Developer to CTO

CS 7002

Romain Untereiner

Manage tech

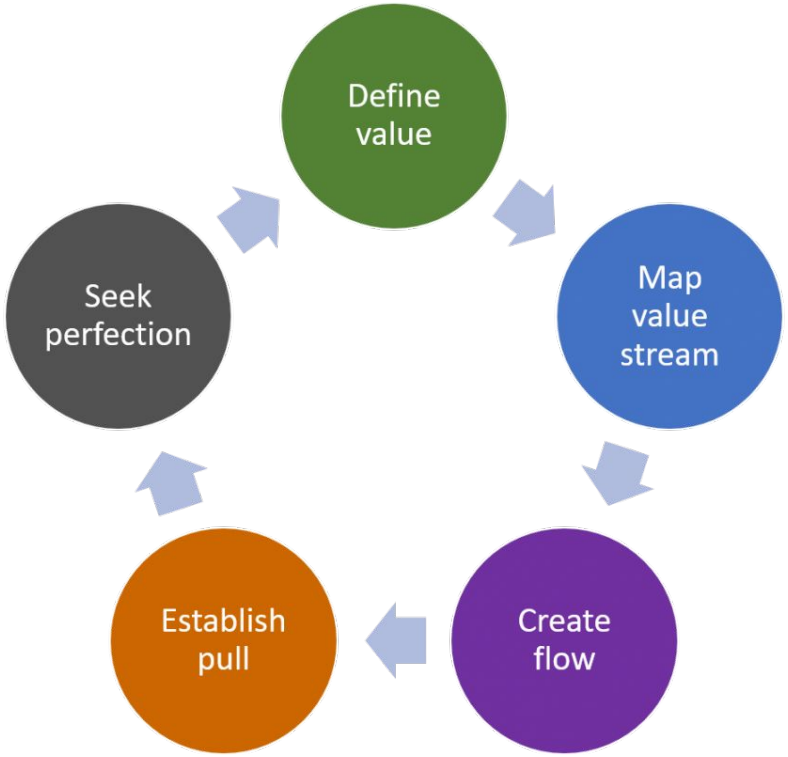
Lean

How to **organize** human activities to **deliver more benefits** to society and **value** to individuals **while eliminating waste**

- Value
- Value streams
- Flow
- Pull
- Perfection

Comes from Lean Manufacturing

Lean


Define value
Map value stream
Create flow
Establish pull
Seek perfection

Lean Startup

Create a startup with **minimal waste**

→ Check Ash Maurya's books

Agile software development

Philosophy encouraging **discovering requirements** and **developing solutions** through the **collaborative effort** of **self-organizing** and **cross-functional** teams and their stakeholders

Manifesto values:

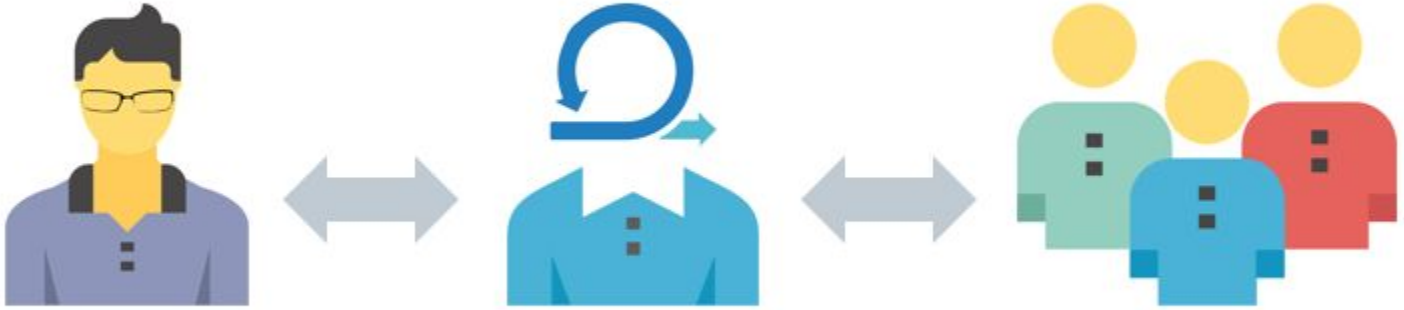
- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

SCRUM

A **lightweight**, **iterative** and **incremental** framework for **managing complex work**.

