OST
Ostschweizer
Fachhochschule

Collaborative Modeling Lesson 3: EventStorming

ITBO Learning Lab 2, Initiative 1 (ZIOL, KAPS)

Spring Term 2025

Departement Informatik and ITBO

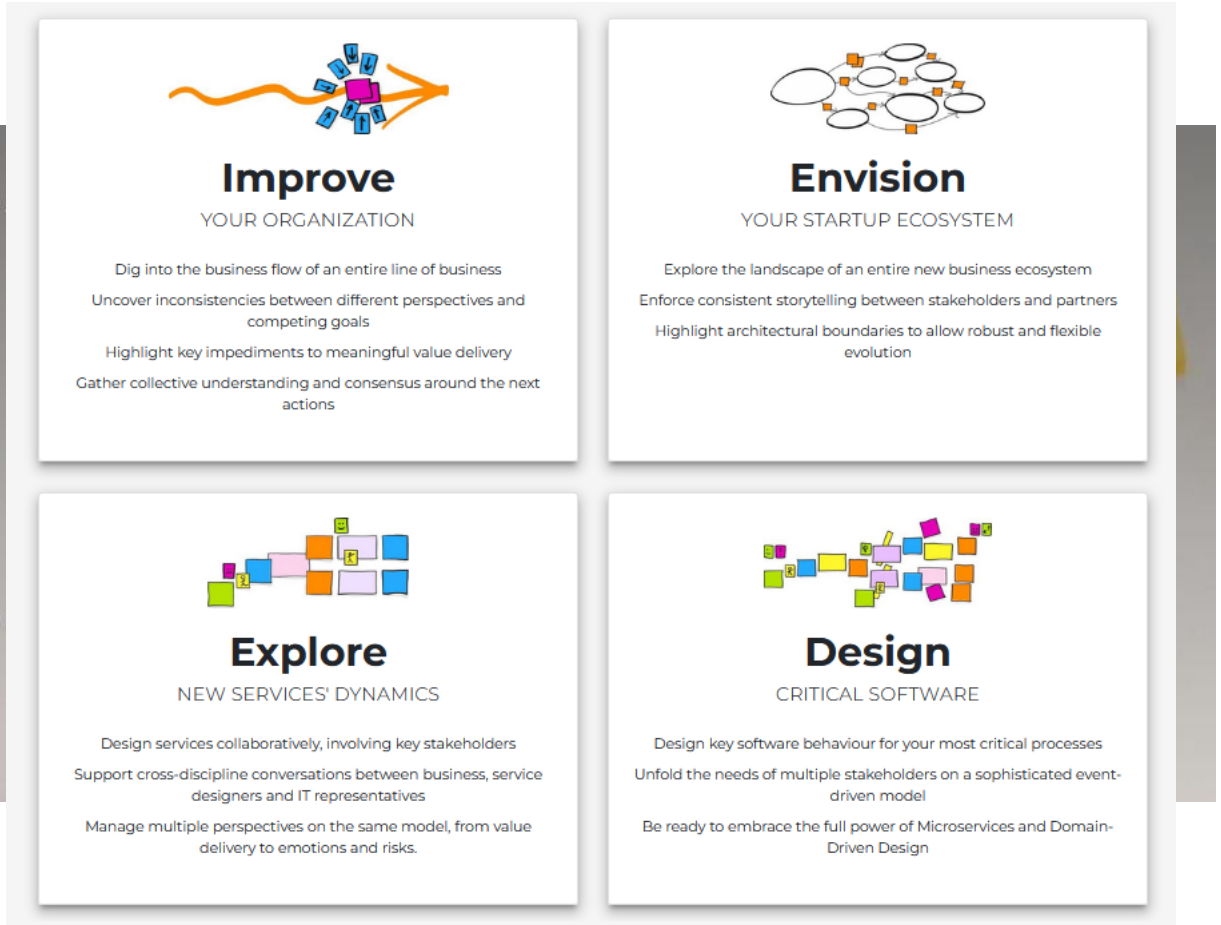
Agenda

- Motivation
- Concepts and activities
- Example
- Advantages and disadvantages
- When (not) to use

