**Event Storming**

<https://miro.com/app/board/uXjVPadv-p0=/>

The first step in arriving at the production of the ubiquitous language was to organize an event storming session. This was carried out on the collaborative platform Miro. Event Storming is an activity for a group of people **to brainstorm and rapidly model** a business process, a tactical tool for sharing business domain knowledge. The scope of the session is the business process that the group is interested in exploring. The base elements of the explorations are ***Domain events***, represented by sticky notes. Step by step, the model is enhanced with additional concepts— actors, commands, external systems, and others—until all of its elements tell the story of *how the business process works*.

We organized the exploration in four steps:

1. Collect Domain Events
2. Refine Domain events and create timelines
3. Track causes using additional concepts
4. Find aggregates and resort them

1.This has basically been **an unstructured exploration** in which we brainstormed the domain events related to the business domain that we were exploring. A domain event i**s something interesting that has happened in the business that must be formulated in the past tense**. At this early stage, we didn't have to worry about ordering events, or even about redundancy.

2. In this second phase we identified missing events and we put them in sequence. The order is the one in which domain events occur in the business domain. The events always start with the “happy path scenario”. In this phase we also highlighted pain points. These are points that caused trouble because of missing domain knowledge. We also defined Pivotal Events,  **significant business events indicating a change in context.** They are marked with a vertical bar dividing the events before and after the pivotal event. Pivotal events are particularly important because t**hey are indicators of potential bounded context boundaries**.

3. In this phase we identified what triggered the events, the users and the system involved. At the end of this phase. The main elements of this phase are:

-Commands: describes **what triggered the event or flow of events.** They are formulated in the imperative. If a particular command is executed by an actor in a specific role, the actor information is added to the command on a small yellow sticky note. Not all commands have an associated actor. Blue sticky notes

-Policies: a scenario in which **an event triggers the execution of a command**. A command is automatically executed when a specific event occurs. The decision criteria is explicitly written on the policy. Violet sticky notes.

-Read Model: the view of data within the domain that the actor uses to make a decision to execute a command. This can be one of the system’s screens, a report, a notification. Green sticky notes.

-External systems: any system that is not part of the business domain being explored.

By the end of this step, all commands should either be executed by actors, triggered by policies, or called by external systems.

4.We aggregated the domain events in sub domains.