



Event Storming In Real Life

Jessé Moraes Braga

Jessé

 Full-stack Developer
@ iits consulting

Passionate about travelling and
cycling

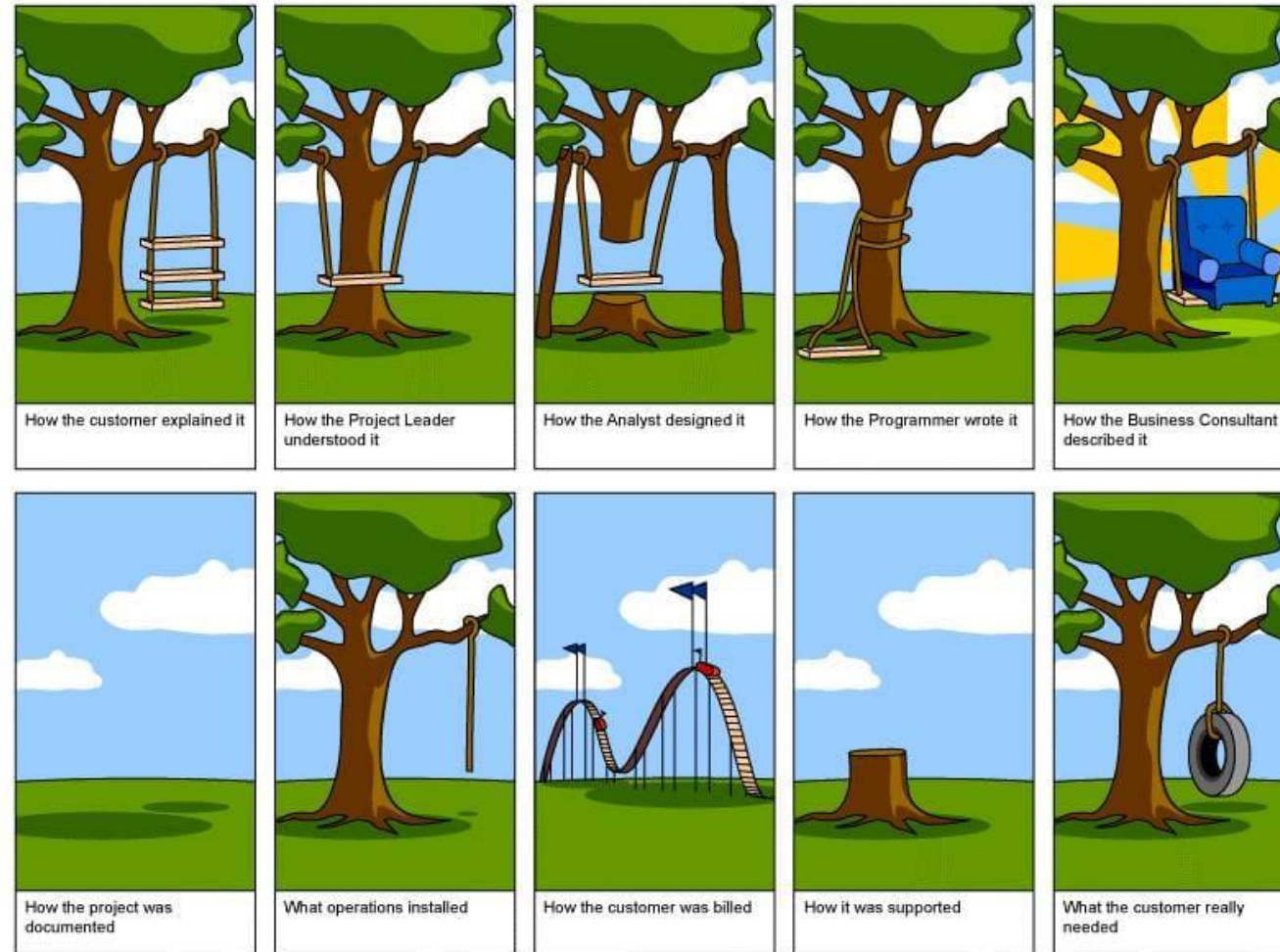


Opportunity

- January 2023
- New greenfield project
- Complete technical freedom



Every Software Project Ever



Naturally, we thought about...

