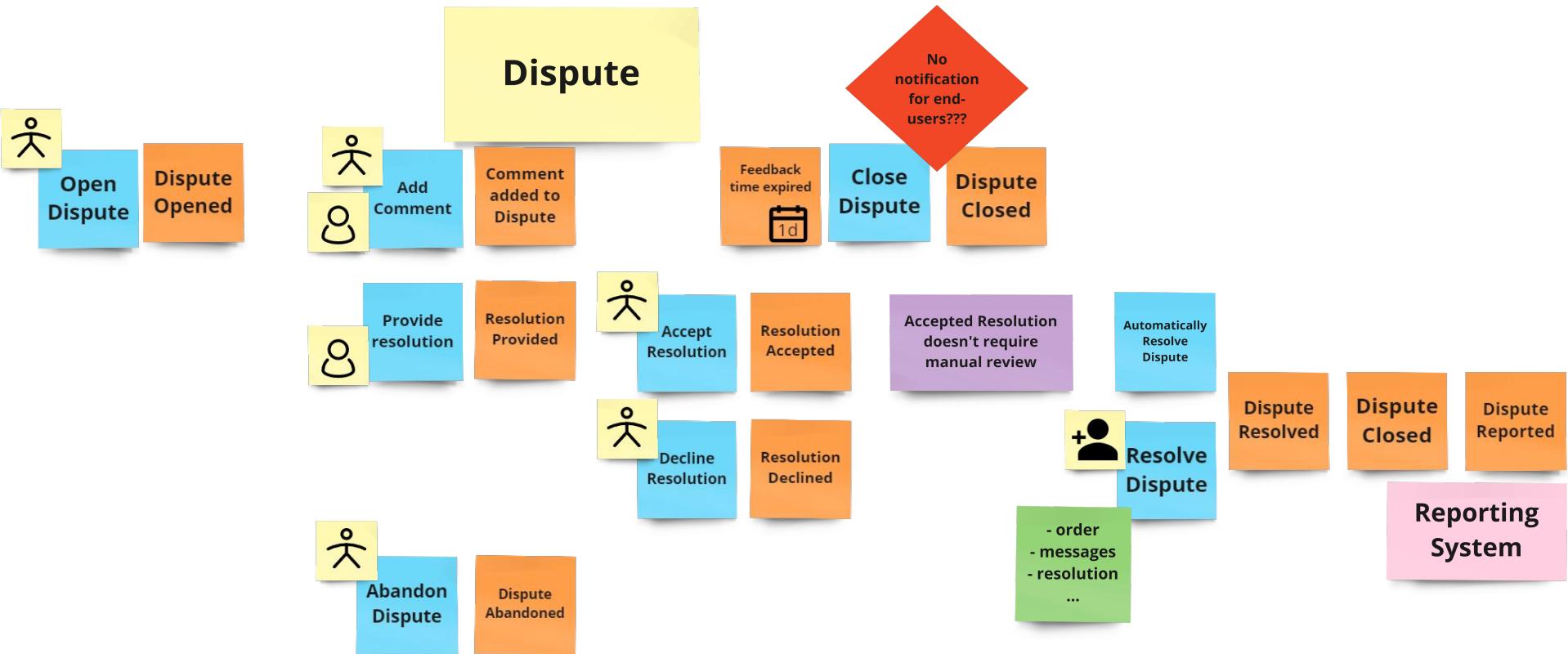
- **Discovering implicit dependencies** on the business level
- **Transferring the knowledge** between important people (domain experts, developers, etc.)
- **Discovering the most important events and problems** (bottlenecks, legal restrictions, etc.)
- **Solving business problems** rather than blindly implementing “requirements”

Maybe it's too abstract...
Could it be closer to the code?

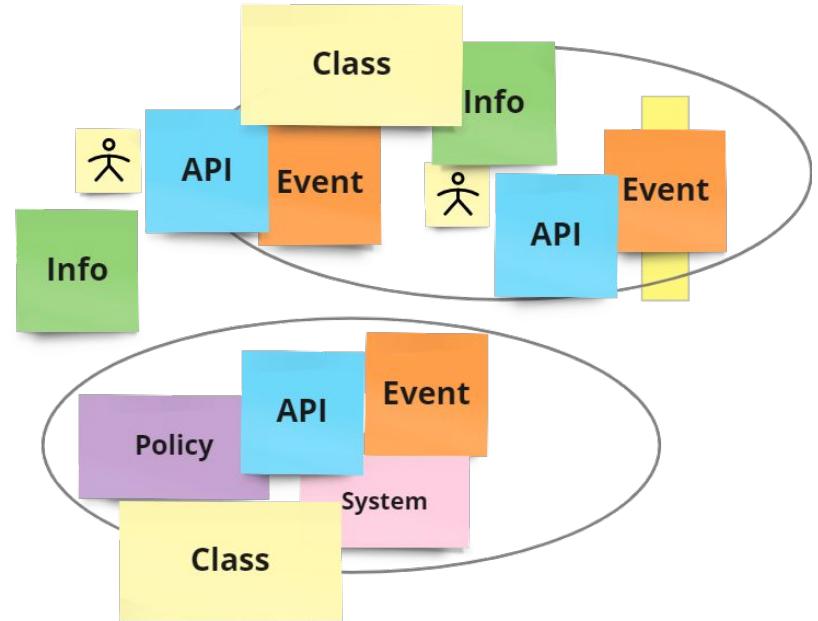
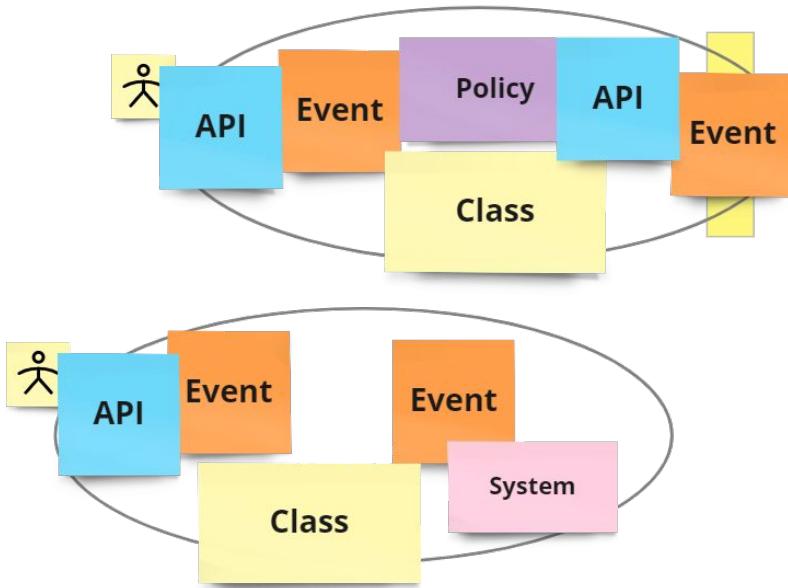