- Exercise
- Lab
- More information

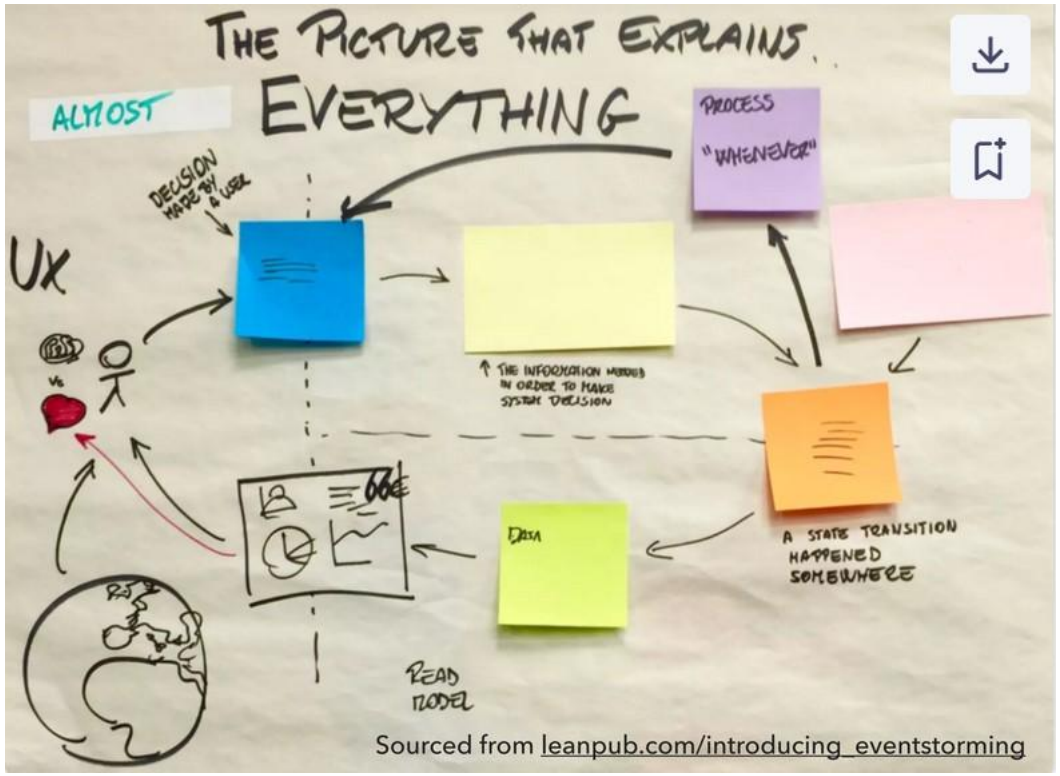
Context and Motivation

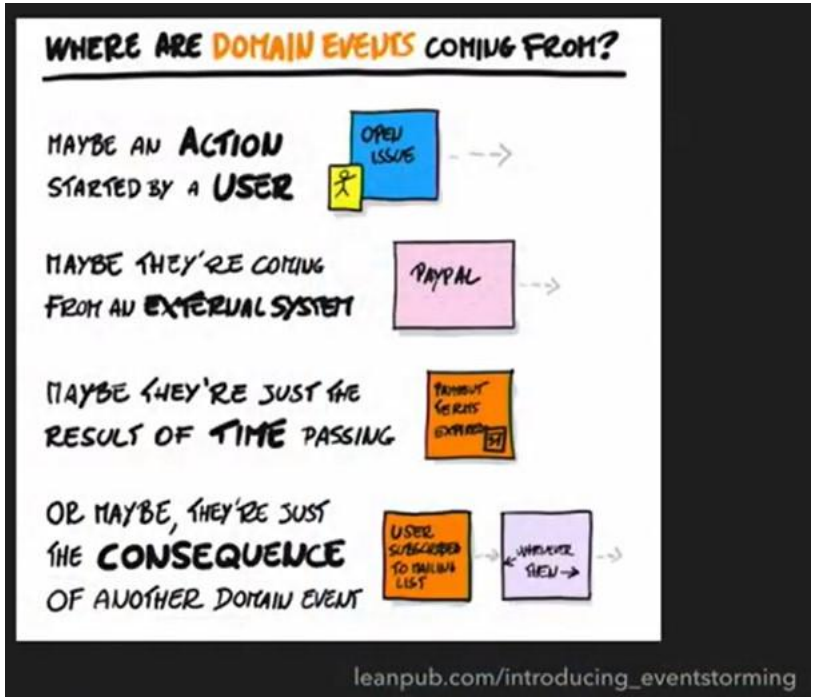
- https://www.eventstorming.com/



Concept Overview

- AgileDenver 2017 presentation by Paul Rayner





Domain Event: Essentials of Concept and Examples

- Something that **happened in the past** (verb in past tense)
- Something the **business cares about** (noun from ubiquitous language)

- More information:
  - Pattern definition in Evans/Vernon books
  - Explanation in Brandolini's LeanPub book
  - https://www.innoq.com/en/blog/2019/01/domain-events-versus-event-sourcing/#domainevents
  - https://contextmapper.org/docs/event-storming/

Three Types of EventStorming's

- **Big Picture**
  - Create a shared state of mind
  - Explore business/domain model
  - Identify Bounded Contexts
- **Process Modelling**
  - Assess specific process
  - Find bottlenecks and identify parts of the system to decouple from the existing software
- **Software Design**
  - Design clean and maintainable Event-Driven software
  - Derive shared language and (domain) model within a Bounded Context