UI-Libs Microservices Self-Contained Systems

Ports/Adapter Architecture CI/CD Monorepo Spring Boot

JS-Frameworks Build System Cloud Hexagonal Architecture

k8s Micro Front Ends Cluster

Authentication Layered Architecture

IAM/ IdP

But how do we make decisions?

Series of **pitfalls** for a greenfield project

- User **domain** completely **unknown**
- Stakeholders think in their own **silo**
- No ubiquitous/ **common** language
- Internal **politics**
- Conflict of interests → what is the actual **goal**?
- Sticking to **old** workflows, systems, processes

And none of that requires a technical solution

But how do we make decisions?

“It’s not the domain expert’s **knowledge** that goes into production. It is the **assumption** of the developers that goes into production.”

Alberto Brandolini



What if...

we found out what the user really needs?



Get inspired



GETTING MODULES RIGHT WITH DOMAIN- DRIVEN DESIGN

MICHAEL PLÖD - INNOQ

Domain-driven Design helps teams achieve a better alignment between the business and the technical architecture in order to design applications that have highly expressive and maintainable domain models. This talk aims at giving you an overview of Domain-driven Design and how the ideas behind it help you to create better modular applications.

We will talk about aspects from strategic Domain-driven Design such as Bounded Contexts and Subdomains, in addition to that the talk will explain the most important patterns from the tactical part of Domain-driven Design (Aggregate, Entity, Value Object). Finally you will learn about methods that help you in getting a better understanding about the domain you are working in.



Get inspired



100.

000

AN
EVENT
STORYING
TALE

ORANGE STICKIES

LATER

@ziobrando

6-8 NOV 2019
Alberto Brandolini
100,000 Orange Stickies Later

Scandinavian Developer Conference
Malmö, Sweden
oredev.org

https://www.youtube.com/watch?v=fGm62ra_mQ8

What is Event Storming?

“EventStorming is a **flexible** workshop format for **collaborative exploration** of **complex** business domains.”



eventstorming.com

Alberto Brandolini

Initial Challenges

Getting representatives of every part of the process together

- “Why? We never needed that”
- “Isn’t management enough?”



Initial Challenges

Scheduling a two-day workshop with many people on short notice

- Getting to know the team: management, users, devs, PO, etc



The Workshop



Everyone showed up

- At least 1 person from each area

And they were willing to get work done

We aligned expectations

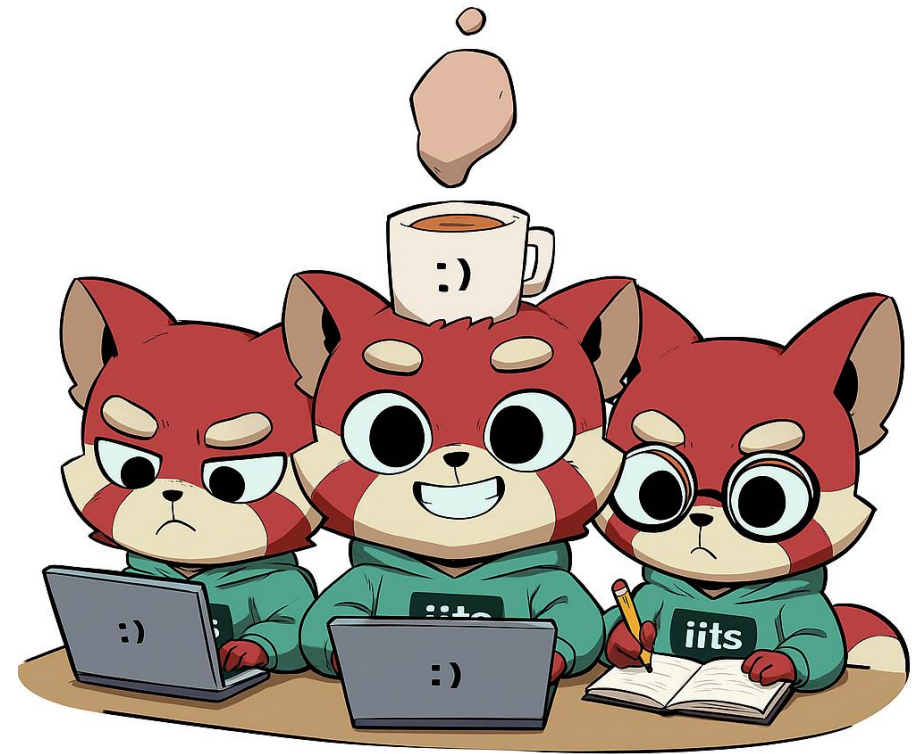
- For the Workshop
- For working together

