

PREGRADO



UNIDAD 3 | HIGH-LEVEL SOFTWARE DESIGN & PATTERNS

# DOMAIN DRIVEN DESIGN PART 2



Al finalizar la unidad, el estudiante elabora y comunica artefactos de diseño de software aplicando principios básicos y patrones empresariales de diseño para un dominio y contexto determinados.

—  
DA  
RO  
ENTSTORMING  
MAIN MESSAGE FLOW  
ODELLING  
UNDDED CONTEXT  
NVAS  
NTEXT MAPPING  
E RECIPE

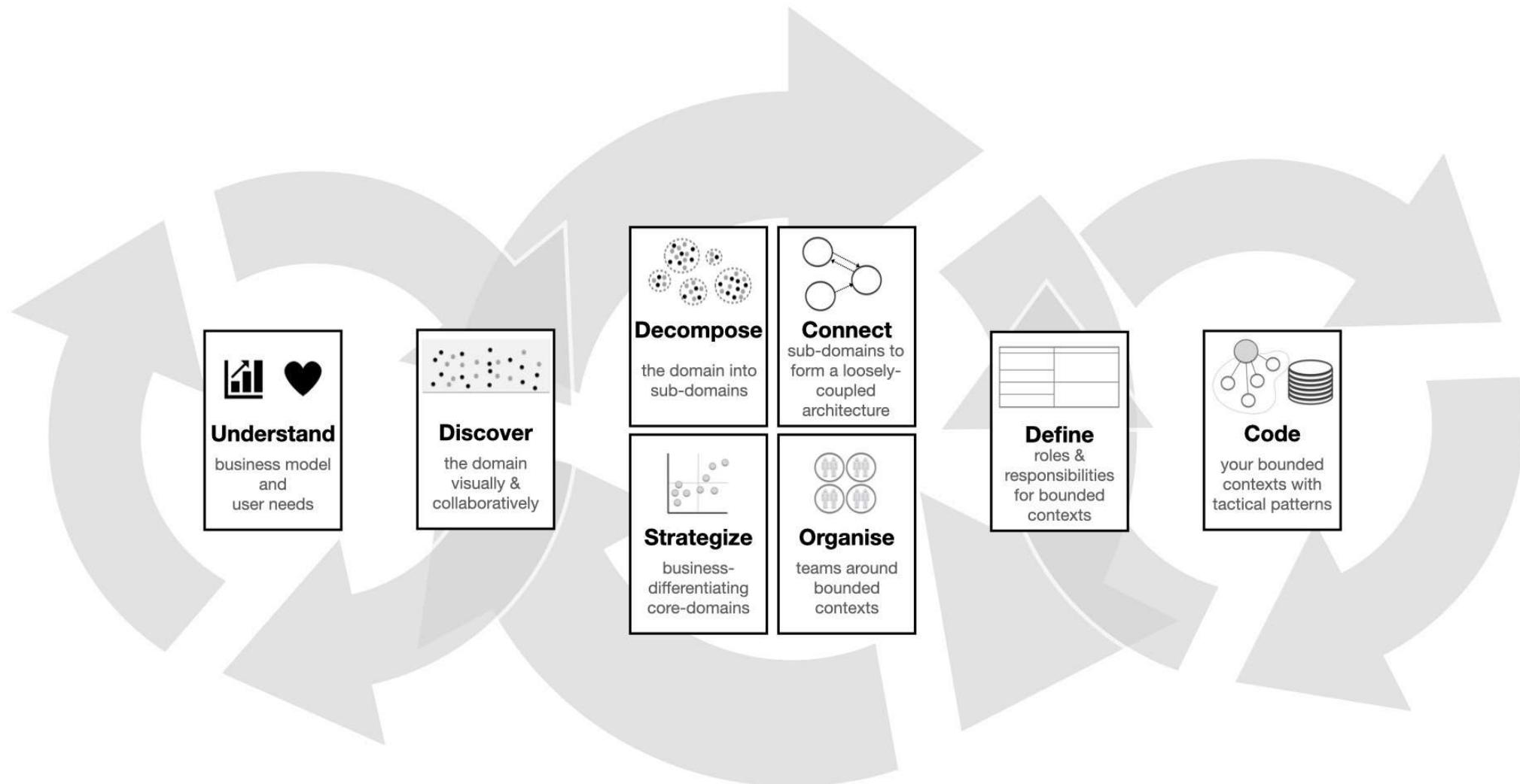


# Modeling Process

Step-By-Step Process:

## Domain-Driven Design Starter Modelling Process

A starter process for beginners, not a rigid best-practice. DDD is continuous, evolutionary, and iterative design.



# Understand the Business

The fact is that the system that people work in and the interaction with people may account for 90 or 95 percent of performance. [William Edwards Deming]

# **Understand the Business**

Esta cita de William Edwards (Bill) Deming revela una gran verdad: el desarrollo de software es una actividad de equipo. El éxito y el fracaso es el éxito o el fracaso del equipo. Según él, si la dinámica de trabajo de los equipos no es buena, la relevancia de las habilidades técnicas de los desarrolladores individuales representa solo el 5% del rendimiento general del equipo. Esa es la razón por la que David J. Anderson, en su obra maestra, libro Kanban, dice que "un equipo con solo una variedad de talento puede producir resultados de clase mundial con el proceso correcto, buena disciplina y liderazgo.