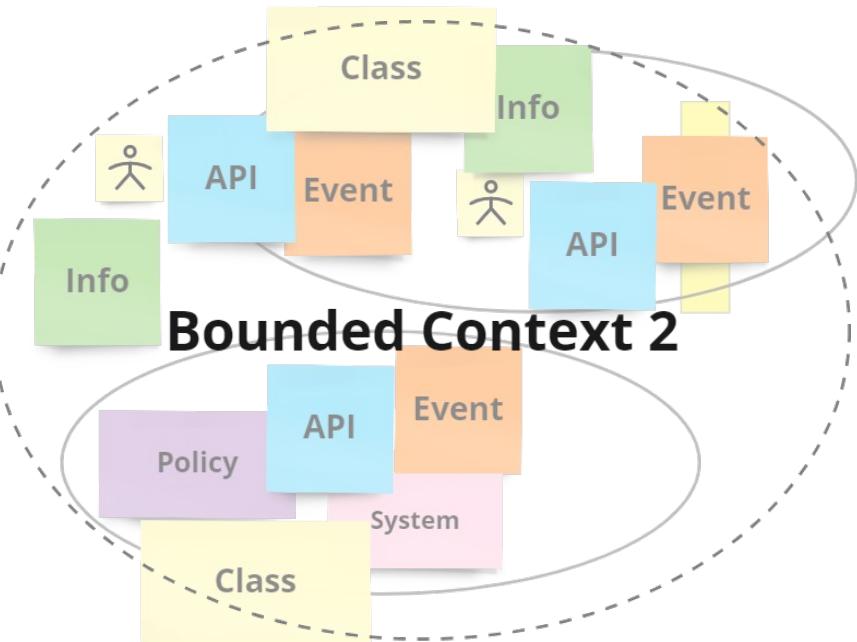
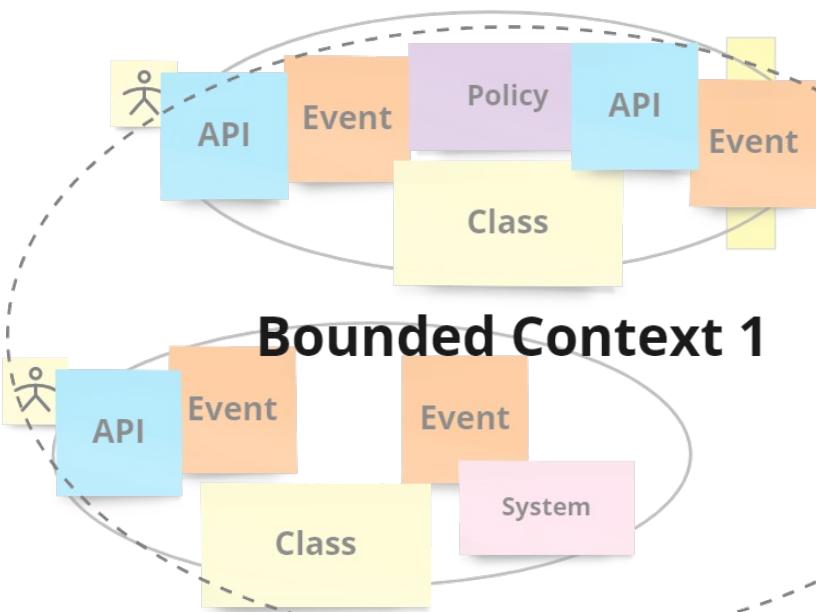
What if you had...



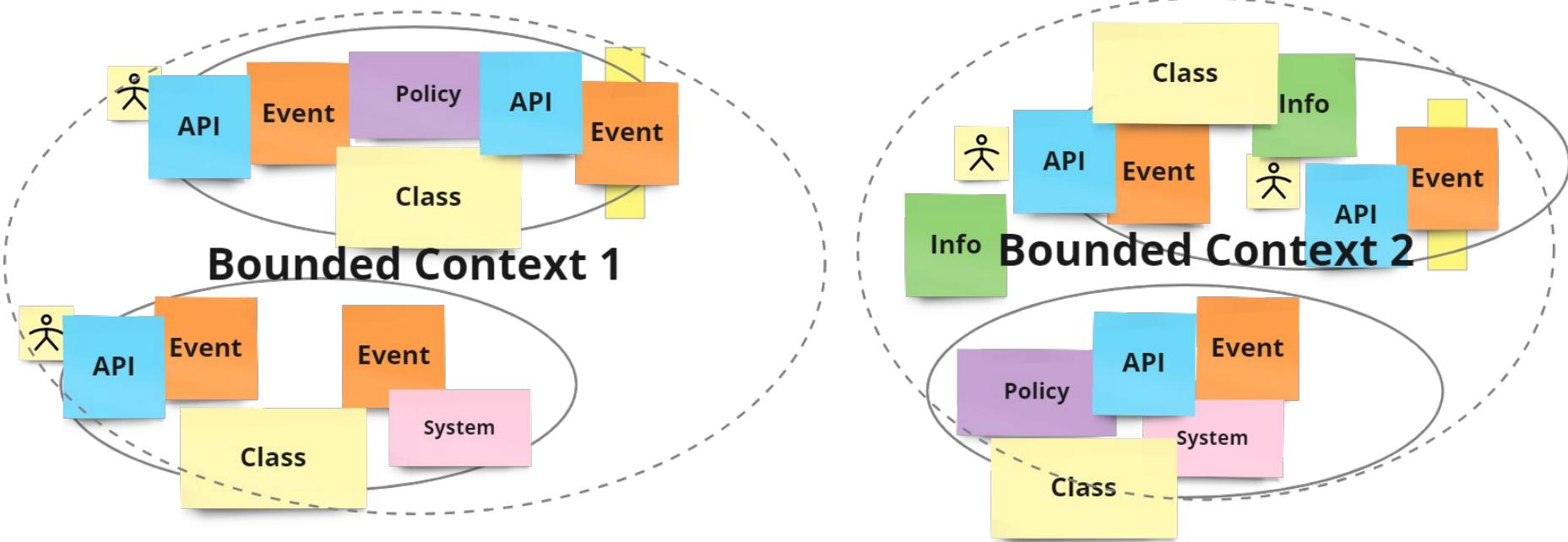
What if you had...



What if you had...



Wouldn't it be nice to have it BEFORE you start coding?



... and this can be done in hours!

Let's understand EventStorming a bit better!



What is EventStorming?

EventStorming is a flexible **workshop** format that brings **together** developers, business experts, and other interested parties to perform a **massive collaborative exploration** of complex domains.

It runs for **hours instead of days, weeks, or months**.



Some examples...

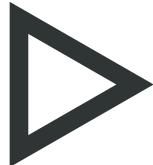


WHY EventStorming?

- **Faster and more enjoyable** than traditional methods
- Helps developers and business people **get know each other**
- Helps find a **common language** between developers and business people
- Helps discover important questions **before writing the code**
- Helps get a **strategic design**, i.e. define boundaries and their dependencies

How to Run EventStorming?

(I will omit some details because of the limited time)

