Domain Driven Design

THEORY AND HANDS ON ,

MAHMOUD FADDO

Course Agenda

- Why Domain Driven design?
- Language and context
- Event storming
- Designing the model
- Implementing the model
- Acting with commands
- Consistency Boundary
- Aggregate persistence
- And much More ..!! (keep you excited)

second session agenda

- Why Domain Driven design? (Done)
- Language and context
- Event storming
- Designing the model
- Implementing the model
- Acting with commands
- Consistency Boundary
- Aggregate persistence
- CQRS
- And much More ..!! (keep you excited)

Why Event Storming

- Discovering domain terminology is essential, and this terminology becomes a part of the Ubiquitous Language.
- However, the process of discovery can be rather lengthy and not always successful.

What is Event Storming

- we want to improve our domain knowledge by talking directly to domain experts and organizing a workshop or meeting with them.
- There are a few fundamental issues we need to be solved here:
  - Provide visibility during the discussion.
  - Have a modeling language that people understand.
  - Involve many people simultaneously.
- Find a way to express terms, behavior, model processes, and decisions, not features and data. Find a way to express terms, behavior, model processes, and decisions, not
- features and data.

Modeling Language

- The basic idea behind EventStorming is that it gives a straightforward modeling notation that is used to visualize the behavior of the system in a way that everyone can understand.
- This approach creates visibility, increases engagement, and involves people who would otherwise be anxious about any participation in a modeling session at all,

Modeling Language

- Event storming is a language for modeling anyone can understand.
- It describes system behavior , not data.
- Event storming Goal is finding out how the business work

Modeling Language system state

- We need to think of system as state machine , this state will be changed when an event happen , those events are called domain events.
- Example : in E-commerce system user can make invoice then pay invoice then company will ship that invoice.
  - Lets extract system events and deal with system as state machine
    - E1 : invoice initiated by user.
    - E2 :invoice paid.
    - E3 : invoice shipped
    - … and so on.

system state scientific words:

- We can populate that each system at any given moment of time is found in particular state.
- This state can be changed when actor interact with the system.
- Jump to another example :
  - Pay invoice online: user will select the payee third party , enter money then confirm the payment. So from person point of view.
  - We can see each action made a state transition like , the payment order was created and signed, the amount was added to the payees account , the amount was deducted from the payers account, the bill was marked as paid.

Facts

- From previous example we can find out that all events became facts of life.
- So , facts are domain events.
- As Domain Events are facts , we can grasp them easily. because anyone can understand facts.
- Facts are something happened not something that some one want to do , not a feature.
- Domain event ⬄ facts.

Extract domain events

- Events must have subject (noun) and predicate (verb)
- Verb must be in past
- Follows time line (ordered , parallel).

Visualization

- How to visualize :
  - The visualization is simple we only need white board, colored sticky notes and black marker .
- Why visualization :
  - When people see what is considered to be the whole picture, some might start asking *what if* questions. *What if* there is not enough money in the account? *What if* the bill reference number is wrong? *What if* the payee account is not correct? What if, what if, what if? It then appears that our simple process is not that simple at the end of the day
  - **WYSIATI** (short for **What You See Is All There Is**)? We base our initial understanding on a simplified view of the world.

Event storming workshop

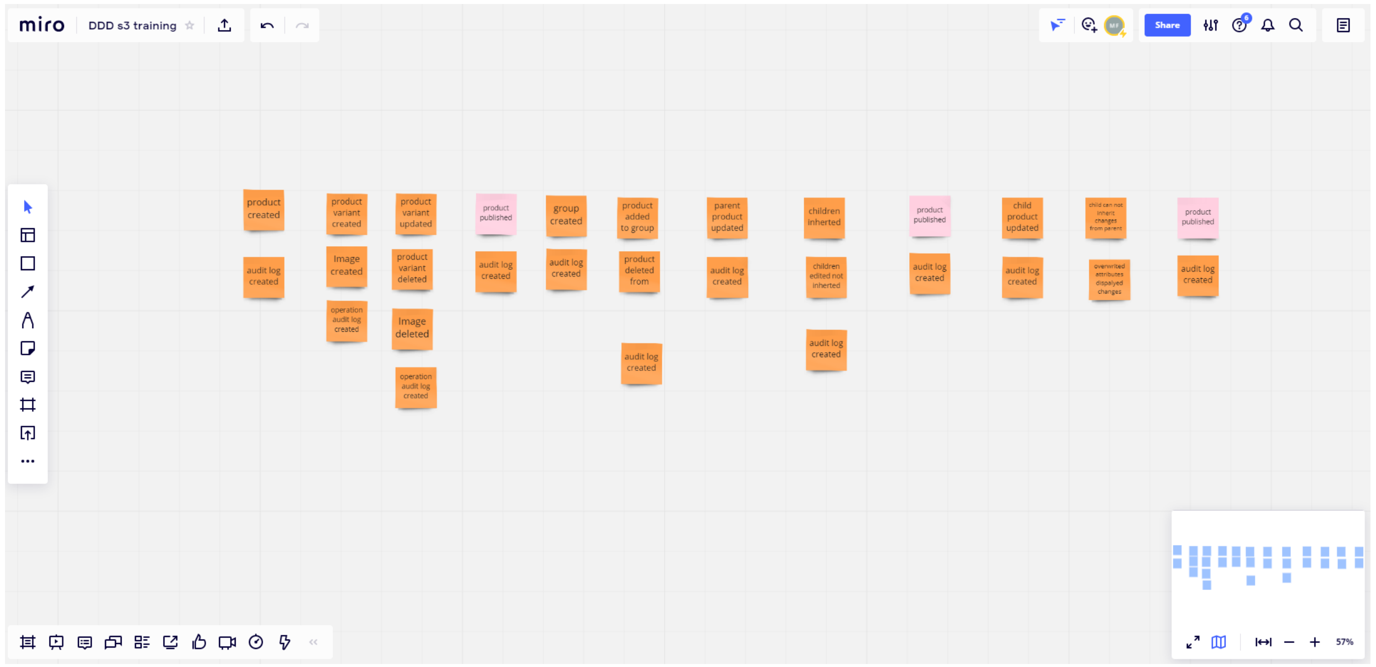
- Who to invite :
  - Domain experts (expert from each department)
  - Developers
  - Testers
  - UX
- Prepare the space:
  - As we discuss we need a white board , but what if there is no enough space while we are modeling?? Human brain tends to treat the space left as a sacred resource. It becomes precious, and people start saving space. Some events become **not important** and therefore not put on the whiteboard. Some ideas become secondary and not worth looking at.
  - Always remember   are very precious.
  - We need to replace the white board with cartoon board can be hanged on the wall and extend the as we need.