# Tools: New Digital-Experience-Based business

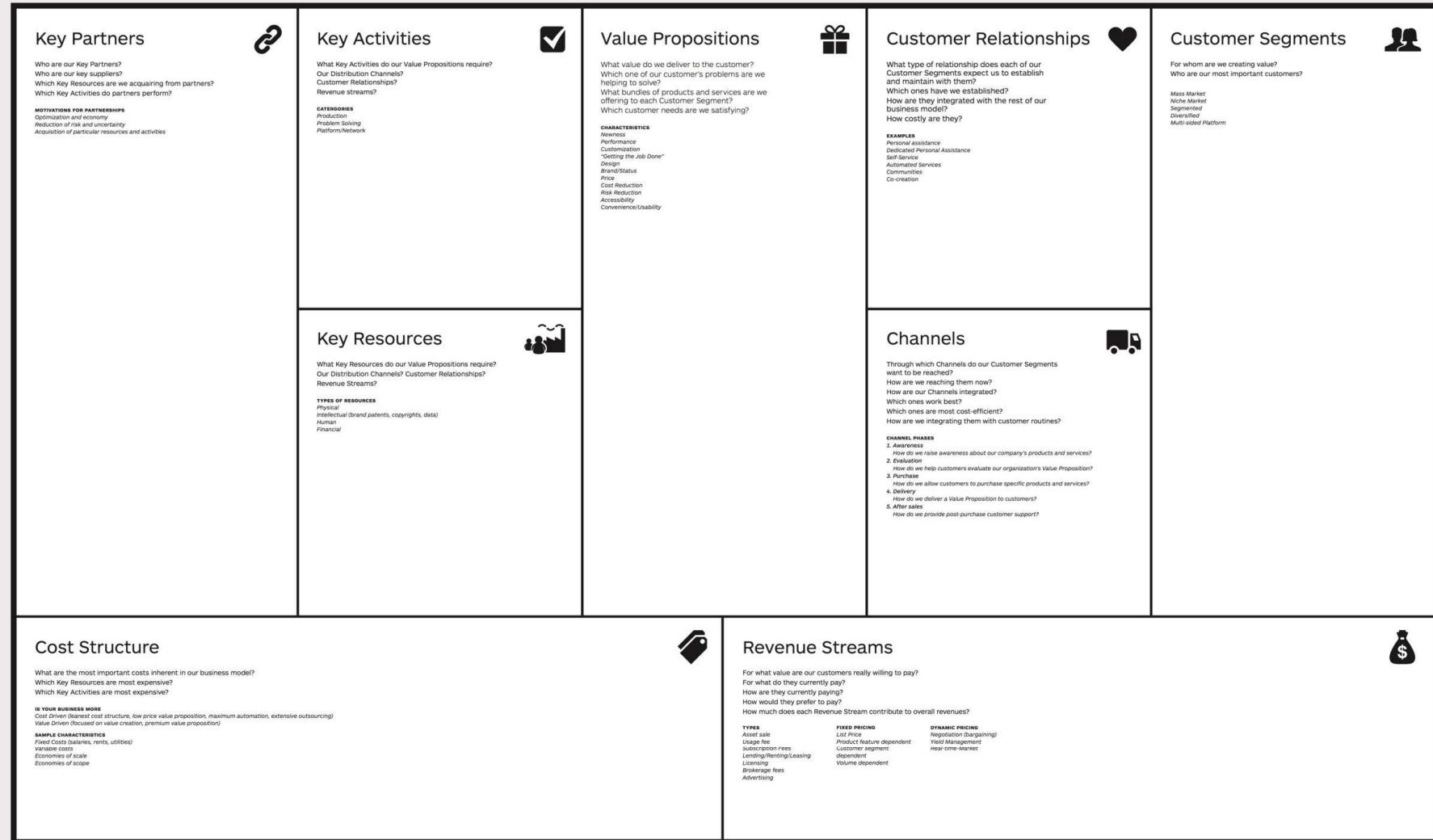
## Lean UX Canvas (v2)

Title of Initiative: _____ Date: _____ Iteration: _____		
<b>Business Problem</b> What problem does the business have that you are trying to solve? (Hint: Consider your current offerings and how they deliver value, changes in the market, delivery channels, competitive threats and customer behavior.)  <b>1</b>	<b>Solutions</b> What can we make that will solve our business problem and meet the needs of our customers at the same time? List product, feature, or enhancement ideas here.  <b>5</b>	<b>Business Outcomes</b> How will you know you solved the business problem? What will you measure? (Hint: What will people/users be doing differently if your solutions work? Consider metrics that indicate customer success like average order value, time on site, and retention rate.)  <b>2</b>
<b>Users</b> What types (i.e., personas) of users and customers should you focus on first? (Hint: Who buys your product or service? Who uses it? Who configures it? Etc)  <b>3</b>	<b>User Outcomes &amp; Benefits</b> Why would your users seek out your product or service? What benefit would they gain from using it? What behavior change can we observe that tells us they've achieved their goal? (Hint: Save money, get a promotion, spend more time with family)  <b>4</b>	
<b>Hypotheses</b> Combine the assumptions from 2, 3, 4 & 5 into the following hypothesis statement: "We believe that [business outcome] will be achieved if [user] attains [benefit] with [feature]." (Hint: Each hypothesis should focus on one feature only)  <b>6</b>	<b>What's the most important thing we need to learn first?</b> For each hypothesis from Box 6, identify its riskiest assumptions. Then determine the riskiest one right now. This is the assumption that will cause the entire idea to fail if it's wrong. (Hint: In the early stages of a hypothesis focus on risks to value rather than feasibility)  <b>7</b>	<b>What's the least amount of work we need to do to learn the next most important thing?</b> Design experiments to learn as fast as you can whether your riskiest assumption is true or false.  <b>8</b>

 Download this canvas at: [www.jeffgothelf.com/blog/leanuxcanvas-v2](http://www.jeffgothelf.com/blog/leanuxcanvas-v2)

# Tools: General Business

## The Business Model Canvas



DESIGNED BY: Business Model Foundry AG  
The makers of *Business Model Generation* and *Strategyzer*