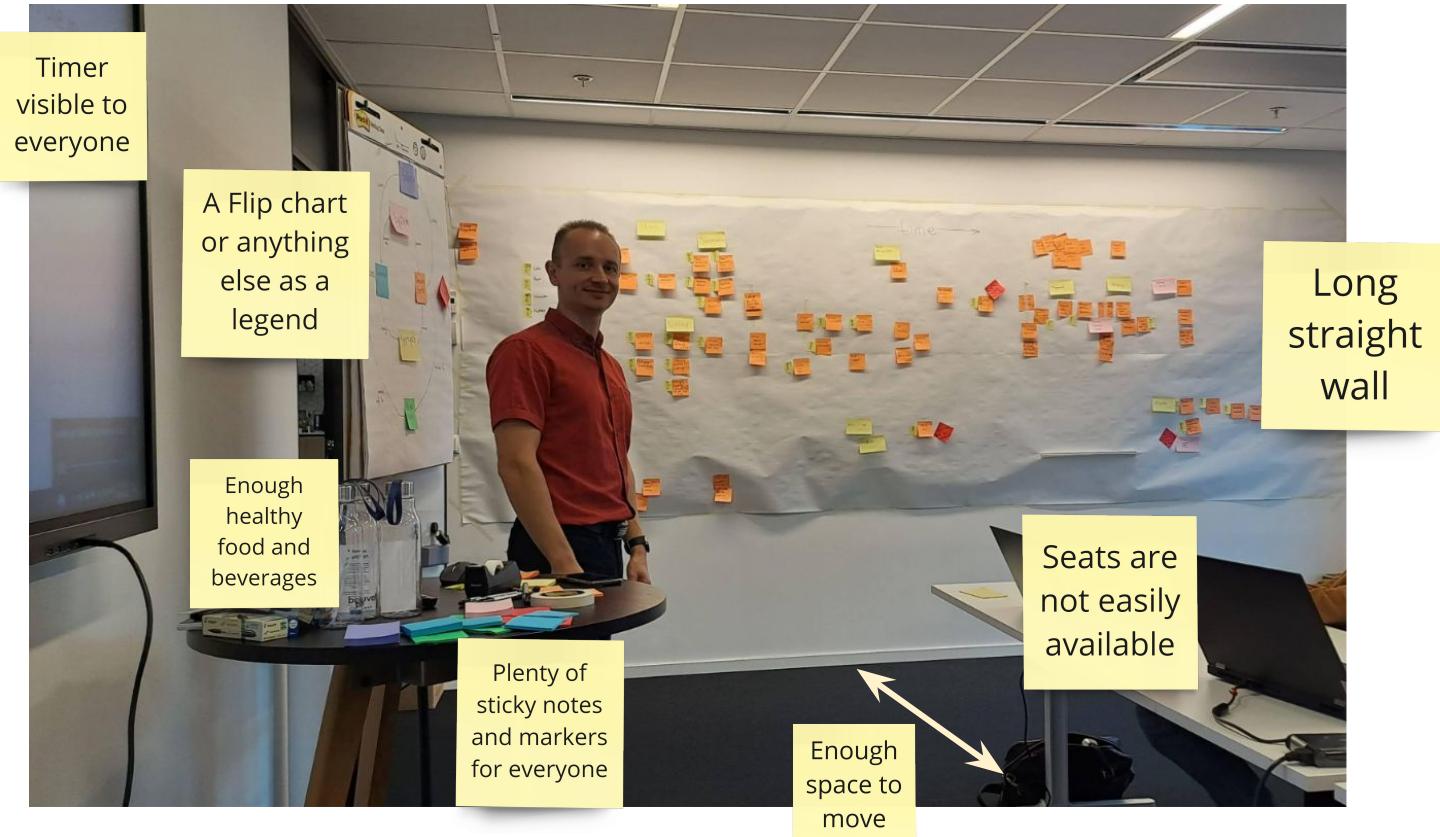


1/9 Invitations

- Try to find as many **people** as you can who are **important** for the exploration:
 - The ones who have the knowledge (domain experts, real users, management, etc.), i.e. those **who have answers**
 - The ones **who ask questions** (developers, testers, etc.)
- Some of them won't show up, therefore think about **backups**
- There should be clear who will be a **Facilitator**:
 - Workshop preparation
 - Explanation of what to do
 - Management of long conversations
 - Being full of energy
 - ...etc.
- Give the context to the audience



2/9 Room Setup

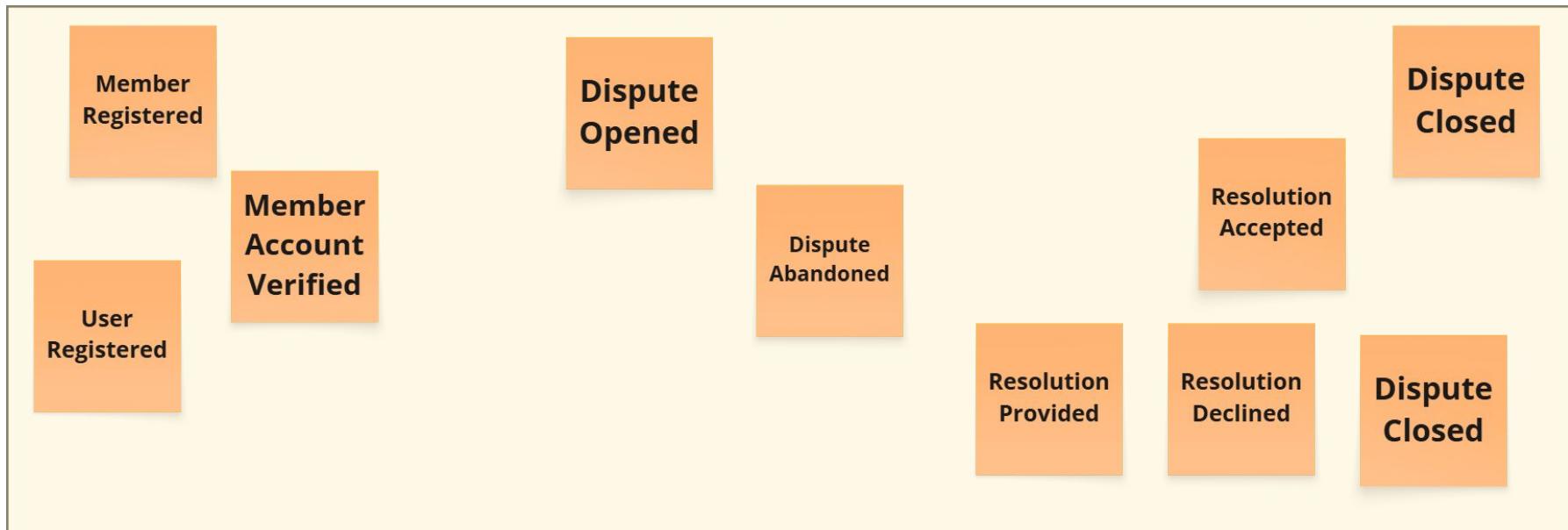


3/9 Chaotic Exploration (Big Picture ES starts)

Find Domain Events and place them along timeline to represent the whole business flow



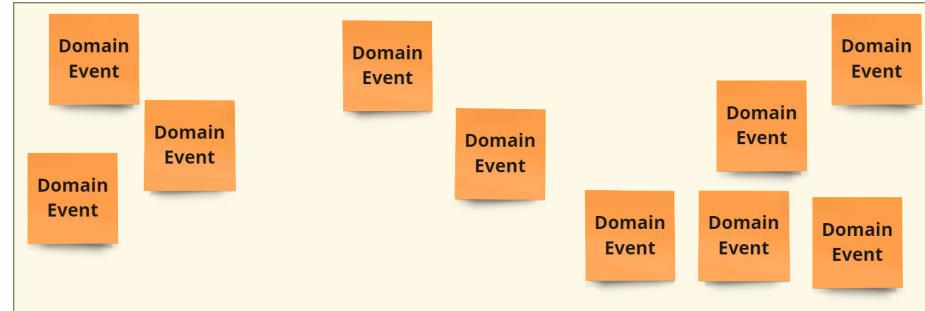
10-15 minutes



Why is it useful?

- It's the **first step to find the “common” vocabulary** for the participants
- Everyone's **knowledge starts to be visible** to others
- Most of the Events will be **events in your code**

Tip: write domain events so others can read

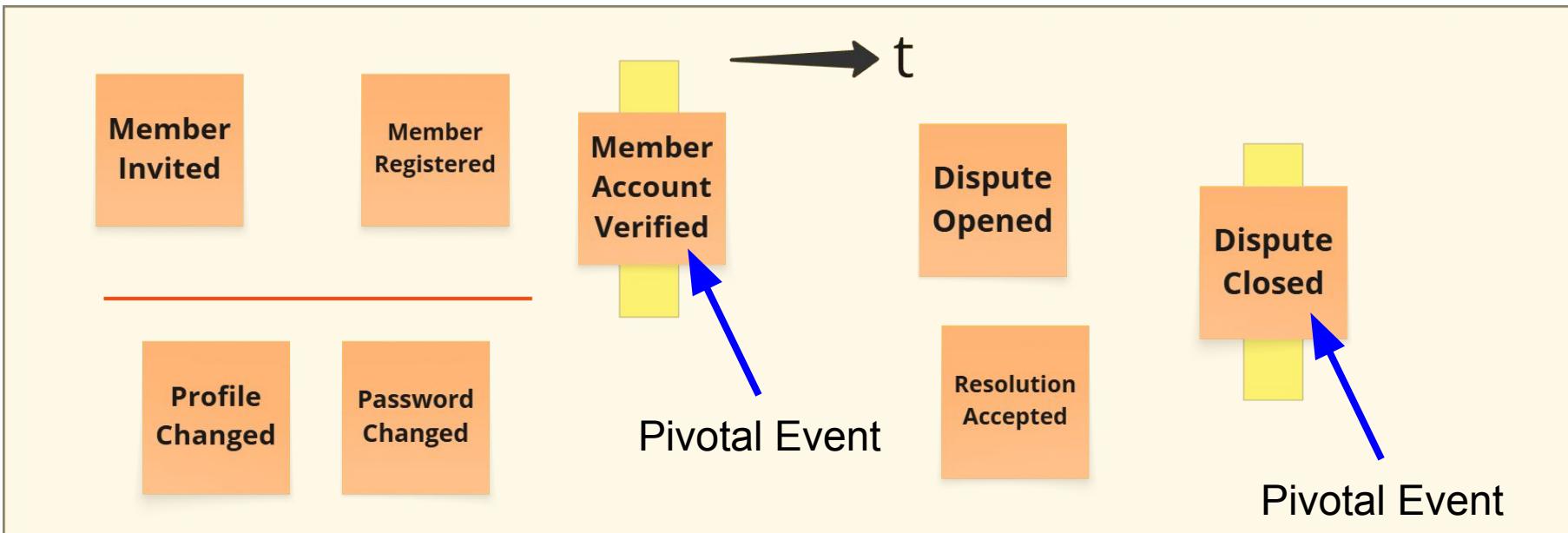


4/9 Enforcing Timeline

Order the events according to the time, employing different strategies.