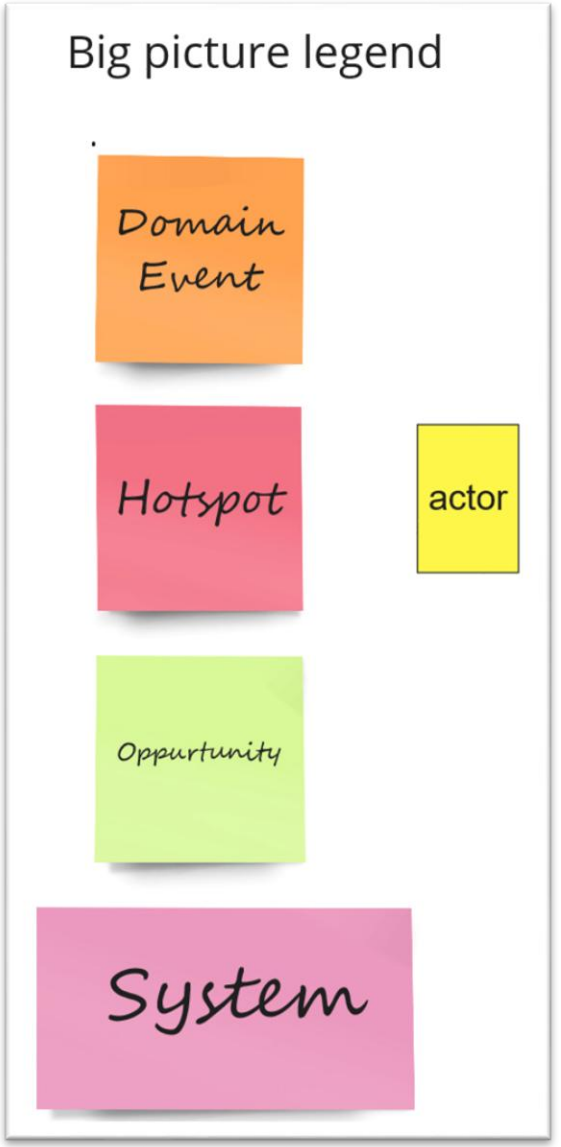


Image Source: DDD Crew, EventStorming Cheatsheet, github.com/ddd-crew/eventstorming-glossary-cheat-sheet