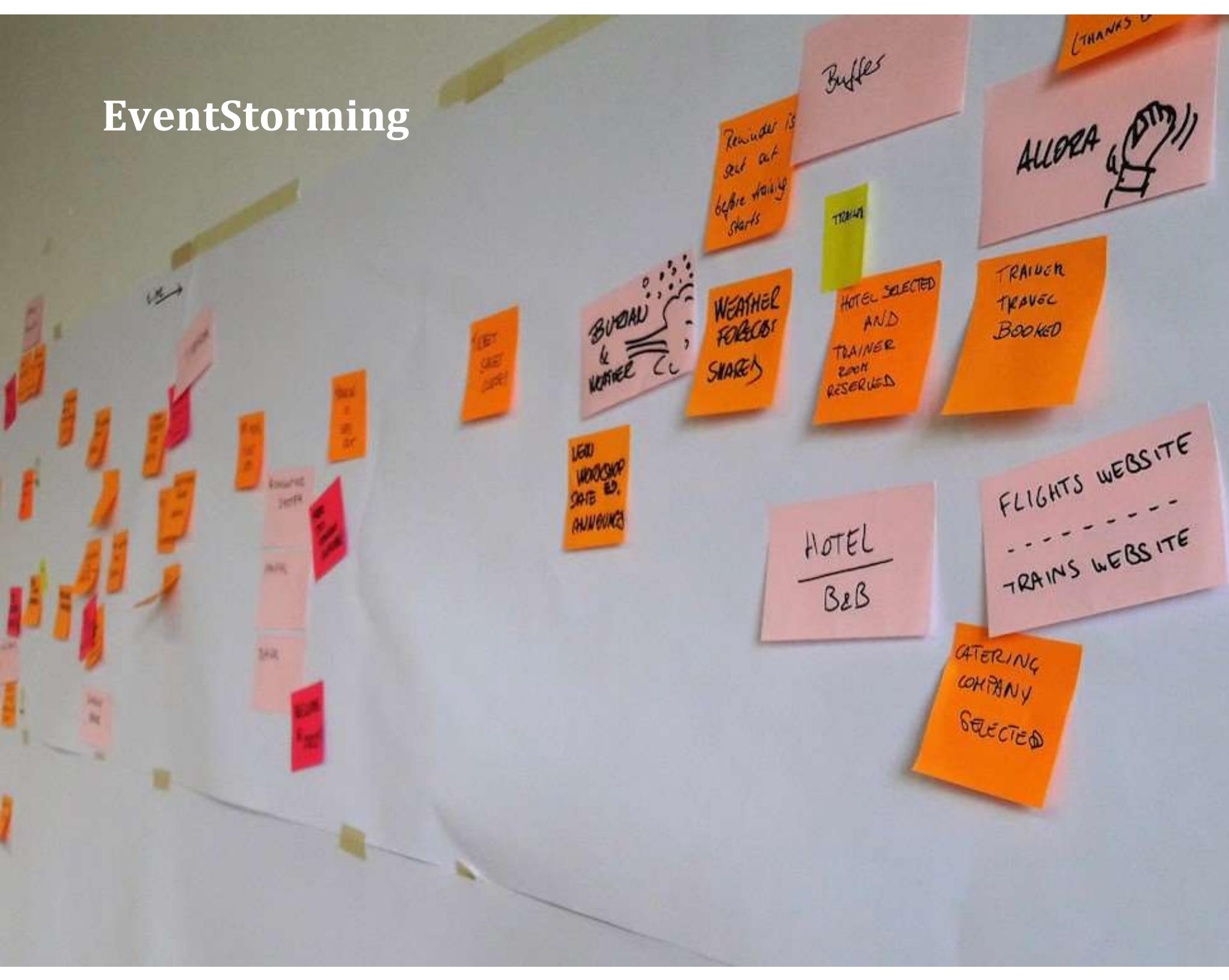
This work is licensed under the Creative Commons Attribution-Share Alike 3.0 Unported License. To view a copy of this license, visit:  
<http://creativecommons.org/licenses/by-sa/3.0/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

DA

RO  
ENTSTORMING  
MAIN MESSAGE FLOW  
ODELLING  
UNDED CONTEXT  
NVAS  
NTEXT MAPPING  
E RECIPE



# EventStorming



# EventStorming

EventStorming [Brandolini 2021] es un método de modelado colaborativo que se desarrolló en el contexto de DDD.

Durante un taller de EventStorming, los desarrolladores y expertos de varios departamentos utilizan notas adhesivas para crear una imagen de los procesos comerciales complejos.

Esta imagen se crea de abajo hacia arriba escribiendo eventos de dominio en las notas adhesivas y colocándolos en una línea de tiempo de izquierda a derecha.

Después de un comienzo deliberadamente caótico, surge una historia, capturada como un flujo de eventos que son relevantes para los expertos en el dominio.

# Who Should Participate in Event Storming?

Idealmente, un grupo diverso de personas debería participar en el taller. De hecho, cualquier persona relacionada con el dominio empresarial en cuestión puede participar: ingenieros, expertos en el dominio, propietarios de productos, probadores, diseñadores de UI /UX, personal de soporte, etc. A medida que se involucren más personas con diferentes antecedentes, se descubrirán más conocimientos.

Sin embargo, tenga cuidado de no hacer que el grupo sea demasiado grande. Cada participante debería poder contribuir al proceso, pero esto puede ser un desafío para grupos de más de 10 participantes.

# The EventStorming Process

## Step 1: Unstructured Exploration

EventStorming comienza con una lluvia de ideas de los eventos del dominio relacionados con el dominio empresarial que se está explorando. Un evento de dominio es algo interesante que ha sucedido en el negocio. Es importante formular eventos de dominio en tiempo pasado; están describiendo cosas que ya han sucedido.