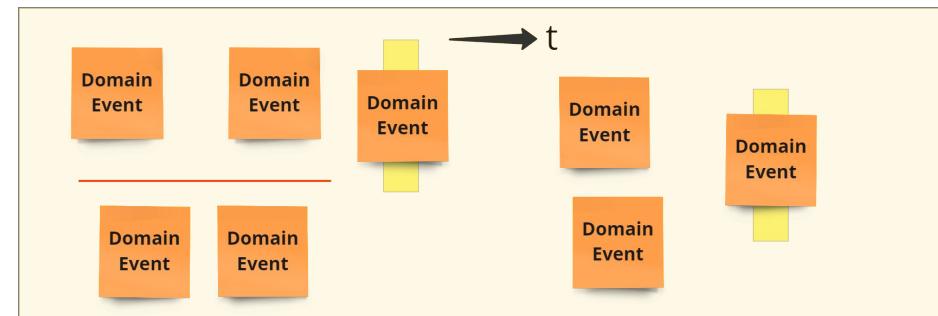
20-60 minutes



Why is it useful?

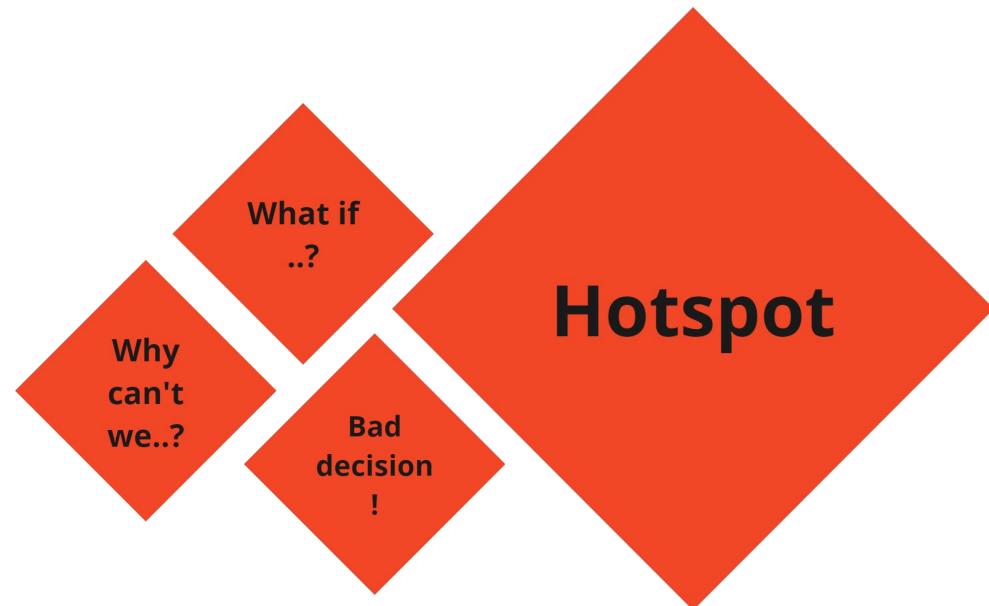
- Pivotal Events and Swimlanes can **help you identify service boundaries** in the future
- **Understanding** of the system **gets better**. Many questions arise. Some of them cannot be answered and turn to Hotspots.

Tip: work in dynamic groups



Feel free to add Hotspots

- Represents **unclear topics or questions** that cannot be easily resolved just at the very moment.

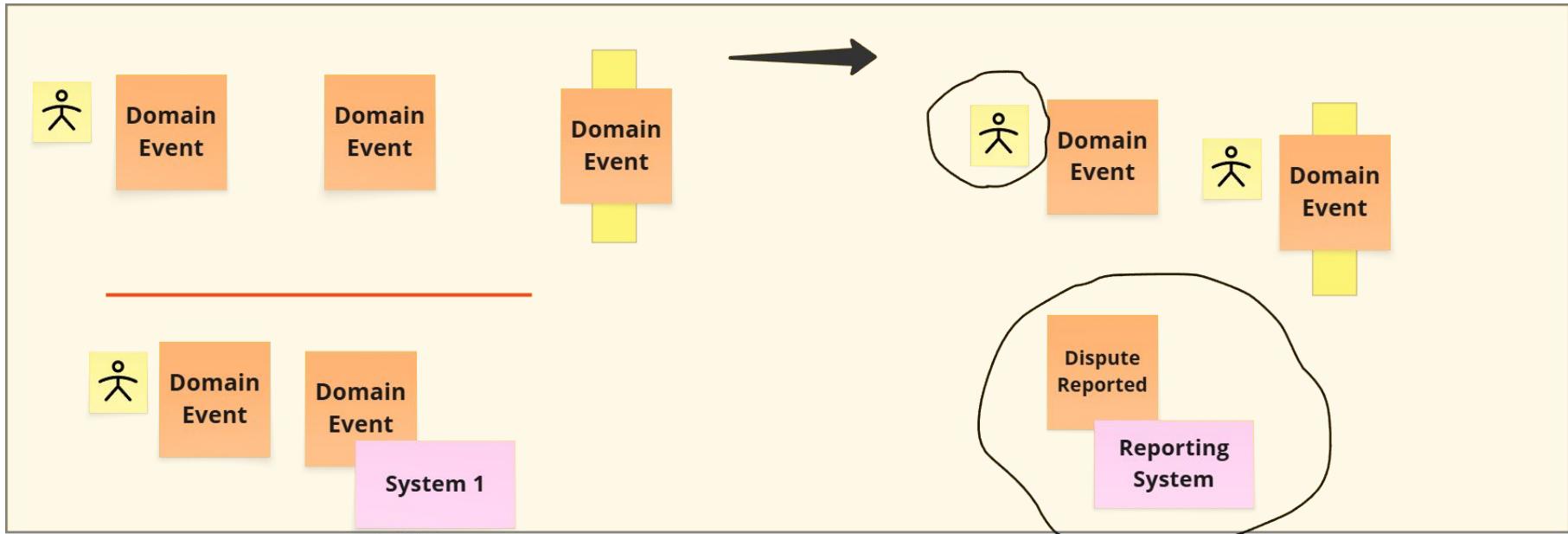


5/9 Actors and Systems

Find Roles (People, Actors) and Systems that participate in the business flow.



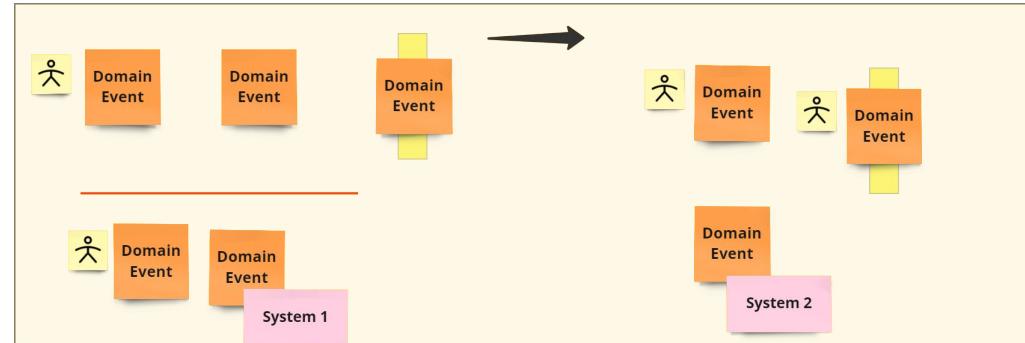
20-60 minutes



Why is it useful?