- Small teams
- Iterative process (comes from Agile)
- Empirical
- Joins Product and tech

SCRUM roles



Product Owner:
Owns "what" is desired
And "why" it's desired

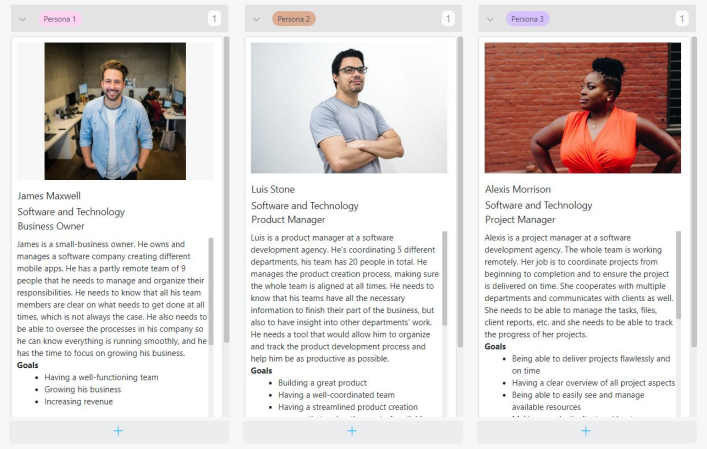
Scrum Master:
Keeper of Scrum
Process, facilitator

Scrum Team Members:
Owns "how" and "how quickly"
work is delivered

SCRUM concepts

Persona

A fictional character created to represent a user type that might use a site, brand, or product in a similar way



Developer to CTO - Romain Untereiner - 2021

SCRUM concepts

Story

It is a natural language description of features of a software system.

“As <**who**> <**when**> <**where**>, I <**want**> because <**why**>”