Concepts Overview for Process Modelling

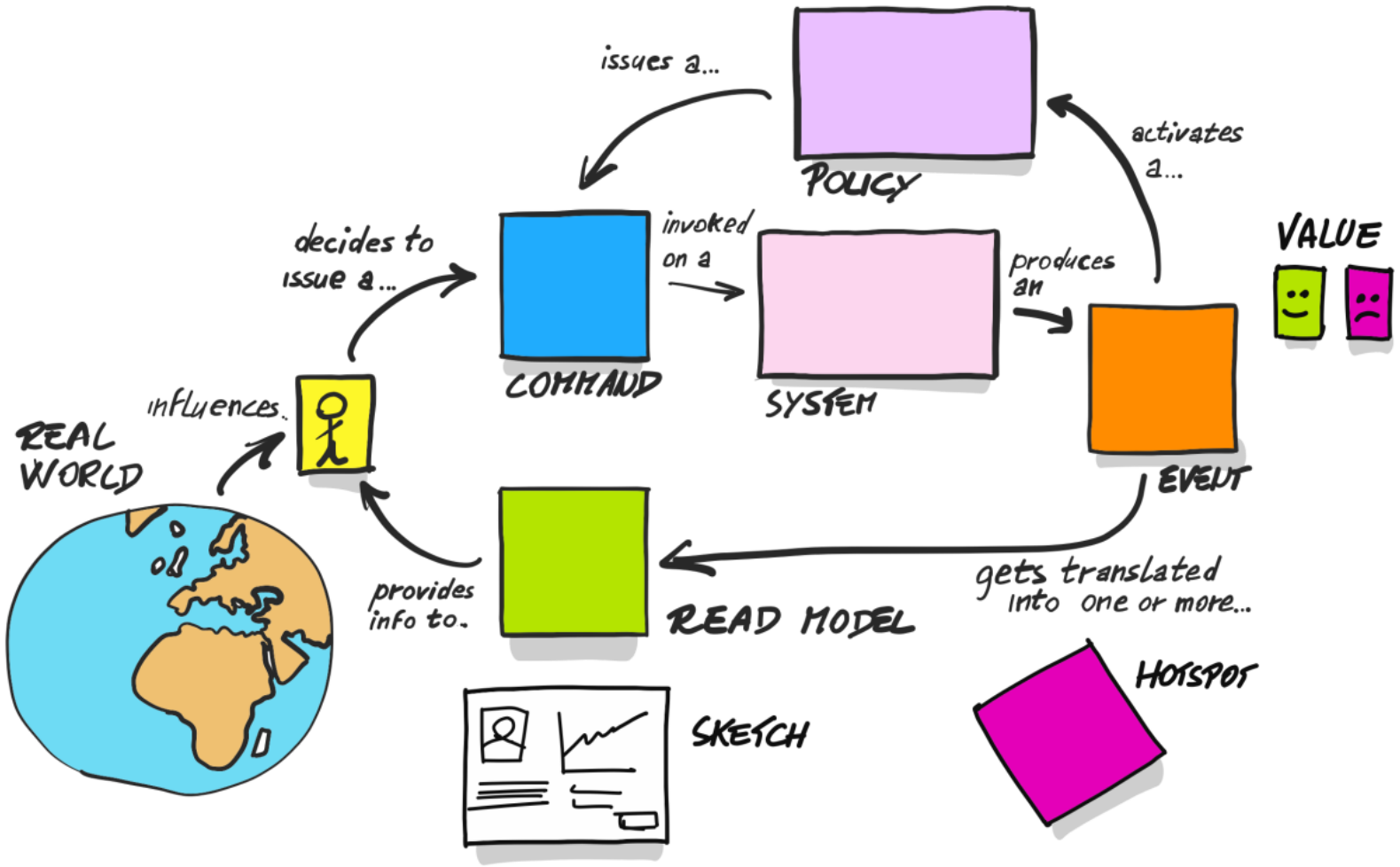


Image Source: Introducing EventStorming, Alberto Brandolini, leanpub.com/introducing_eventstorming

Design-Level EventStorming

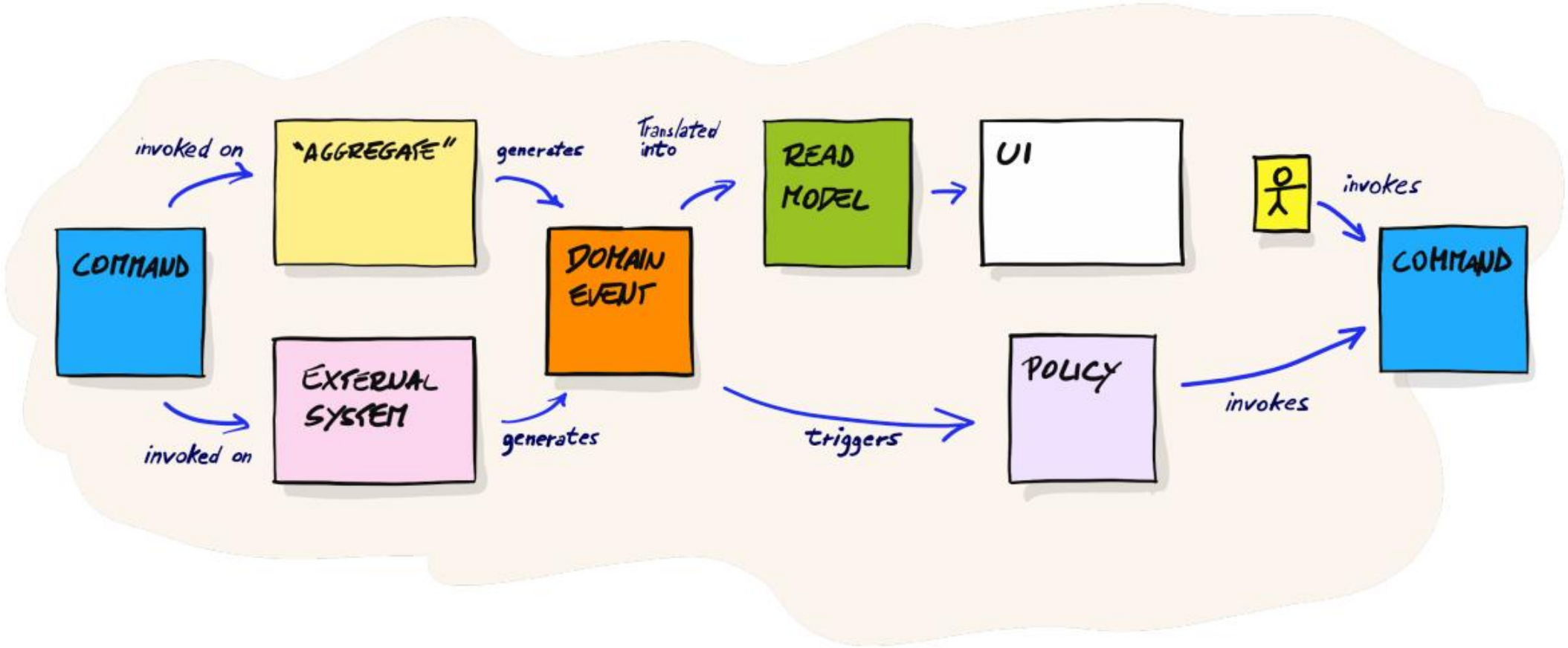


Image Source: Introducing EventStorming, Alberto Brandolini, leanpub.com/introducing_eventstorming