- You get **more knowledge about who** will use your system
- External systems show the **integrations you will implement**
- This part usually leads to even more “unanswered questions” (Hotspots)

Tip: don't force participants to draw humans



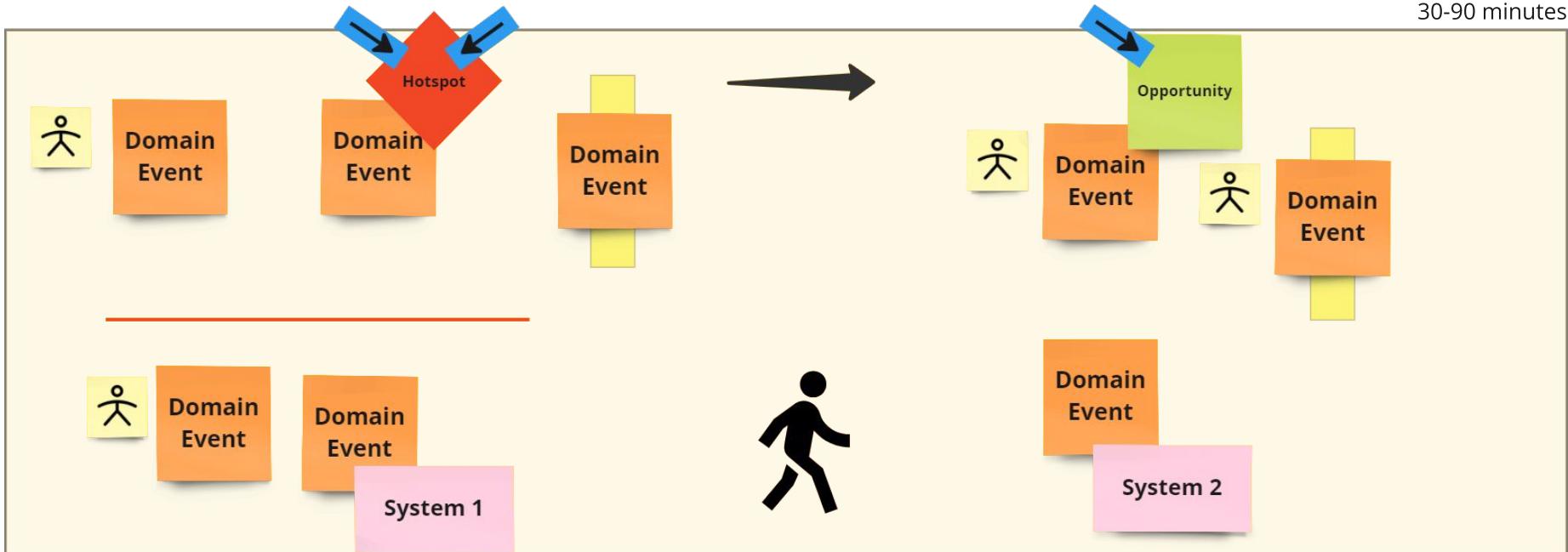
6/9 Explicit Walk-Through, Problems and Opportunities

A narrator tells the story from the beginning to the end, and/or vice-versa.

Placing **Hotspots** and **Opportunities**. [Optional] Arrow voting.



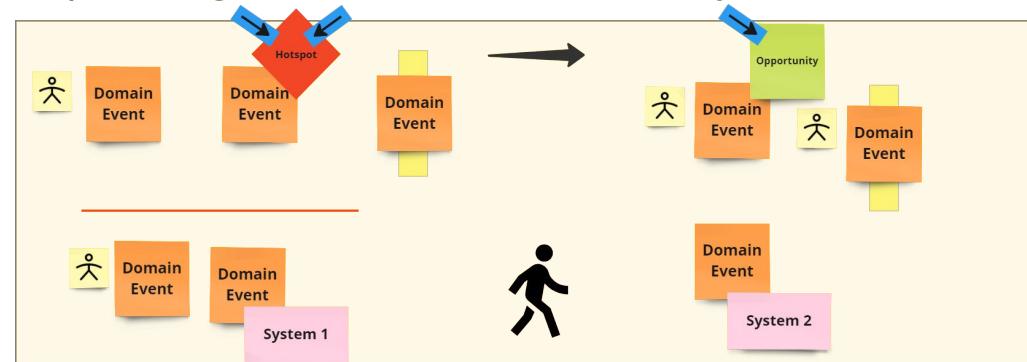
30-90 minutes



Why is it useful?

- The whole group can **verify the truthfulness** of the entire story
- Common **understanding** becomes **even better**
- This part usually results in more sticky notes like Domain Events, Hotspots, and Opportunities

Tip: change the narrator, if necessary

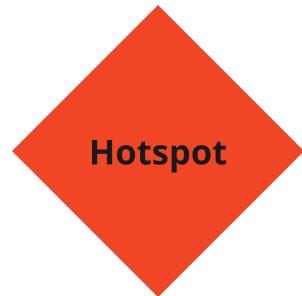


Big Picture EventStorming Done!

Some Useful Tips:

- ⏳ All the phases should be time-boxed
- ! Not all domain events with the same text are duplicates
- 💬 Long conversations should be politely interrupted and replaced by hotspots
- 🖐 Avoid using technical terms ("aggregate root", "bounded context", etc.) while talking with domain experts
- 👑 Prepare for “personal” issues (long discussions, a person dominating others, etc.)
- 👀 Discuss only visible things

Before going further, it's important to resolve
the most important hotspots

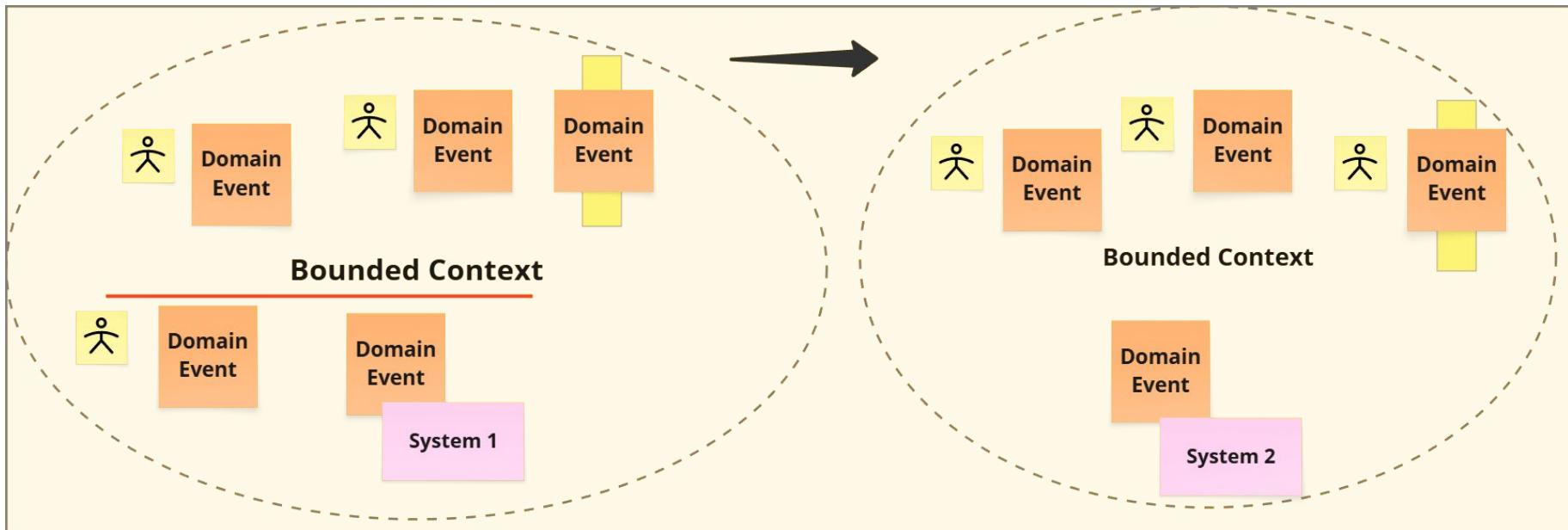


7/9 Finding Bounded Contexts

Outline Bounded Contexts on a technical meeting. Evaluate vocabulary, roles, related business scenarios, and importance of different parts of the entire system.