Step 1: Identify Business Processes

Mapped all main business processes through events*

- Core functionality
- Find parallel operations
- Pinpoint synergies

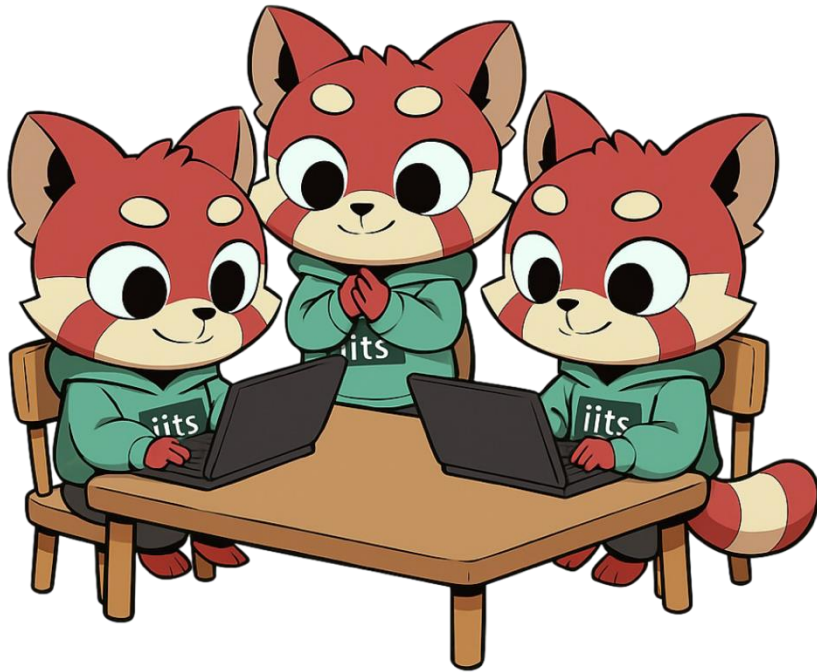
*orange post-its



Step 1: Stakeholders explain their business processes



Step 2: Identify Hotspots



What are possible problem areas*?

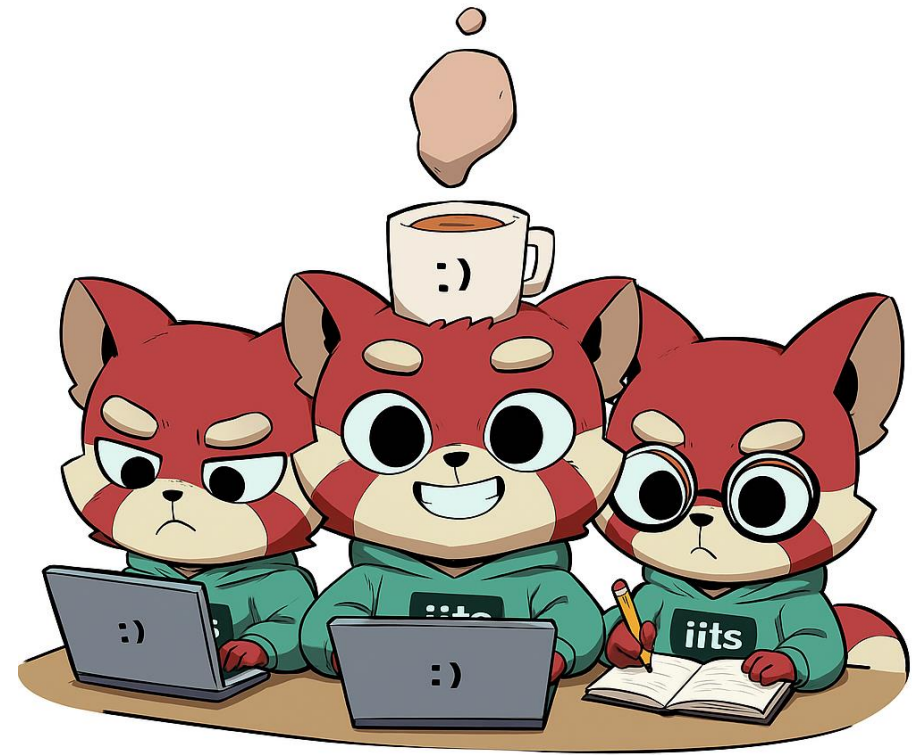
- What do we need to clarify?
- Hotspots are especially relevant in code

