What is Event Storming?

Event Storming (https://en.wikipedia.org/wiki/Event_storming) is a communicative brainstorming method in which knowledge and understanding of a specific, delimited field of knowledge (a domain of expertise) is jointly developed and visualised in a workshop. The starting point are so-called domain events.

General Information

(Setup for distributed event storming sessions)

People
6 to 10

Prep Time
20 min

Time
3 to 6 hours

Difficulty
Hard

Who shoud be involved?

It's important for an event storming workshop to have the right people present. This includes people who know the questions to ask (typically developers) and those who know the answers (domain experts, product owners).

1. People who know the questions to ask - Developers, Testers
2. Those who know the answers - Domain Experts, Product Owners, Clients
3. A good moderator and facilitator - Scrum Master, Agile Coach

Preparation

Make sure which area of knowledge you want to model or understand.
The more extensive the business process, the more time this method takes.

Think about whether you can divide the knowledge area into smaller business processes, which you model one after the other in several event stromings.
Example:
Knowledge area - *online shop*
Business processes - *shop library/offer, order process, etc.*

If you are the moderator, make sure that you have already dealt with moderating an event storming session in advance. There are some training videos on youtube for this purpose.

Terminology

Domain Event
An event that occurs in the business process. Written in past tense.
Domain Event

User/Actor
A person who executes a command through a view.
User/ Actor

Business Process
Processes a command according to business rules and logic. Creates one or more domain events.
Business Process

Command
A command executed by a user through a view on an aggregate that results in the creation of a domain event.
Comm-and

Aggregate
Cluster of domain objects that can be treated as a single unit.
Aggre-gate

External System
A third-party service provider such as a payment gateway or shipping company.
External System

View / Read Model
A view that users interact with to carry out a task in the system.
View/ Read Model

Question Marks / Risks
Use red Post-Its for unclear topics or questions that arise during the session.
Question / Risk

Step 1 - Collect Domain Events (Big Picture)

Time
30 to 45 min

Each participant uses only orange Post-Its in the first round.
Each orange Post-it stands for a professional event. A professional event is a technical relevant fact that happened in the course of business. The verb on the Post-it must therefore be in the past.

The first round is a pure brain storming process about the existing domain events. Ask people to hang the events in the chronological order in which they occur.

Step 2 - Refine Domain Events (Big Picture)

Time
30 to 45 min

Go through the Domain Event Post-Its with the participants. Ask participants to explain what each event means. Check for syntactical correctness.

Also discuss again whether the events are in the right order in terms of time. Unify occurring synonyms (different terms for the same thing) and sharpen differences if the same term was used to describe different things.

Step 3 - Track causes (Process Modelling)

Time
1 to 3 hours

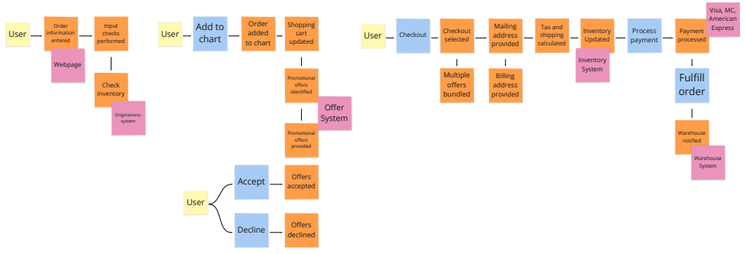
Get into the cause study now. Where do the professional events come from?
There are four main causes:
user actions (Commands, User/Actors, View/Read Mode),
external systems,
time (e.g. appointment elapsed) (Business Process) and
other domain events (through automatic reactions).
Ask about the triggers of the events.

Step 4 - Re-sorting & result (Software Modelling)

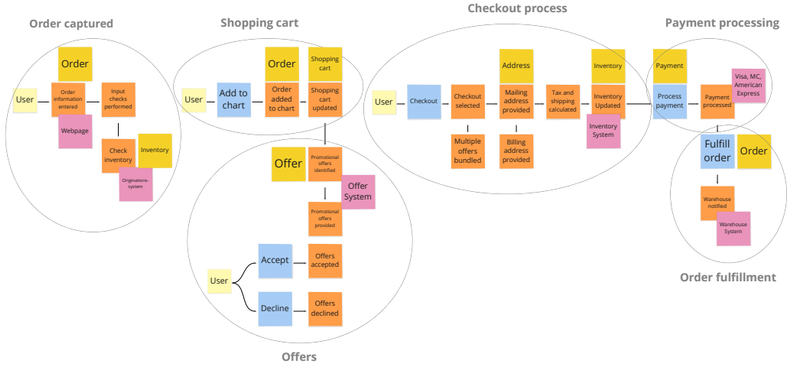
Time
60 to 90 min

Now it's time to remove the post-its from the timeline and group them around the aggregates found. This makes following new relationships clear:
Which users trigger which commands?
Which commands affect which aggregates or external systems and trigger which changes?
Which aggregates or external systems trigger which events during command processing?
Which events trigger which policies?
Which events create which read models for which use cases?
Which policies call up which new commands?

Example output after identifying missing events (outlined in red) after putting them in sequence



Example output after identifying what triggers the events, the users and systems involved



Example output with aggregates added in and bounded context applied