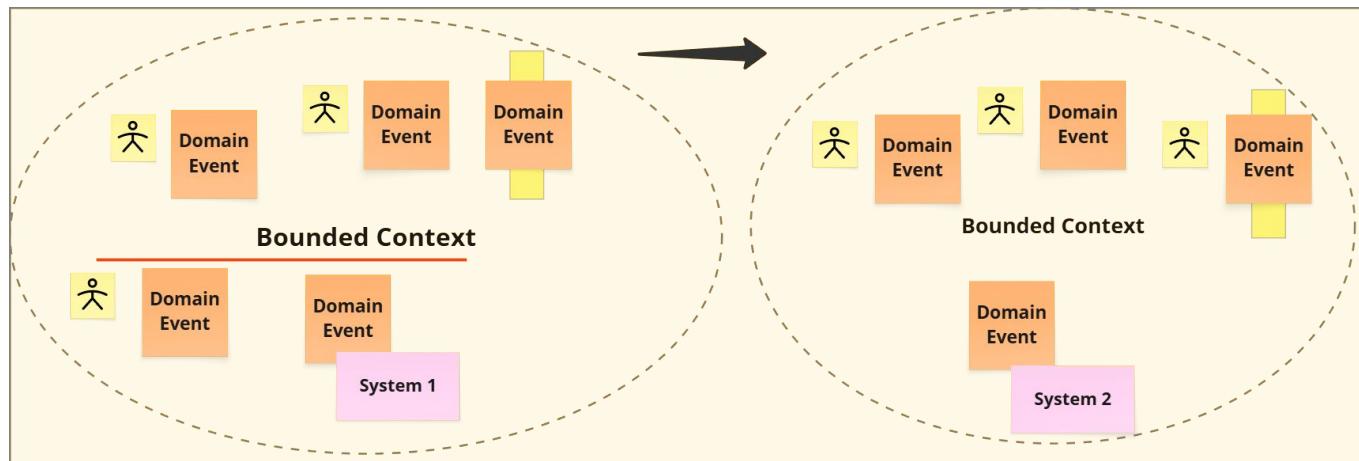
15-60 minutes



Why is it useful?

- You understand what services you will have

Tip: domain experts may also help find bounded contexts

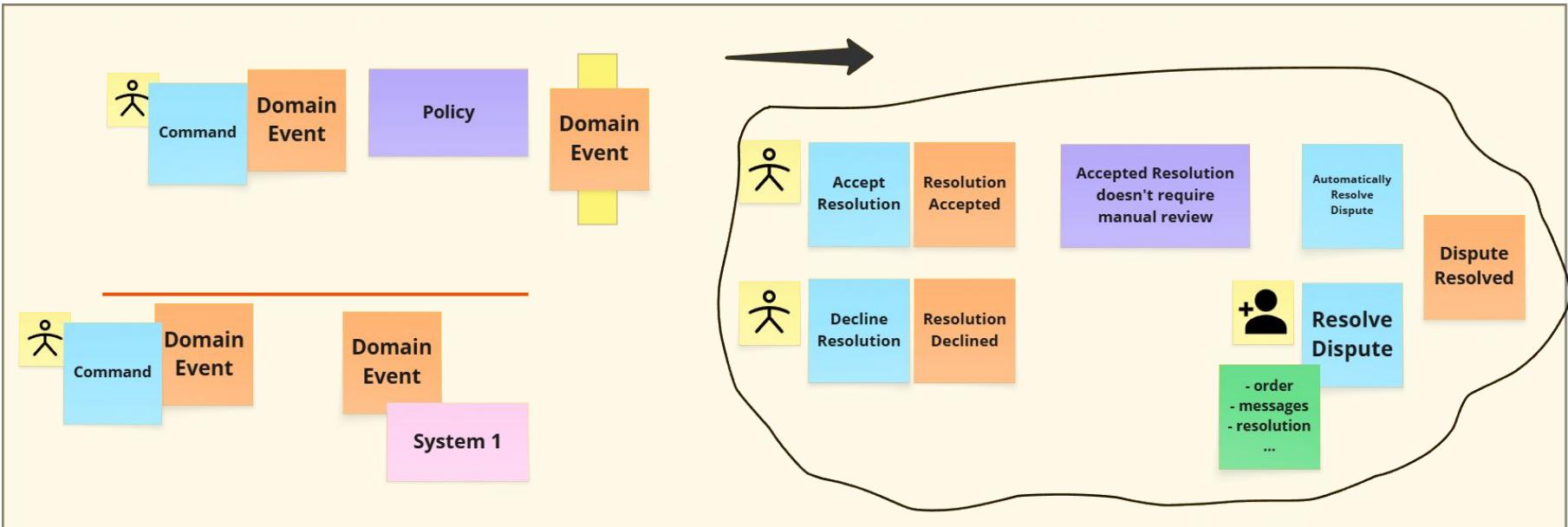


8/9 Process Modelling

Focus on a specific part of the entire business process in smaller groups.
This is a step further towards the technical solution.