Event storming workshop

- Materials:
  - A lot of Sticky notes :
    - Orange for events
    - Red for hot spot
    - Pink for external systems
  - A lot of black marker
  - A lot if cartoons.
- Time and scheduling
  - We should schedule event storming session not more than 2 hours per session.

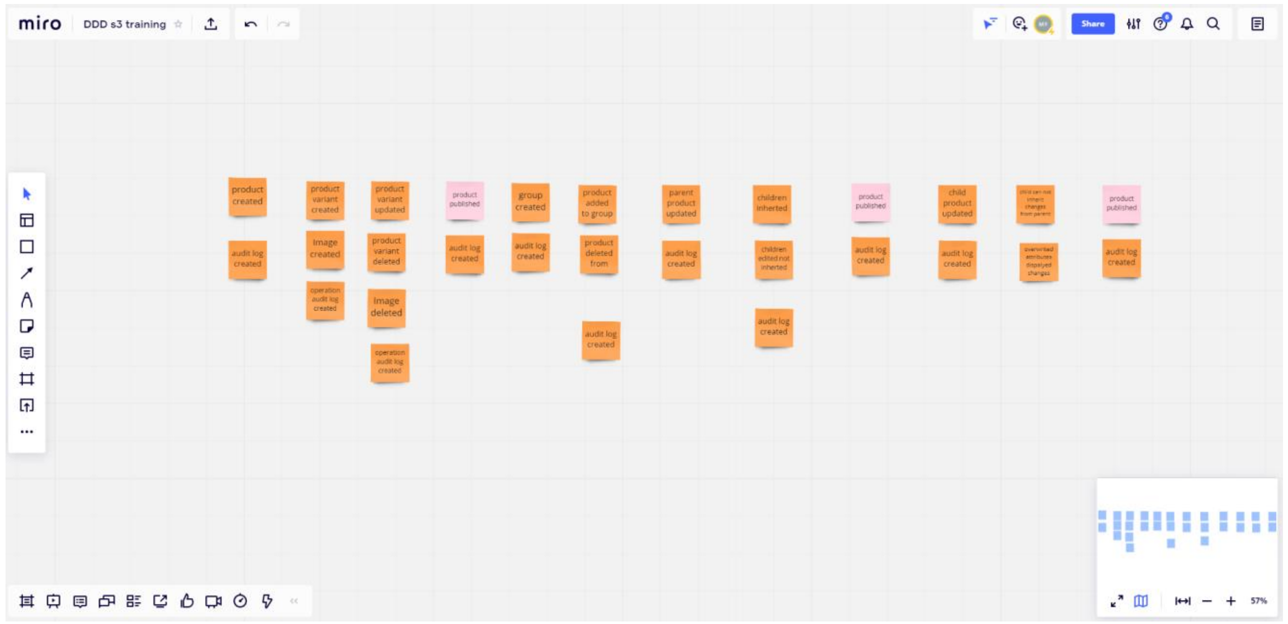
Our first model:

- Lets make our first model based on our project ***<u>Product manager =D.</u>***
- ***<u>Lets try to model product with domain experts.</u>***
- ***<u>Offcourse you can not see any thing from below image :V.</u>***



Our first model: after what if

- I made a startup modeling for product.
- ***<u>So the big picture completed. The domain experts will start to say what if and you will find out by your self.</u>***
- ***<u>Lets apply a life workshop together and make what if questions.</u>***



What is next !! Keep excited

Designing the model with more Event storming techniques.

Hands on

- Your turn:
  - ***For who interested in increase his skills on event storming Kindly you are requested to make an event storming modeling for Matrix payment cycle.***
  - ***The Event storming should be done by all of you together.***
  - ***Use this app to do the Event storming workshop :***
    - *https://miro.com/*
  - ***Invite me to workshop after you done all events storming.***

Thank you

SESSION 3