But: There are different «schools» …

- … and interpretations.
- DDD crew design-level notation:

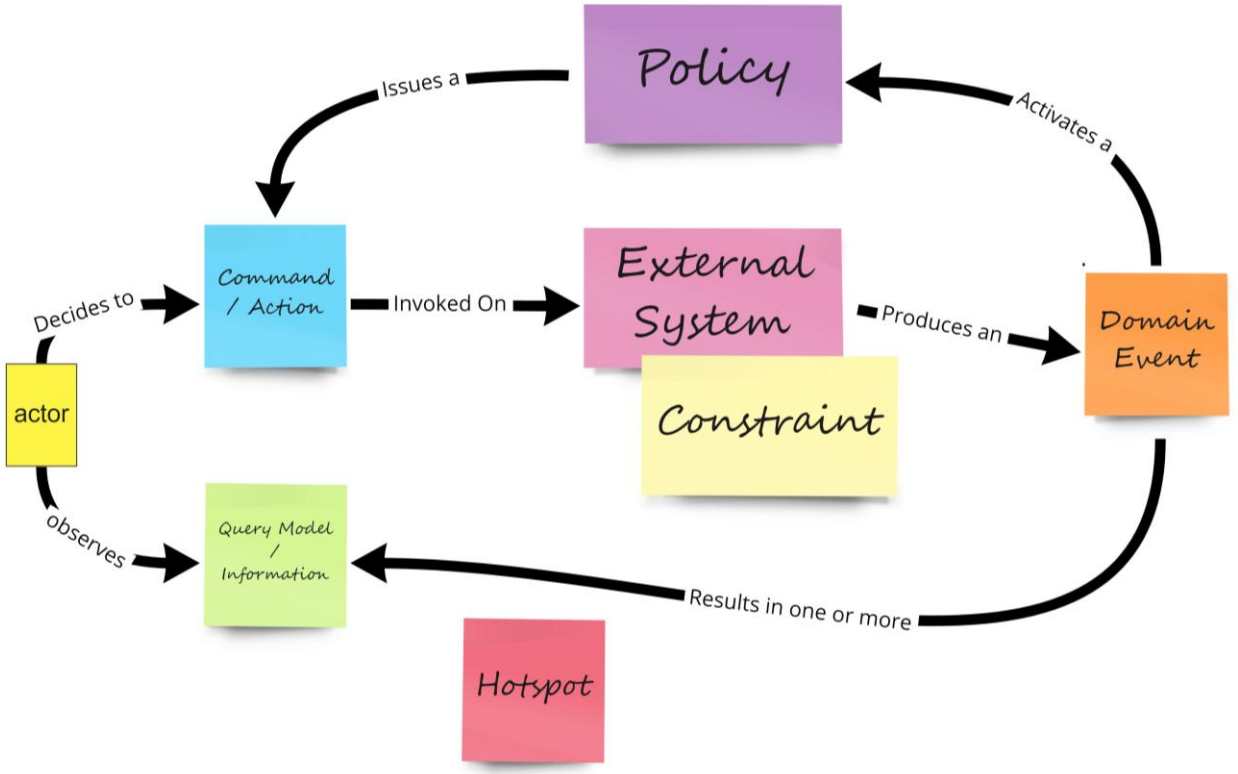


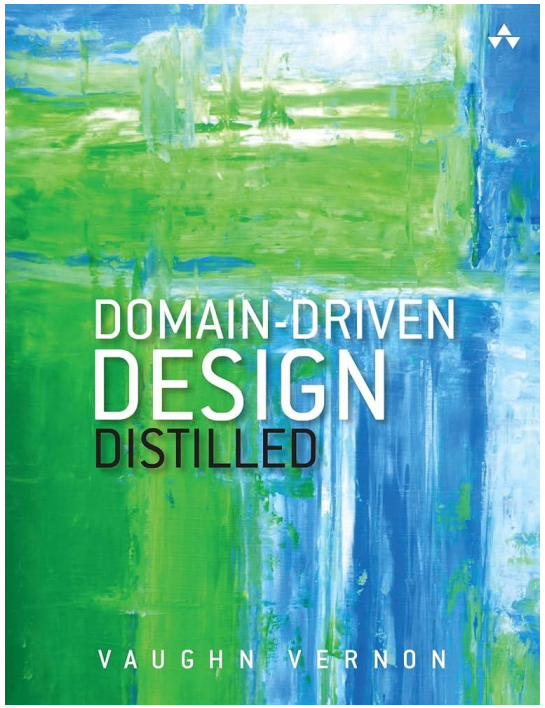
Image Source: DDD Crew, EventStorming Cheatsheet, github.com/ddd-crew/eventstorming-glossary-cheat-sheet