SCRUM concepts

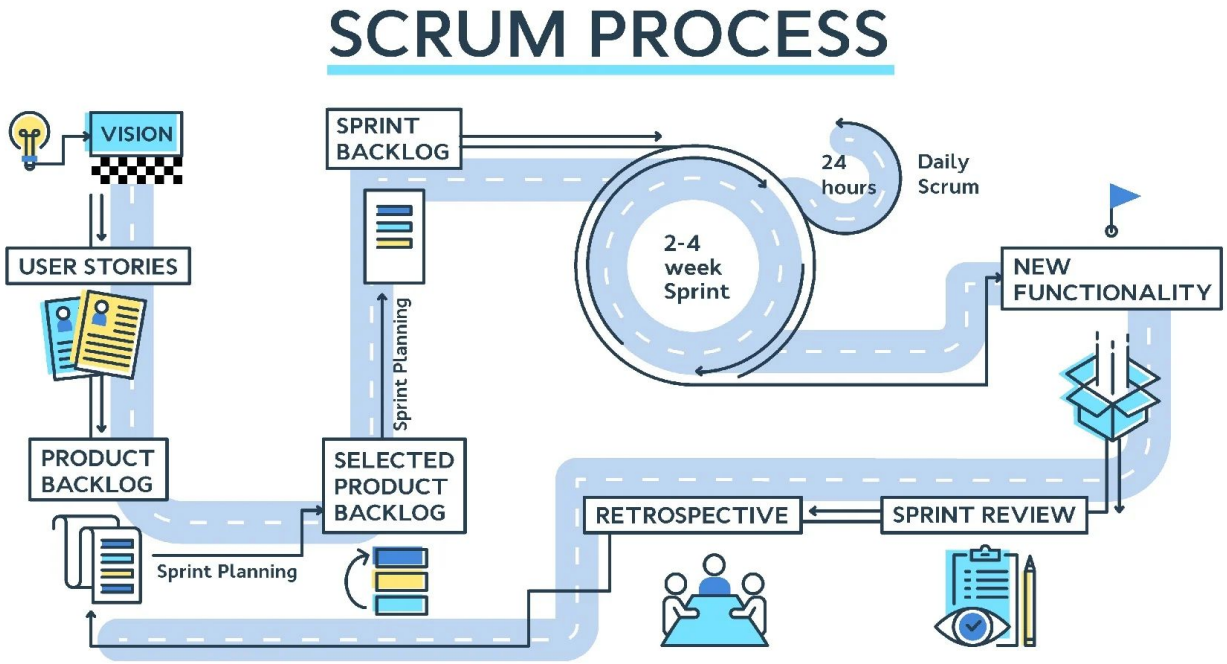
Value

Value brought by a story to the stakeholder

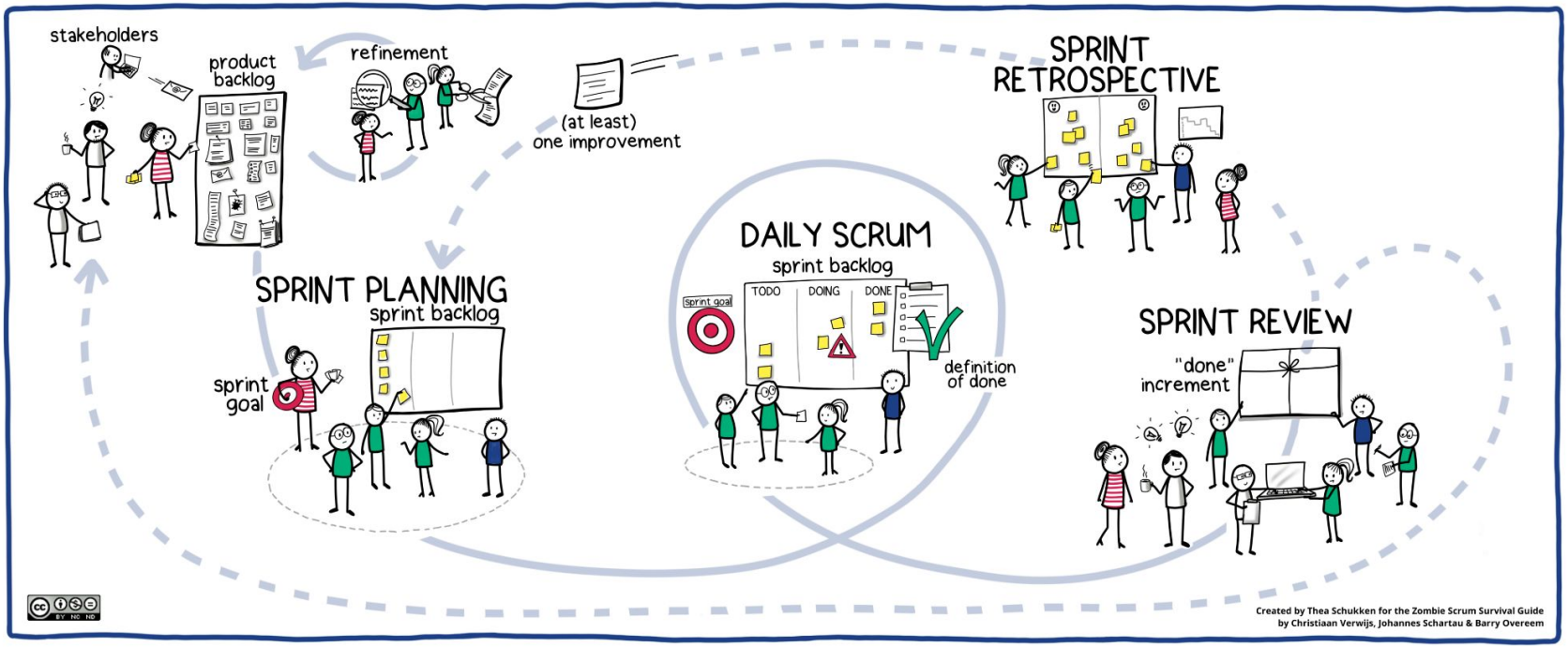
Complexity

Complexity estimation of implementing a story

SCRUM process

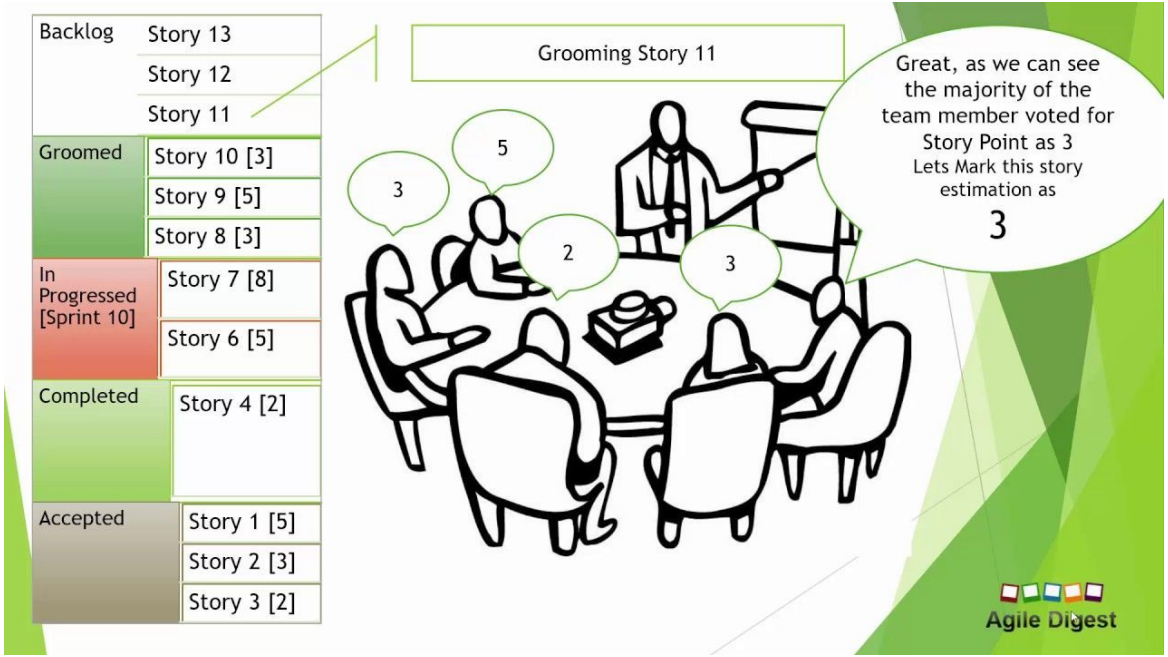


SCRUM organisation



SCRUM planning poker

SCRUM planning poker



DDD

Domain Driven Design is a **methodology** that enabling to **design** structure and language of software code to match the **business domain**.

Its goals:

- focus on the **core domain** and **domain logic**

- basing **complex designs** on a **model of the domain**

- collaboration between technical and domain experts to **iteratively refine** a **conceptual model** that addresses particular **domain problems**.

DDD concepts

Context: The setting in which a word or statement appears that determines its meaning;

Domain: A sphere of knowledge (ontology), influence, or activity.
The subject area to which the user applies a program is the domain of the software;

Model: A system of abstractions that describes selected aspects of a domain and can be used to solve problems related to that domain;

Ubiquitous Language: A language structured around the domain model and used by all team members to connect all the activities of the team with the software.

DDD definitions

Entity

An object that is not defined by its attributes, but rather by a thread of continuity and its identity.

Example: Most airlines distinguish each seat uniquely on every flight. Each seat is an entity in this context. However, Southwest Airlines, EasyJet and Ryanair do not distinguish between every seat; all seats are the same. In this context, a seat is actually a value object.

DDD definitions

Value object

An object that contains attributes but has no conceptual identity.
They should be treated as immutable.

Example: When people exchange business cards, they generally do not distinguish between each unique card; they are only concerned about the information printed on the card. In this context, business cards are value objects.

DDD definitions

Aggregate

A group of objects that are bound together by a root entity: the aggregate root.
Objects outside the aggregate are allowed to hold references to the root but not to any other object of the aggregate.
The aggregate root is responsible for checking the consistency of changes in the aggregate.

Example: When you drive a car, you do not have to worry about moving the wheels forward, making the engine combust with spark and fuel, etc.; you are simply driving the car. In this context, the car is an aggregate of several other objects and serves as the aggregate root to all of the other systems.

DDD definitions

Domain Event

A domain object that defines an event (something that happens).
A domain event is an event that domain experts care about.

Service

When an operation does not conceptually belong to any object. Following the natural contours of the problem, you can implement these operations in services.