*purple post-its

Step 3: Discover the Actors

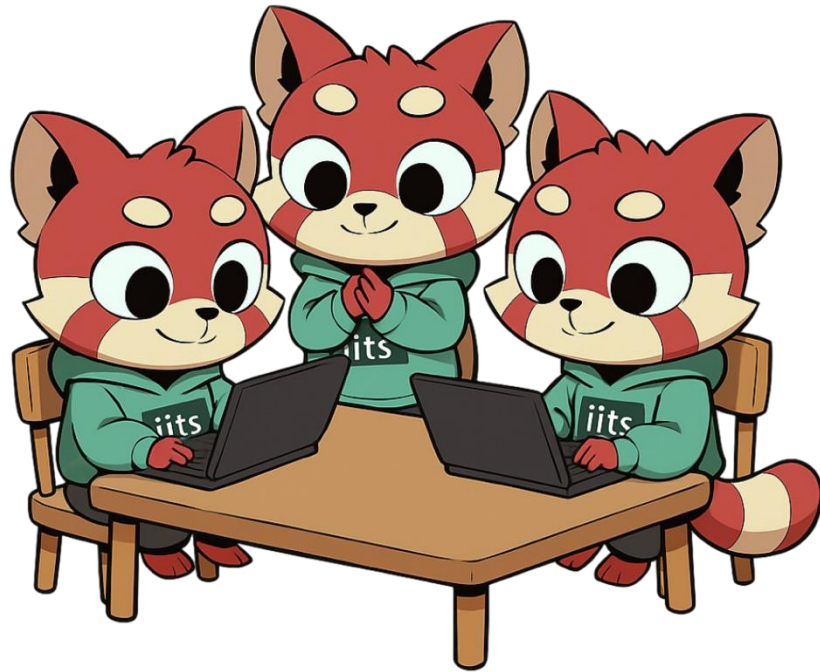
Who* acts on/ triggers those events?

- Are they our core users?
- Do they work in other areas?



*usual "yellow-ish" post-its

Step 4: External Systems



Which systems* do we have to communicate with?

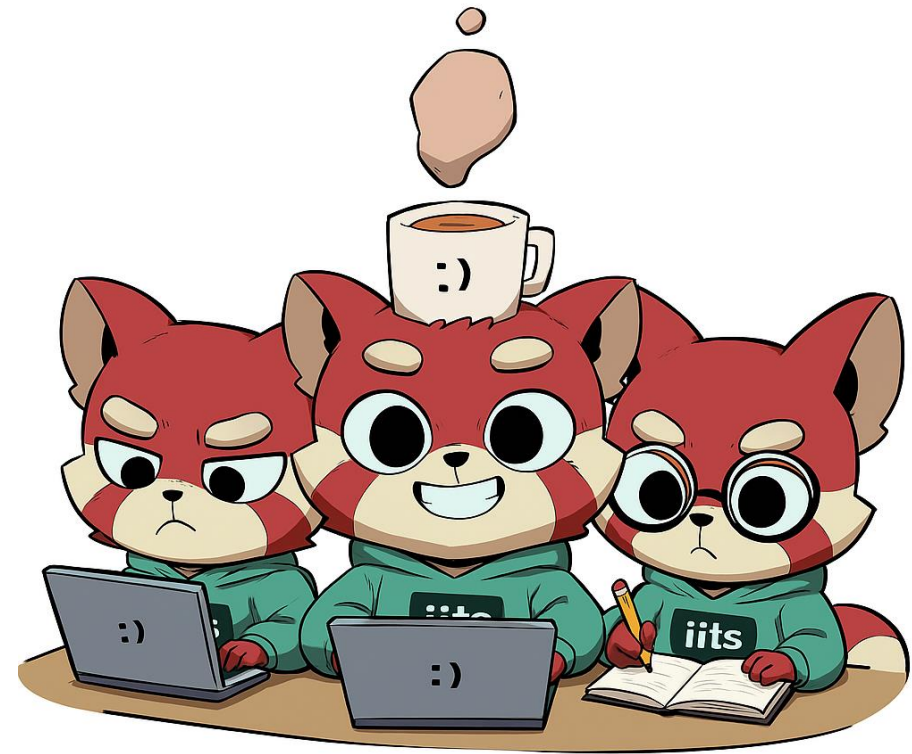
- How do we access data?
- Which system is the single point of truth?
- How do we transform data?