Recommendation

- Clarify the **purpose**
- Start **simple**
- Match scale to **scope**
  - Big Picture for broad organizational storytelling, Process/Design level for detailed workflows, and Software Design for implementation conversations.
- **Experiment** and iterate
- Lean on **facilitation**, not just technique
- Stay **pragmatic**
- **Blend** styles if needed

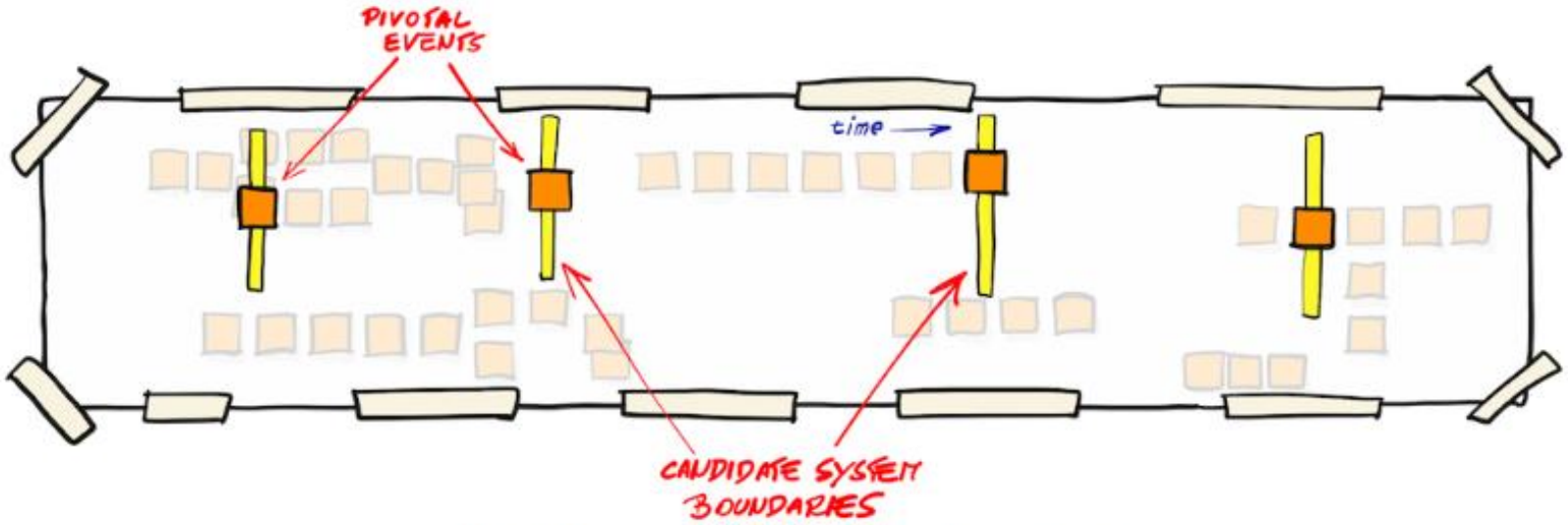
For our case study we will follow the steps recommended by Vaughn Vernon in his «DDD Distilled» book:

amazon.com/Domain-Driven-Design-Distilled-Vaughn-Vernon/dp/0134434420



Relation to DDD; and how to find Bounded Contexts

- One important concept to decompose your domain, are **pivotal events**:



Pivotal Events are a great source of information.

- **How to identify those?**

Image Source: Introducing EventStorming, Alberto Brandolini, leanpub.com/introducing_eventstorming

Heuristics for the Identification of Pivotal Events (by M. Plöd)

1. **Events That Trigger Significant Downstream Activity:** The "So What?" test.
2. **Events That Represent Key Business Decisions or Policy Enforcement:** Decision points and policy application.
3. **Events That Involve Hand-Overs to External Parties:** Boundary crossings and external triggers.
4. **Events That Lead to Lasting State Changes:** Significant data updates and process milestones.
5. **Events That Indicate Potential Bottlenecks or Failure Points:** Points of contention and exception handling.
6. **Compliance and Regulatory Significance:** Events tied to legal or regulatory requirements.
7. **Change in Resource Allocation:** Events causing significant shifts in personnel, time, or budget.
8. **Impact on Multiple Stakeholders:** Events affecting various departments or stakeholders.
9. **High Business Value Realization Potential:** Events that correlate with achieving key business objectives.