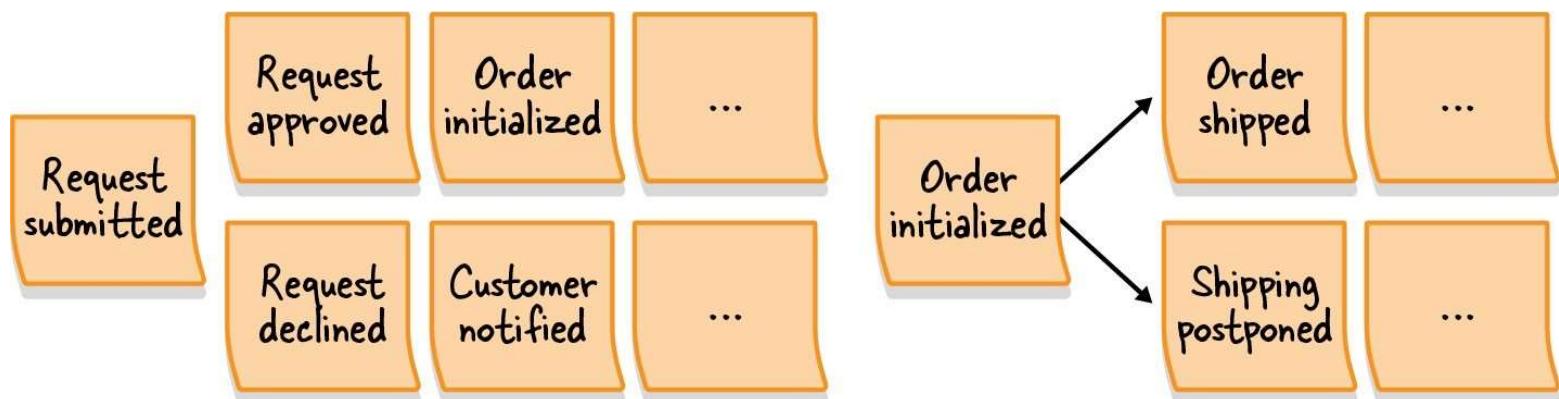
# The EventStorming Process

## Step 2: Timelines

A continuación, los participantes revisan los eventos de dominio generados y los organizan en el orden en que ocurren en el dominio empresarial.

Los eventos deben comenzar con el *happy path*: el flujo que describe un escenario empresarial exitoso.

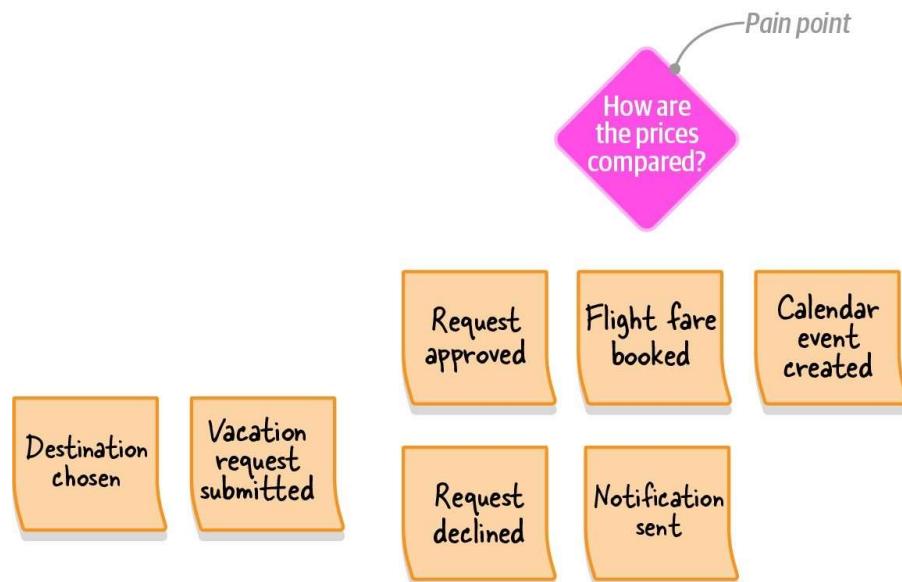
Una vez que se realiza el *happy path*, se pueden agregar escenarios alternativos



# The EventStorming Process

## Step 3: Paint Points

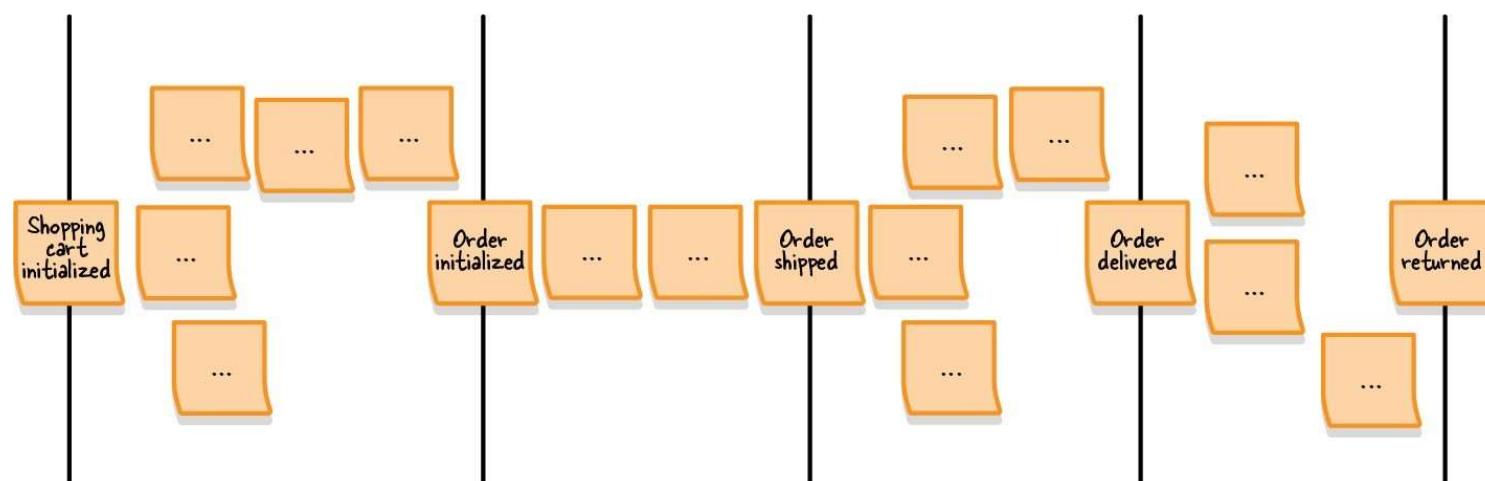
Una vez que tenga los eventos organizados en una línea de tiempo, use esta vista amplia para identificar puntos en el proceso que requieren atención. Estos pueden ser cuellos de botella, pasos manuales que requieren automatización, documentación faltante o conocimiento de dominio faltante.