*green post-its

Step 5: Pivotal Events

Pivotal Events

- Most significant events
- Important to many stakeholders
- Candidates for system boundaries
- Hints at bounded contexts



Step 6: Prioritize



This isn't exactly *kosher*, but it's a flexible method, right?

Voting with blue dots on the most important parts of the software

We were able to identify what is important for each area and what is important for everyone

The Workshop



What did we do?

Stakeholders discussed their complex domain in front of us devs for two days

We listened, we asked questions, we learnt the business logic

We talked in a **common** language



Business Results



Business understood their own processes

Mistakes were corrected

Teams with different takes of the same process aligned

Our Results

They **thanked** us for helping them map their processes

- We found out it was 3+ years in the making

They **communicated** their needs

- To one another and to us!

We **understood** the assignment

- There was no “assumption” anymore



And then there was only



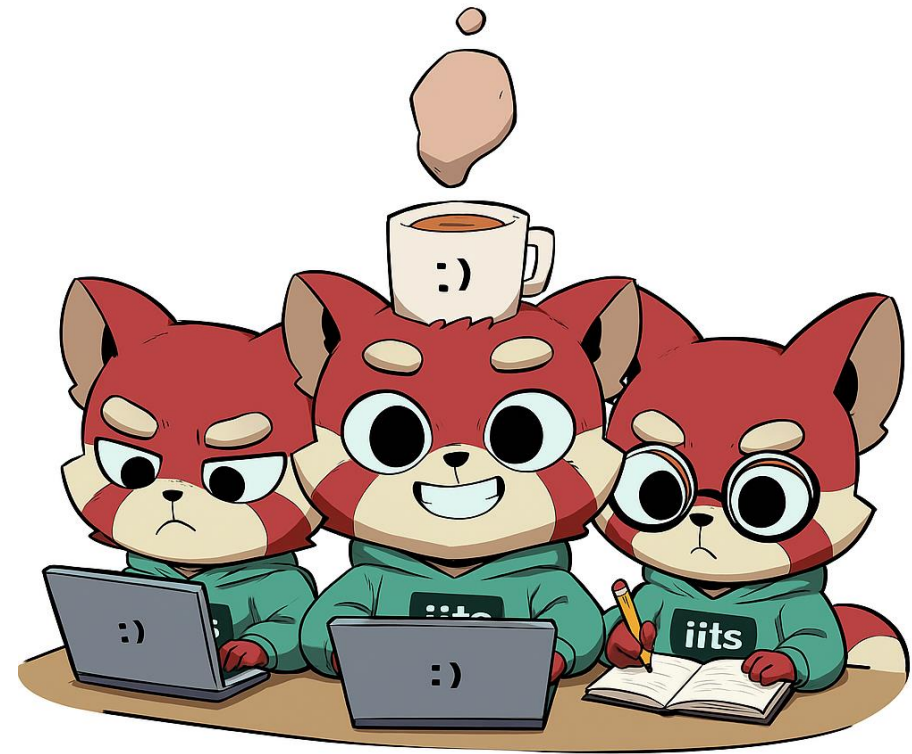
trust



Bounded Contexts

Bounded Contexts

- Cluster of domain events that are naturally grouped together
- Encapsulate domain logic
- Are autonomous and clear
- May only be accessed through a well-defined API



What about the technical part?

Self Contained Systems

Bounded Contexts

Micro Frontends

Detached UI-Library

Joined all together in a Monorepo

Monolithical Backend/ Frontend

Next: maybe Moduliths?

Key Takeaways



In an “AI-World” what we devs do is to understand [complex] business processes and turn it into code

To minimize risks, we must understand the full picture, and remove silos

Do not pull apart what naturally belongs together

Key Takeaways



Conway's Law

- Can we break it? How?

Trust is priceless

- Challenging situations will arise and if the customer trusts you, it will all be fine

Code vs. Expertise?

- What is more important?

Questions?



Resources

<https://www.eventstorming.com/>

https://leanpub.com/introducing_eventstorming

https://leanpub.com/ddd_first_15_years

<https://github.com/ddd-crew/eventstorming-glossary-cheat-sheet?tab=readme-ov-file>

<https://github.com/ddd-crew>

Thanks!