Source: Finding the Turning Points: Pivotal Events in Big Picture EventStorming, Michael Plöd, michael-ploed.com/blog/finding-the-turning-points-pivotal-events-in-big-picture-eventstorming

Application Example

- Paul Rayner, AgileDenver 2017
  - https://www.slideshare.net/slideshow/event-storming-76390807/76390807

BREAK

Pros and Cons

- Advantages (source: “DDD Distilled”, page 113):
  - Tactile approach, inclusive
  - Focus on events and business process for everybody (not on classes or the database)
  - Very visual, no formal notation but sticky notes (multiple colors)$
  - Fast and cheap to perform
  - Everybody learns something, breakthroughs in understanding possible
- Disadvantages, liabilities:
  - Not easy to schedule, busy stakeholders
  - Strong moderation required, risk of derailing
  - Results might get lost if not processed immediately (tool support?)
  - Introverted stakeholders might be reluctant to sketch on the board/wall

When (not) to Use, Alternatives

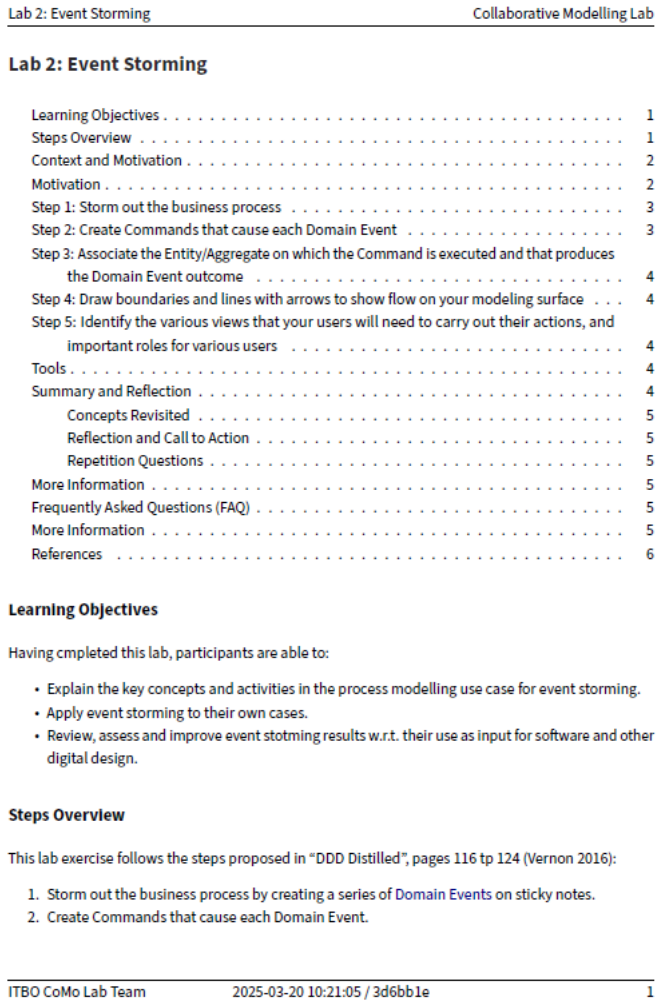
- When to use:
  - Big picture modeling, process modeling (analysis level)
  - Design-level modeling
- When not to use:
  - Other practices sufficient
  - Prerequisites cannot be met (e.g., availability of domain experts)
- Alternatives:
  - Domain storytelling
  - EPC modeling (BPMN?)
  - DPR practices

Critical Success Factors (Do's and Don'ts)

- Participant selection and scheduling
  - Hints: apply stakeholder elicitation techniques (aka interest-holders), create RACI matrix
- Real estate (storming infrastructure)
- Moderation, trust, workshop culture

Preview: Lab Exercise for This Lecture Lesson/Topic

1. Storm out the business process by creating a series of Domain Events on sticky notes.
2. Create Commands that cause each Domain Event.
3. Associate the Entity/Aggregate on which the Command is executed and that produces the Domain Event outcome.
4. Draw boundaries and lines with arrows to show flow on your modeling surface.
5. Identify the various views that your users will need to carry out their actions, and important roles for various users.



Lab 2: Event Storming — Collaborative Modelling Lab

Lab 2: Event Storming

Learning Objectives

Having cmpleted this lab, participants are able to:

- Explain the key concepts and activities in the process modelling use case for event storming.
- Apply event storming to their own cases.
- Review, assess and improve event stotming results w.r.t. their use as input for software and other digital design.

Steps Overview

This lab exercise follows the steps proposed in "DDD Distilled", pages 116 tp 124 (Vernon 2016):

1. Storm out the business process by creating a series of Domain Events on sticky notes.
2. Create Commands that cause each Domain Event.

ITBO CoMo Lab Team 2025-03-20 10:21:05 / 3d6bb1e 1

Preview: Sample Solution

OST

Discussion: Opinions, Tips and Tricks

- See boxes in Learning Lab PDF, example:

Hints: This step is described in (Vernon 2016). You might also want to consult the presentations by Paul Rayner (see "More Information").

- Focus on pivotal events
- Ask questions like "what's next?", "how do/did we get here?" (see FAQs and pointers at the end).
- Do not hesitate to move sticky notes around; no need to be correct or perfect in this step (and the following ones).
- Leave a lot of space between events because more sticky notes will appear when the next steps are performed! ::: info

Note to instructors: The book chapter provides very helpful "basic guidelines" for each step. You might want to use them to answers questions, to steer the group work, to check intermediate outcomes. We decided to leave out processes, which appear in the "DDD Distilled" book in the instructions here; if needed you can bring such modeling aspects in, for instance via red "hot spot" sticky notes calling for (sub-)process design.