# The EventStorming Process

## Step 4: Pivotal Points

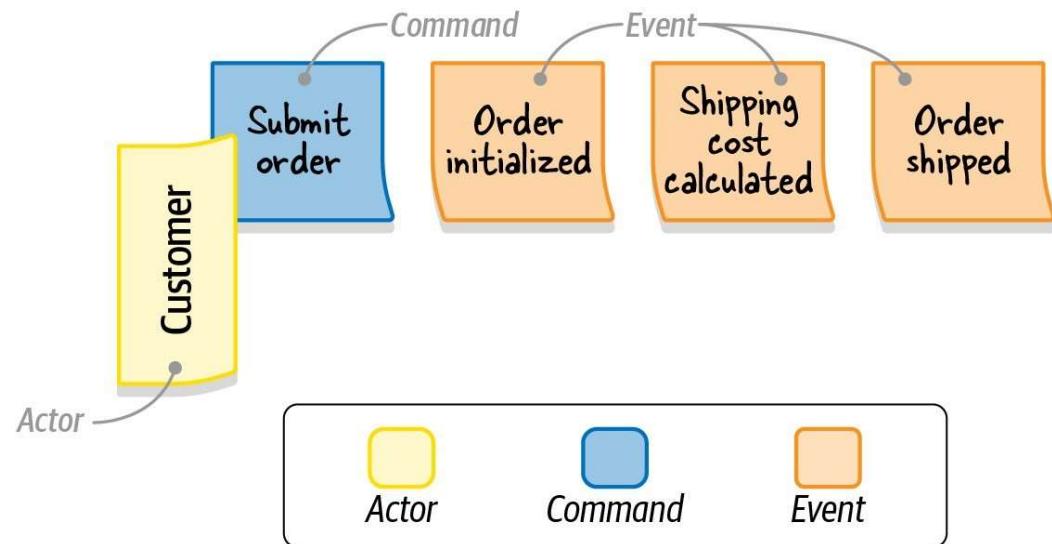
Una vez que tenga una línea de tiempo de eventos aumentada con *paint points*, busque eventos comerciales importantes que indiquen un cambio en el contexto o la fase. Estos se denominan eventos fundamentales y están marcados con una barra vertical que divide los eventos antes y después del evento fundamental.



# The EventStorming Process

## Step 5: Commands

Mientras que un evento de dominio describe algo que ya sucedió, un comando describe qué desencadenó el evento o el flujo de eventos. Los comandos describen las operaciones del sistema y, contrariamente a los eventos de dominio, se formulan en imperativo.



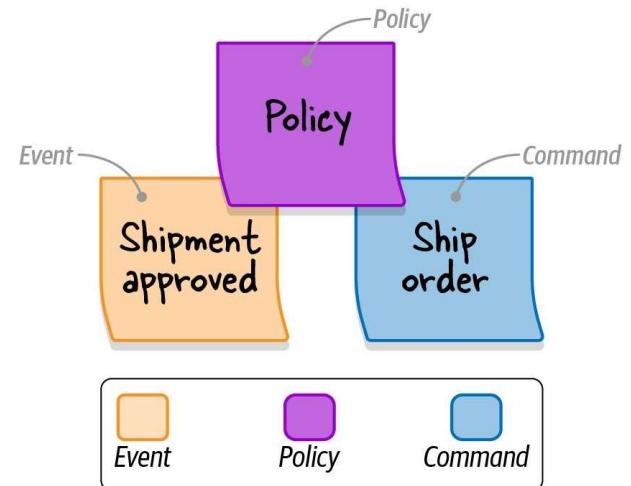
# The EventStorming Process

## Step 6: Policies

Casi siempre, algunos comandos se agregan al modelo pero no tienen un actor específico asociado con ellos. Durante este paso, busca *automation policies* que puedan ejecutar esos comandos.

Una *automation policy* es un escenario en el que un evento desencadena la ejecución de un comando.

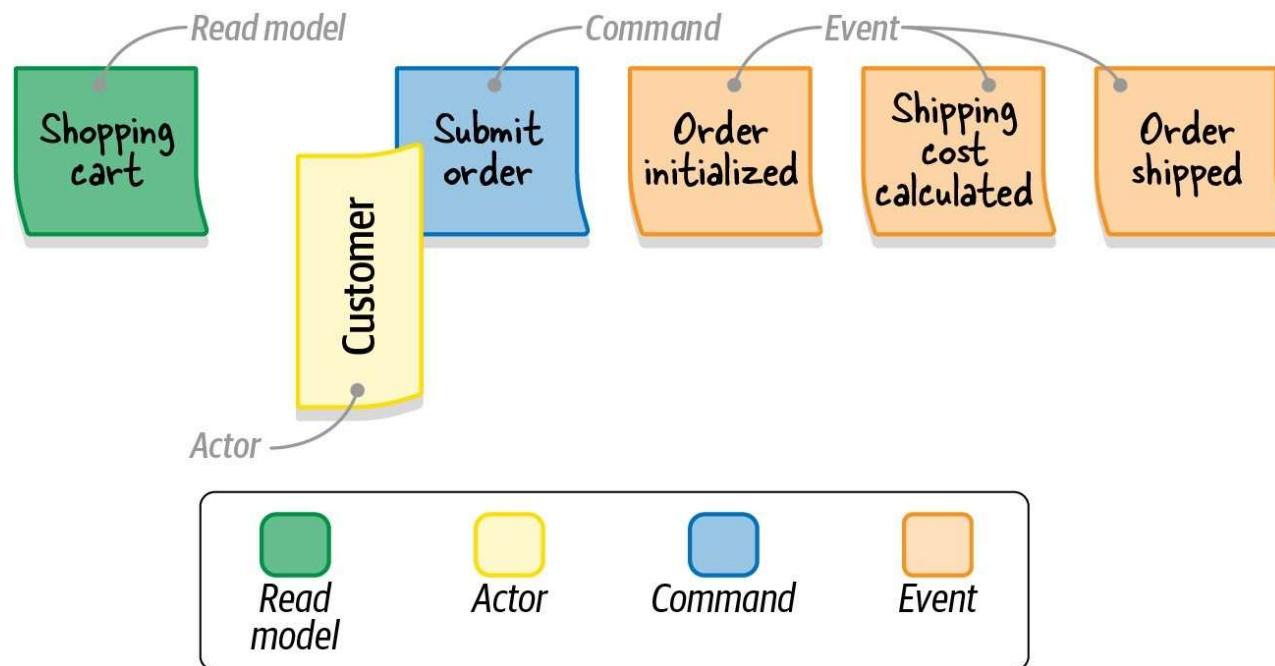
En otras palabras, un comando se ejecuta automáticamente cuando ocurre un evento de dominio específico.