DDD definitions

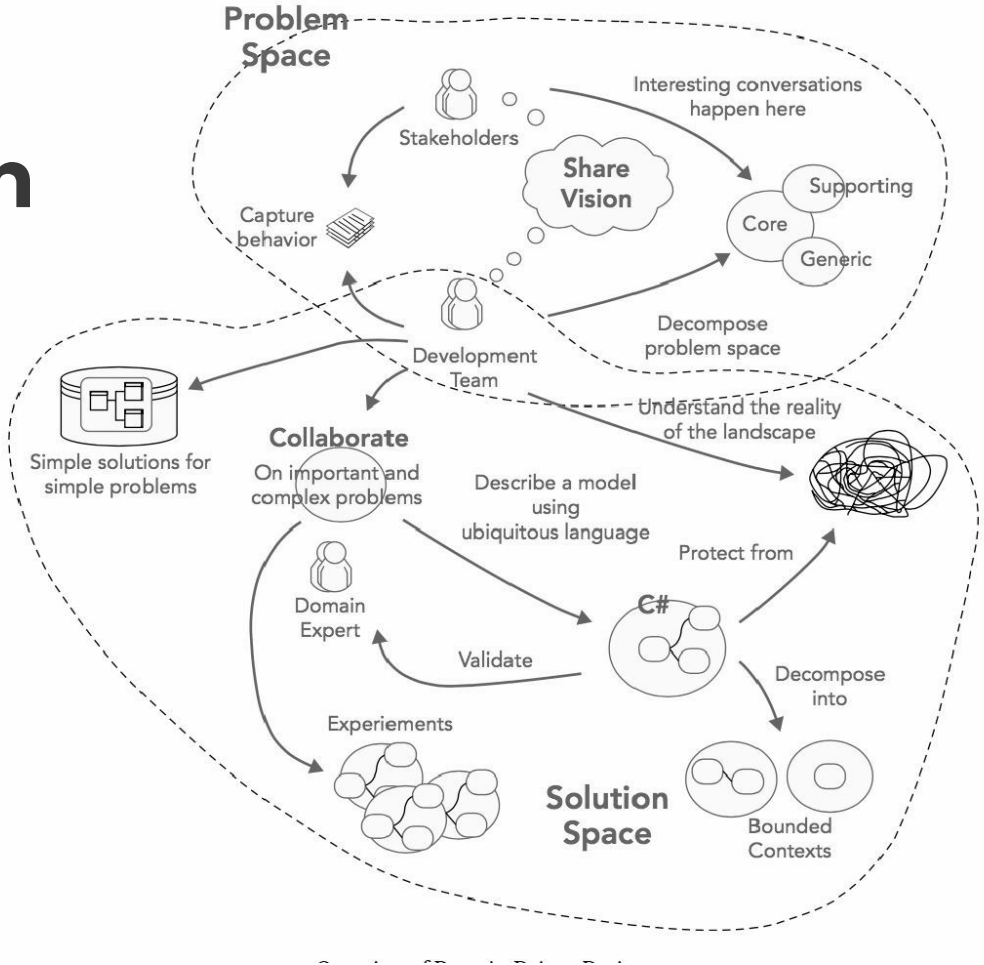
Repository

Methods for retrieving domain objects should delegate to a specialized Repository object such that alternative storage implementations may be easily interchanged.

Factory

Methods for creating domain objects should delegate to a specialized Factory object such that alternative implementations may be easily interchanged.

DDD ecosystem



EDA

Event Driven Architecture is a software architecture framework based on **generation** and **reaction** of **events**

- Event generation / production
- Event consumption
- Event channels

Event storming

It is a method to quickly find out **what** is happening in the **domain** of an IT system

Bridges the gap between product & technical points of views, particularly suited for DDD and EDA contexts

Event storming definitions

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

User
A person who executes a command through a view.