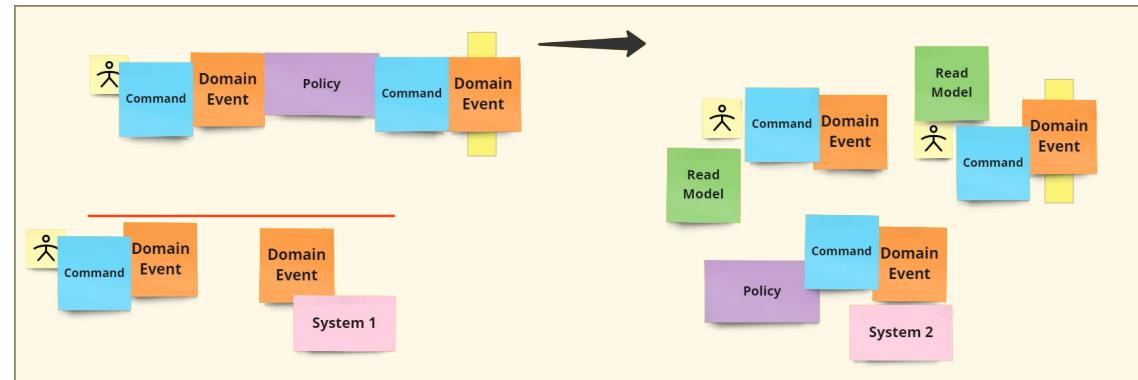
1-2 hours



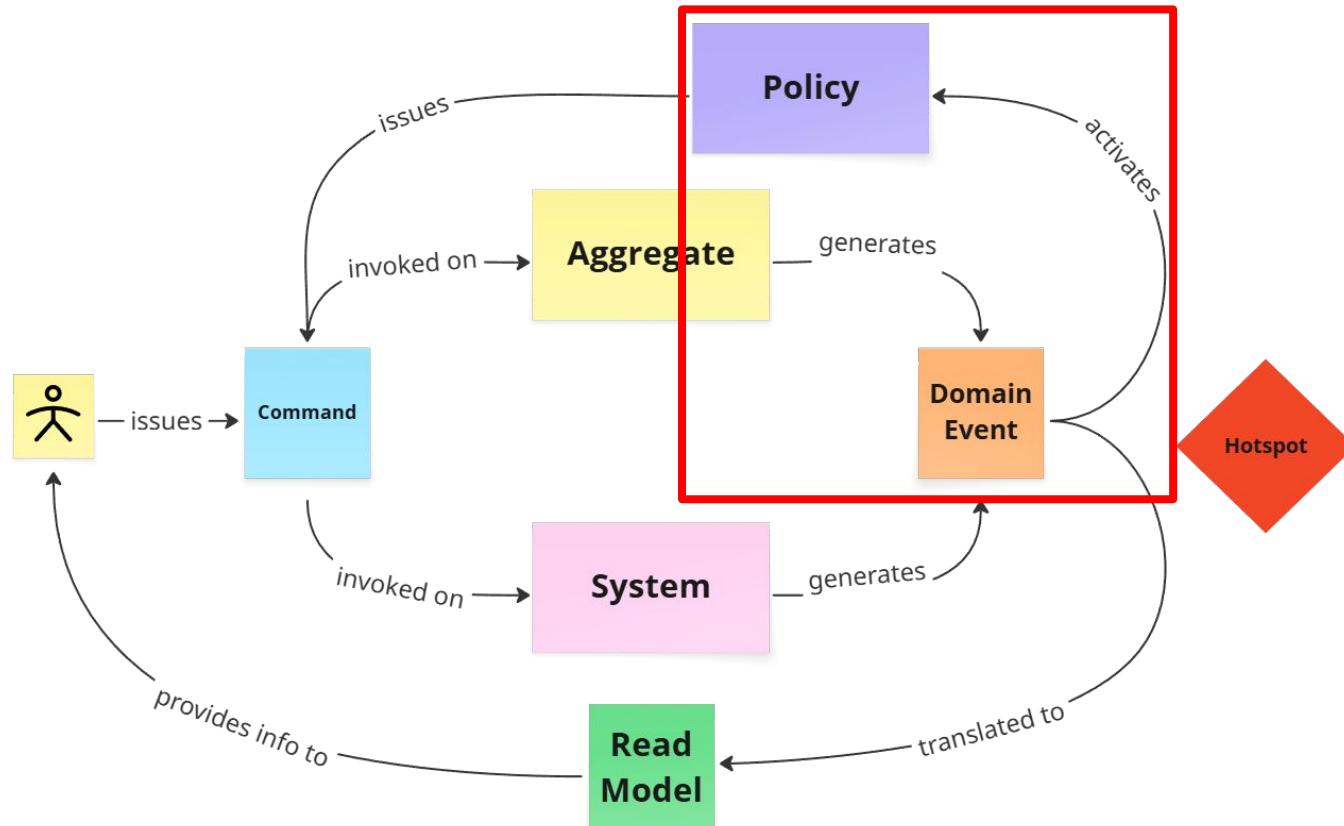
Why is it useful?

- **Commands** heavily contribute to your future **API design**
- **Read Models** can help with **UI-related decisions**
- **Policies** may turn to **important domain logic** in your code

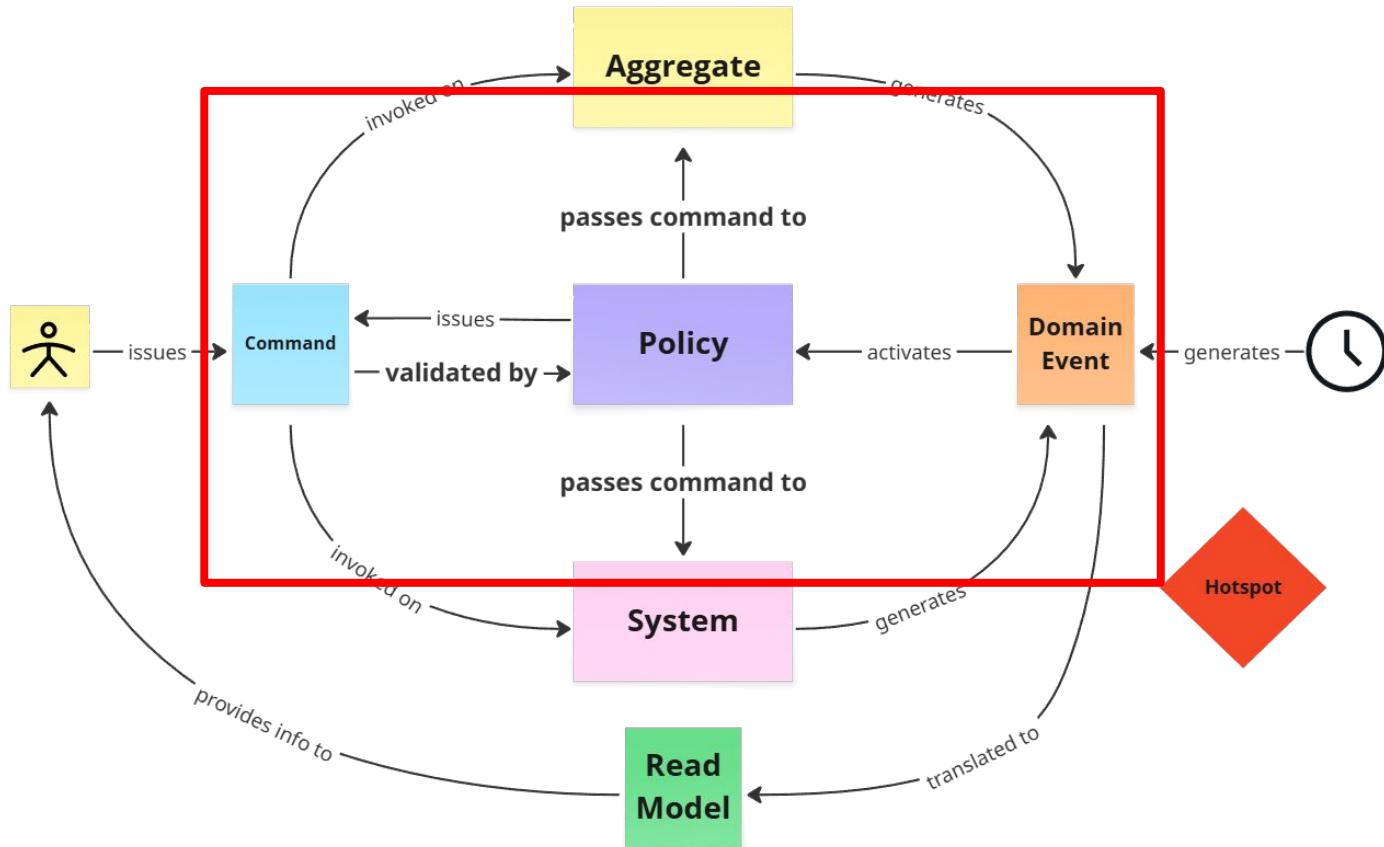
Tip: Don't wait for process modeling if you discover a policy earlier



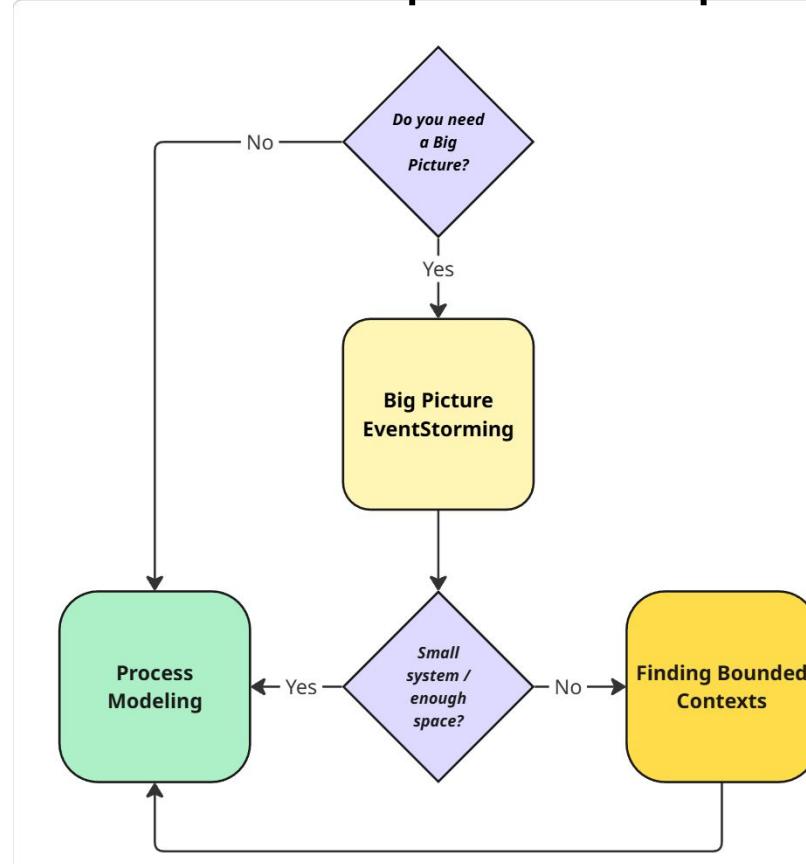
Main Elements of EventStorming



Main Elements of EventStorming (Trainitek's view)