# The EventStorming Process

## Step 7: Read models

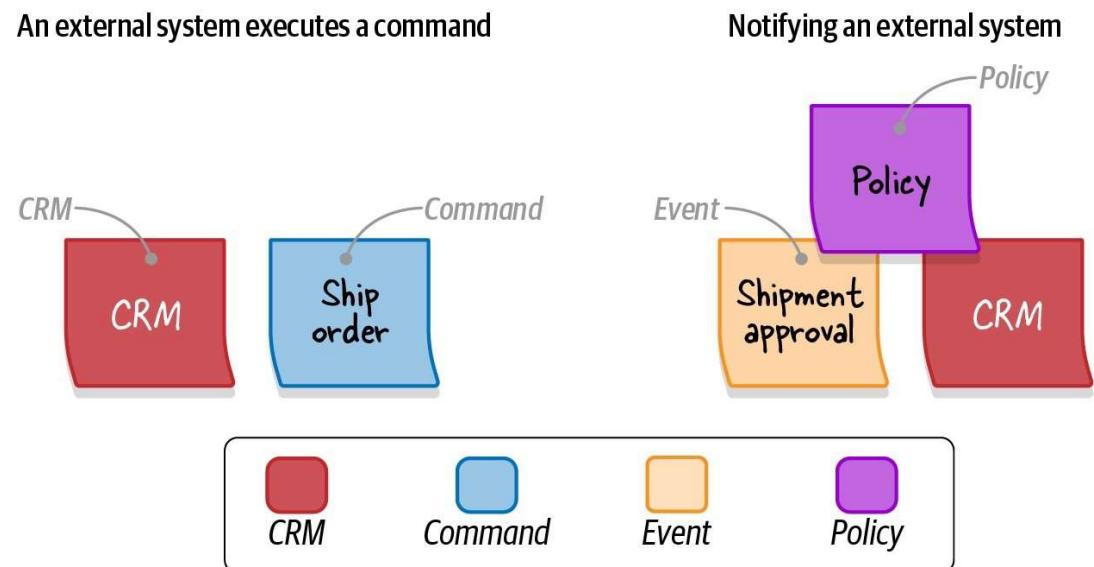
Un modelo de lectura es la vista de datos dentro del dominio que el actor usa para tomar la decisión de ejecutar un comando. Puede ser una de las pantallas del sistema, un informe, una notificación, etc..



# The EventStorming Process

## Step 8: External Systems

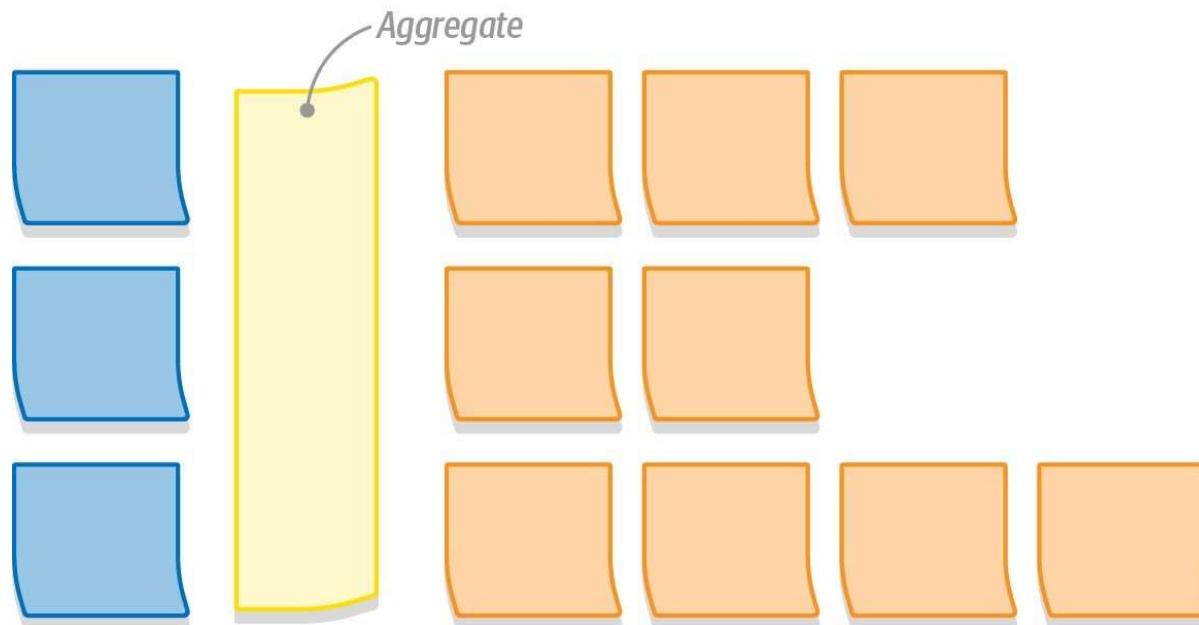
Este paso consiste en aumentar el modelo con sistemas externos. Un sistema externo se define como cualquier sistema que no forma parte del dominio que se está explorando. Puede ejecutar comandos (entrada) o puede ser notificado sobre eventos (salida).