Summary

- Three usage modes, always about dynamics
- Centered on domain event concept, several more concepts

- Inclusive, tactile, collaborative
- Needs careful preparation and execution
- Well-accepted and popular in industry

Resources (Links, Books, etc.)

- EventStorming: Collaborative Learning for Complex Domains, AgileDenver 2017 presentation by Paul Rayner, comprehensive introduction and application example
- EventStorming Workshop Cheat Sheet by Wolfgang Werner, quick introduction to the topic and workshop preparation

Questions (Repetition, Practice)

- What are the three use cases for EventStorming according to Brandolini?
- What domain makes event so interesting and relevant for analysis and design work?
- What can the event storming results be used for?

- Who should be invited?
- Which specification (of any kind) already exist?
- Who is in charge of workshop logistics and moderation (stickies, wallpaper, …)?

Strengths and Weaknesses of CoMo Practices

Practice	When to Use	Strengths	Weaknesses	What Space
Big Picture EventStorming	When modelling or designing an enterprise, business or domain	Adaptable and quick to learn, Chaotic nature gives a lot of insights	A lot of people in one room, requires experience of facilitators, Only works with a timeline	Problem space
Process Modelling EventStorming	When modelling or designing a story, process, or timeline		Difficult concepts to grasp, Can feel like a high time investment, Only works with a timeline	Problem space and solution space
Software Design EventStorming	When designing software for stakeholders needs			Solution space
Domain Storytelling	When modelling one specific scenario, process, or timeline	No learning curve, instant documentation	Structured approach that can lower the amount of discovery	Problem space and solution space

Source: *Collaborative Software Design*, Evelyn van Kelle, Gien Verschatse, Kenny Baas-Schwegler

Do you agree? 

Event Storming vs. Domain Storytelling (1/2)

- **EventStorming** seems to be better to:
  - Discover
  - Storm new ideas, be creative (due to chaotic nature)
  - model complex domains, where experts cannot yet concretely put the story into sentences
  - model processes -> strong time reference present
  - scales better with many people
- **Domain Storytelling** is in advantage, if:
  - Documentation is required (by using tools, you already have something «docs»-ready)
  - collaboration/communication between actors must be modelled
  - Workshop needs to be recorded (EventStorming is often too chaotic for that); works very good for distributed teams (online workshops)
  - the company culture prefers a more structured approach
- Both formats have in common that they help in **finding Bounded Contexts** for systems; although this might go faster with EventStorming ☺

Sources: - Event Storming und Domain Story Telling - Ein Vergleich, Johannes Seitz (innoq), innoq.com/de/blog/2019/02/vergleich-event-storming-und-domain-storytelling,
- *Domain Storytelling*, Stefan Hofer, Henning Schwentner

Event Storming vs. Domain Storytelling (2/2)

- Both focus on collaboration between domain experts
- EventStorming displays results on timeline (**what happen when?**),
  while Domain Storytelling represents the collaboration between actors (**who does what with whom?**).
- Important **difference in moderation**:
  - With Domain Storytelling, the moderator channels inputs and does the modelling
  - With EventStorming, the participants “storm” into the domain and are more active.

Source: *Domain Storytelling*, Stefan Hofer, Henning Schwentner

Combining DST and ES?

- Both can be used on different levels of detail; therefore there are **many ways for combination**
- Two examples:

- **From EventStorming to Domain Storytelling**
  - Get an overview of the domain with Big Picture EventStorming
  - Then develop Domain Stories for Bounded Contexts or specific (sub-) processes

- **From Domain Storytelling to EventStorming**
  - Create coarse grained domain story
  - Then go into details with Process or Design Level EventStorming

} as we applied it today … ;)

And User Story Mapping / Splitting?

- We would recommend it **after** DST and ES…
- … as soon as we have a common understanding about the domain and processes, we can **derive requirements** and user stories **from both**, **ES** and/or **DST**
- Key advantage: you get a **structured backlog** the team can work on
- This might be easier with DST, as you can almost read the user stories from the domain stories

- Additional **suggestion for combination**: (with ES and DST)
  - When stories need to get more detailed (as they need to be moved from product backlog to sprint backlog), use finegrained DST or design-level ES to refine how story should be implemented.

On the connection to DDD…

- All practices shown and applied today can help to discovery subdomains and Bounded Contexts (**Strategic DDD**)
  - With EventStorming use pivotal events as discussed
  - With Domain Storytelling use „grouping“ notation
- Both, EventStorming and Domain Storytelling, generate output for a domain/data model … and therefore help for **Tactic DDD** later
  - With EventStorming use „data“ („Aggregate“) stickies
  - With Domain Storytelling look at the modelled „work objects“
- Both, DST and ES, establish our „**Ubiquitous Language**“