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

Event storming definitions

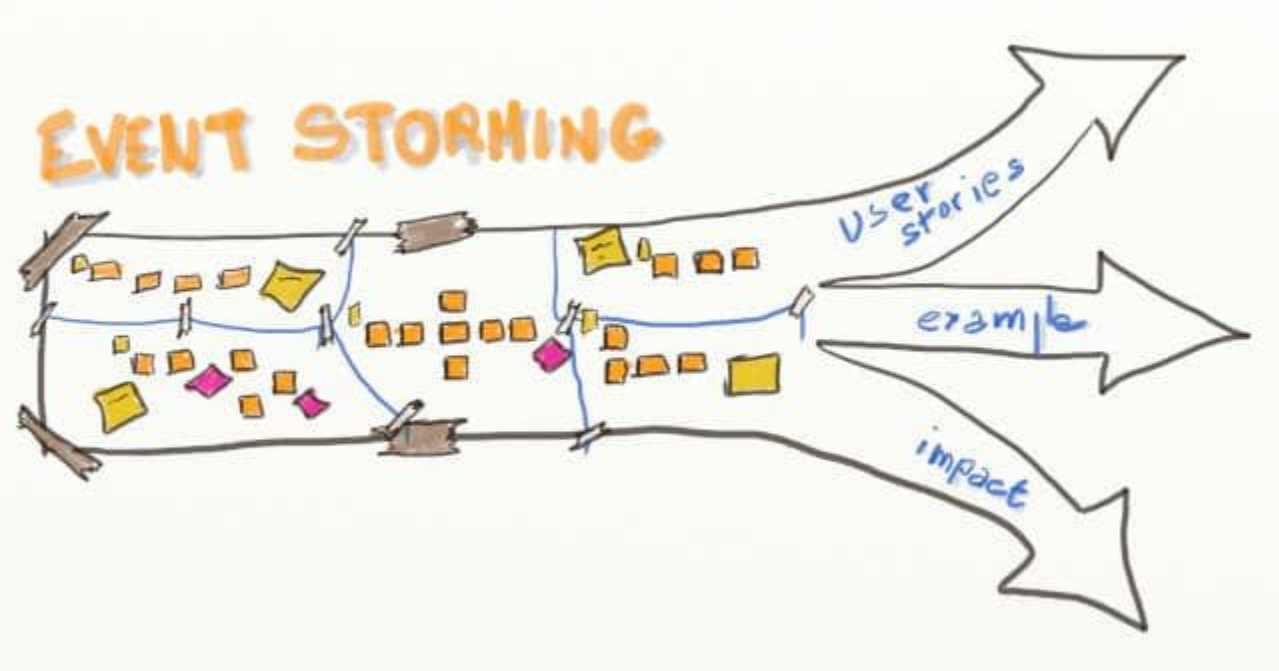
Command
A command executed by a user through a view on an aggregate that results in the creation of a domain event.

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

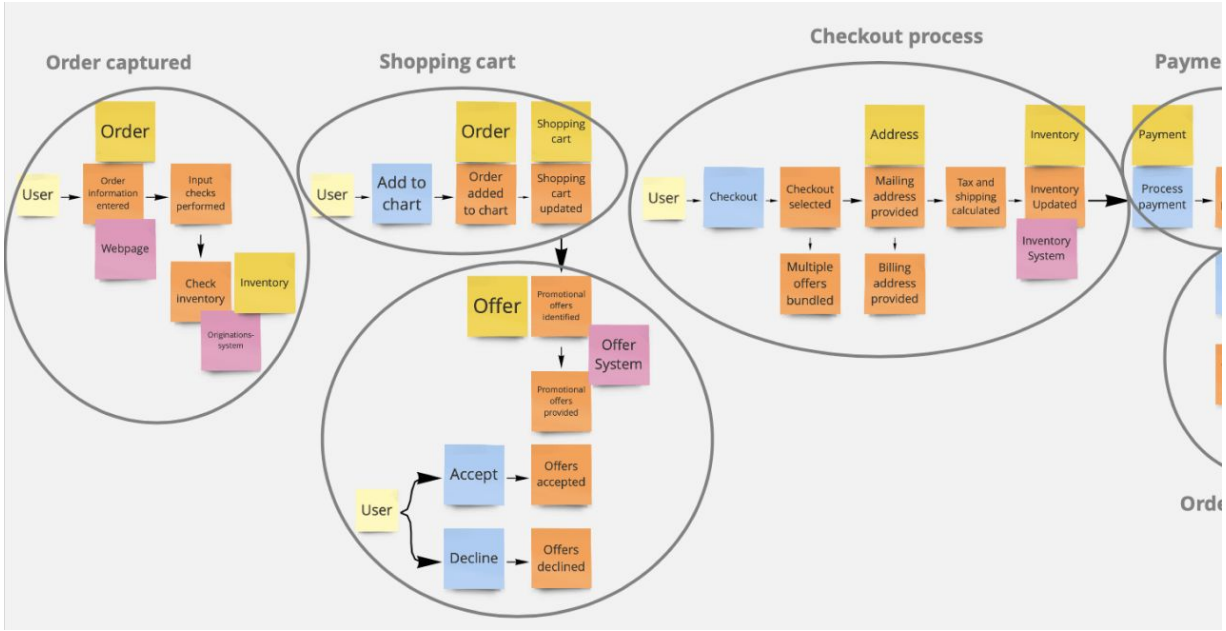
External system
A third-party service provider such as a payment gateway or shipping company.

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

Event storming process



Event storming



Session learnings

- Lean
- SCRUM
- DDD
- EDA

For the next session

- Define your project **personas**
- Draw your **Event Storming scheme** and write your Ubiquitous language lexical

Next session

- Tech teams and roles
- Team organisation
- Team management