# The EventStorming Process

## Step 9: Aggregates

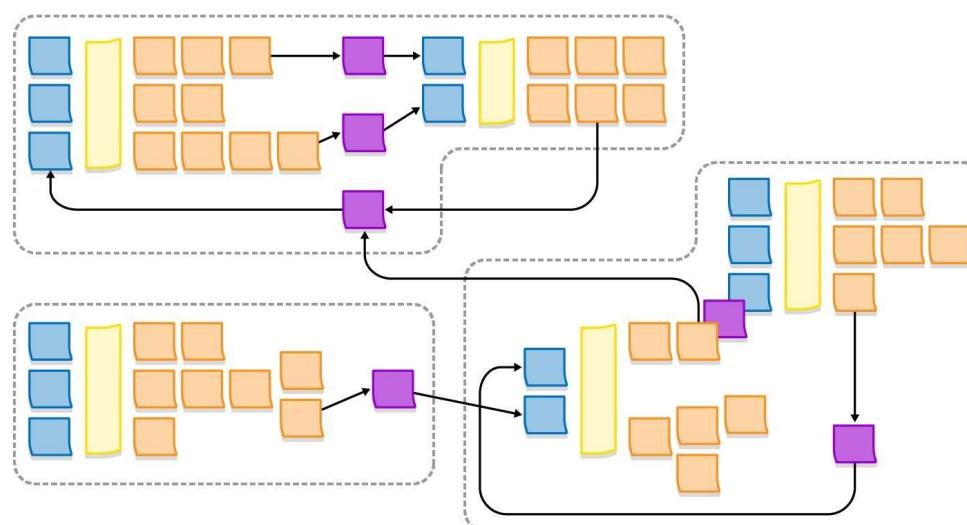
Una vez que todos los eventos y comandos están representados, los participantes pueden comenzar a pensar en organizar conceptos relacionados en agregados. Un agregado recibe comandos y produce eventos.



# The EventStorming Process

## Step 10: Bounded Contexts

El último paso de una sesión de tormenta de eventos es buscar agregados que estén relacionados entre sí, ya sea porque representan una funcionalidad estrechamente relacionada o porque están acoplados a través de políticas. Los grupos de agregados forman candidatos naturales para los límites de los contextos delimitados.



DA

RO  
ENTSTORMING  
MAIN MESSAGE FLOW  
ODELLING  
UNDDED CONTEXT  
NVAS  
NTEXT MAPPING  
E RECIPE



# Domain Message Flow Modelling

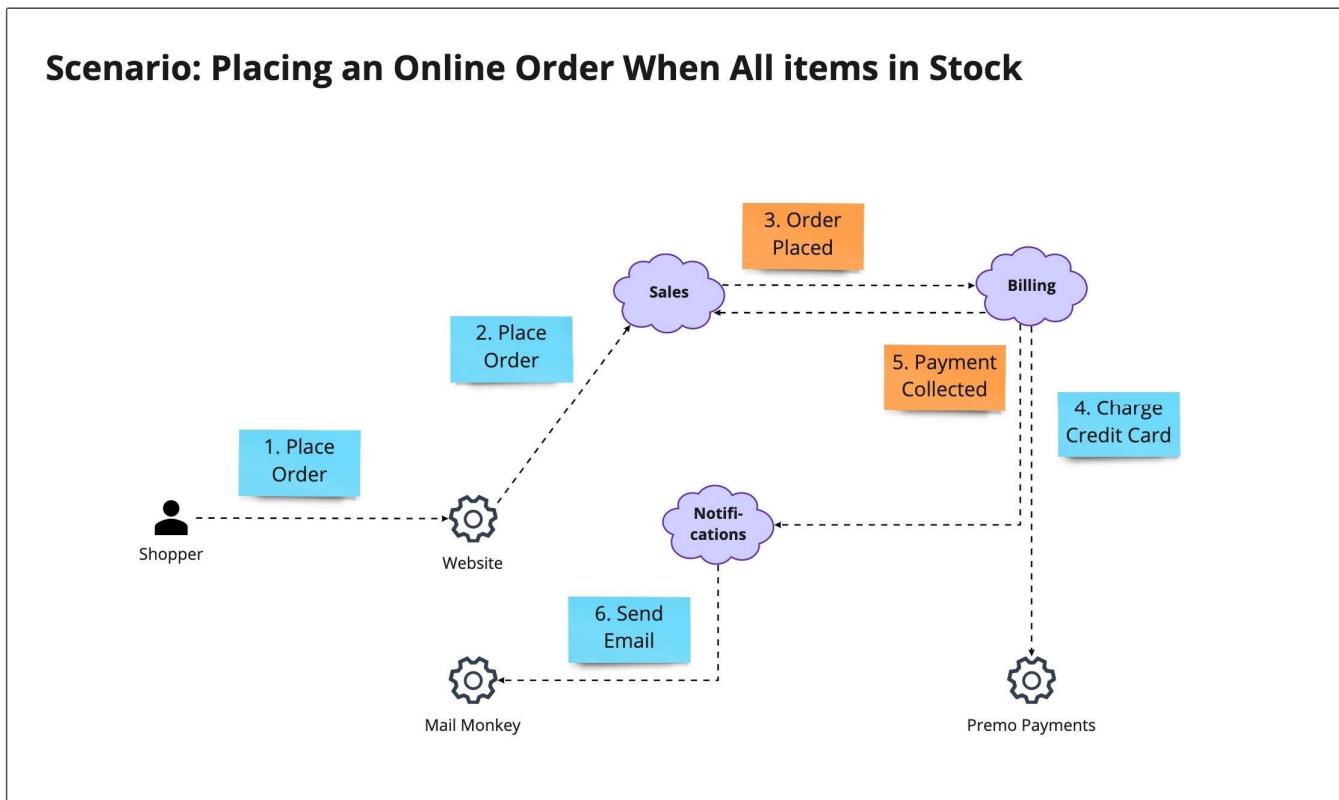
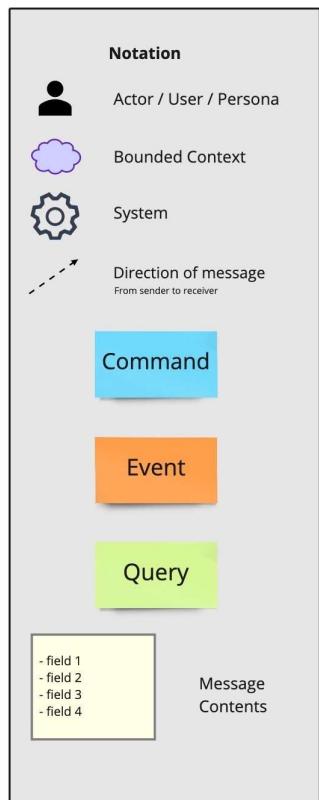
Diseñar sistemas con bajo acoplamiento require más que límites cuidadosamente diseñados. También se requiere interacciones entre bounded contexts cuidadosamente definidas.