Important note about the sequence of phases

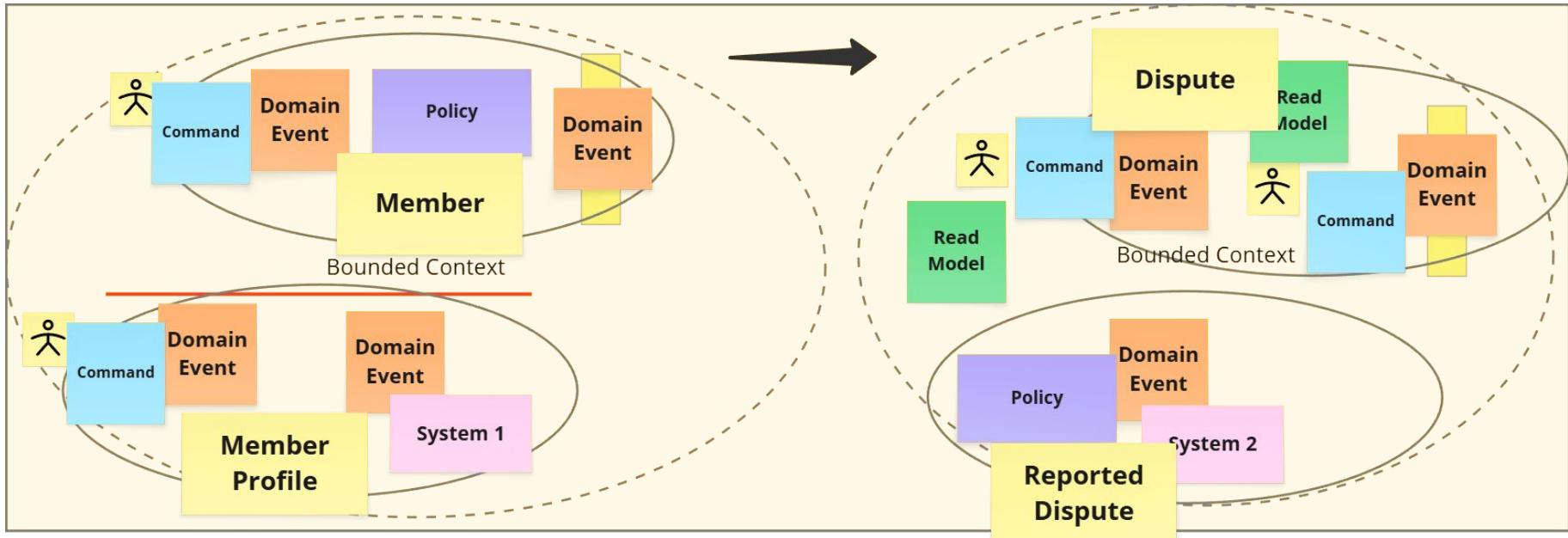


9/9 Discovering Aggregates

Discover Aggregates on a technical meeting. Vocabulary, related events, life-cycles, connected business rules are evaluated.



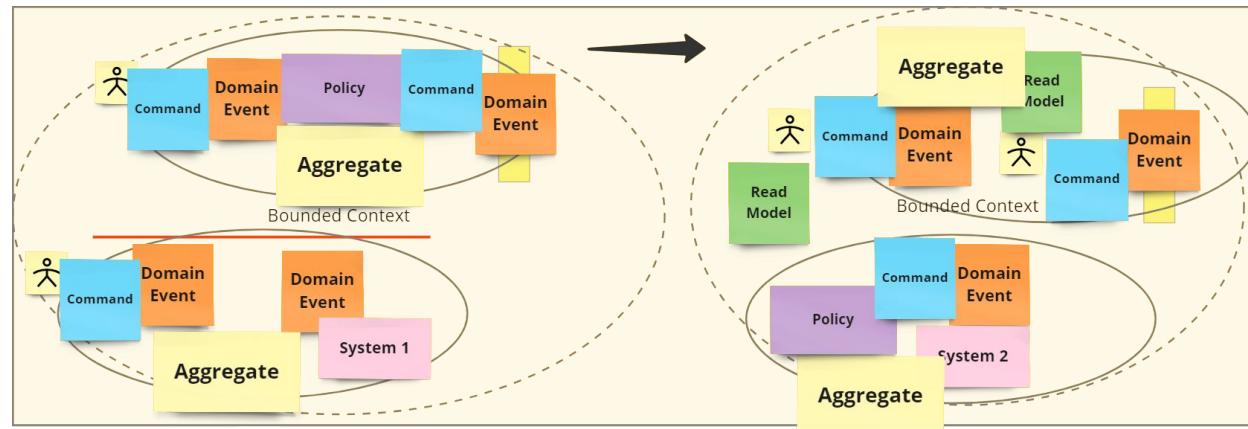
30-60 minutes



Why is it useful?

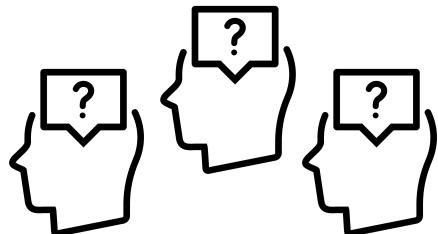
- Aggregates are the **most important classes** in your domain model

Tip: avoid identifying too big aggregates



That's it :)

However, you may still have the following questions...



Will EventStorming give me 100% accurate solution?

- Not always. EventStorming is not a Golden Hammer.
-  However, it's an **affordable good hammer** :)

Does EventStorming work only for new projects?

- EventStorming **can** also **help understand** the **current** systems and processes.

Is EventStorming about any hard rules?

- Not at all!
- Feel free to add e.g. read models during the Explicit Walk-Through phase, if needed
- As a Facilitator, you create an **illusion** that everything is under control, however, it's not :)

What if I need to remember additional info?

- **Use the board or big sticky notes** where you can write that info like:

order = [1..N] order
lines

order line = [1..M]
product lines

How to run EventStorming in my company?

- The very first suggestion is to “**start small**”. You will learn a lot during your first session.

When to “stop” EventStorming?

- Stop **when the value** of additional sessions **diminishes**, and revisit as needed to resolve outstanding issues
- EventStorming **is** a tool, **not a religion!**

Can I run EventStorming online?

- Yes, however, there are **some caveats** like:
 - Technical challenges
 - Distractions and multitasking
 - Interpersonal Interactions
 - etc.

OK, but what's next after EventStorming?

- **Digitize it** (photos, Miro, etc.) or keep the paper version
- **Document it** in a place accessible to the team
- **Plan technical-related work** (system architecture, backlog, etc.)

Will you say anything about EventStorming and AI today?

- Not this time :) I didn't find anything useful at the moment. If you found, please share!

[If we have time] Two **interesting** real-life **stories**

Conclusions

- (1/5) EventStorming is a **tool for exploration of complex domains and processes.**
- (2/5) We can call EventStorming “**the most effective meeting**”.
- (3/5) **Start “small”** if you decide to use it. Reading and watching is not enough.
- (4/5) **Don’t try to be precise, focus rather on learning.**
- (5/5) EventStorming is the **best starting point** we can get...
- ... in **hours**.

Thank You!

Presentation Slides



 vadim-prudnikov