Un bounded context es un sub-Sistema en la arquitectura de software, que se alinea con una parte del dominio.

Puede implementarse como un microservicio o un modulo en un monolito. Un Domain message Flow Diagram es una visualización simple mostrando el flujo de mensajes (commands, events, queries) entre actors, bounded contexts y systems, para un scenario.

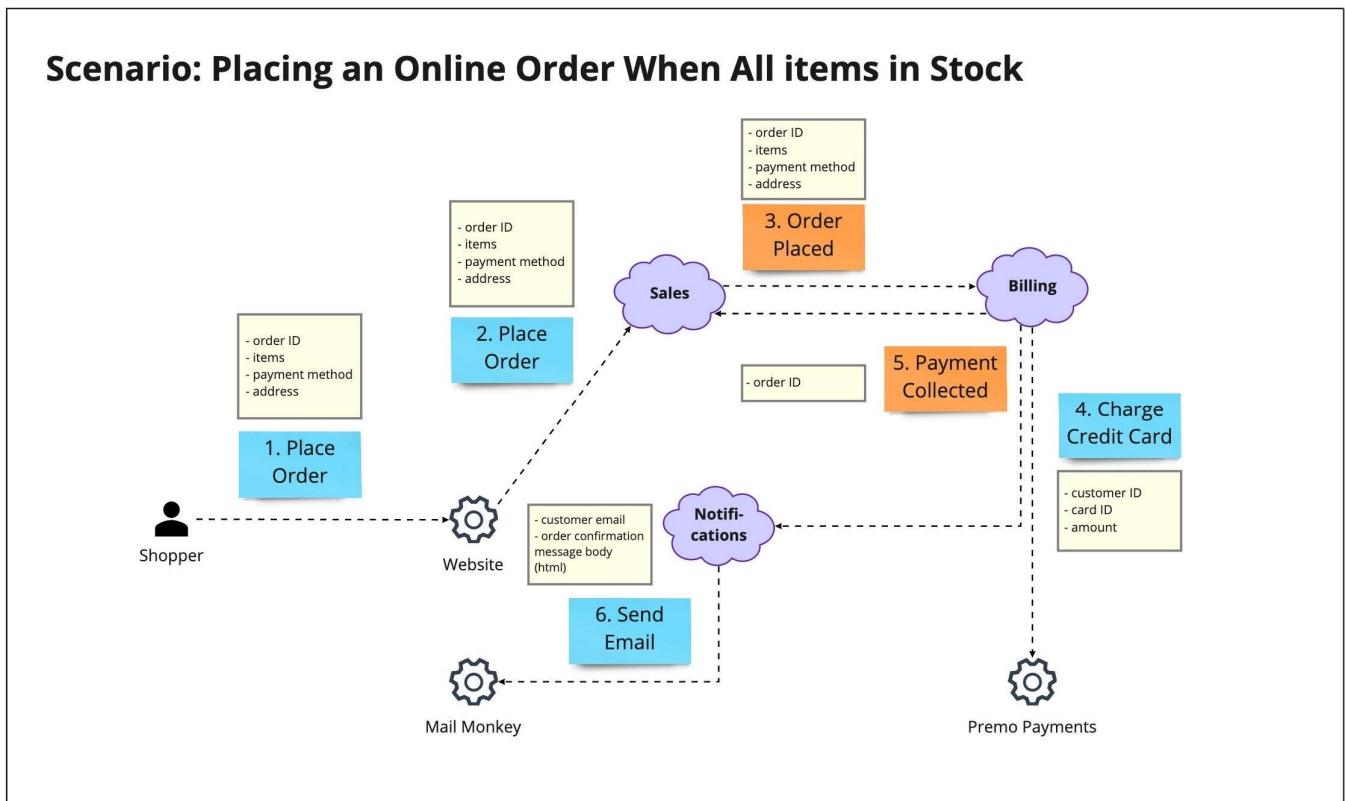
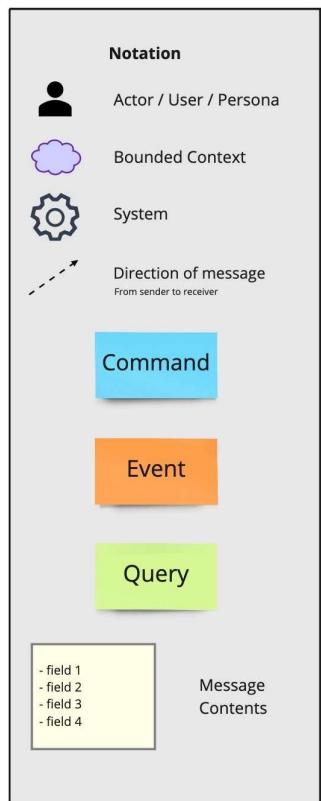
# Formats

## Format 1: Separate message and Contents



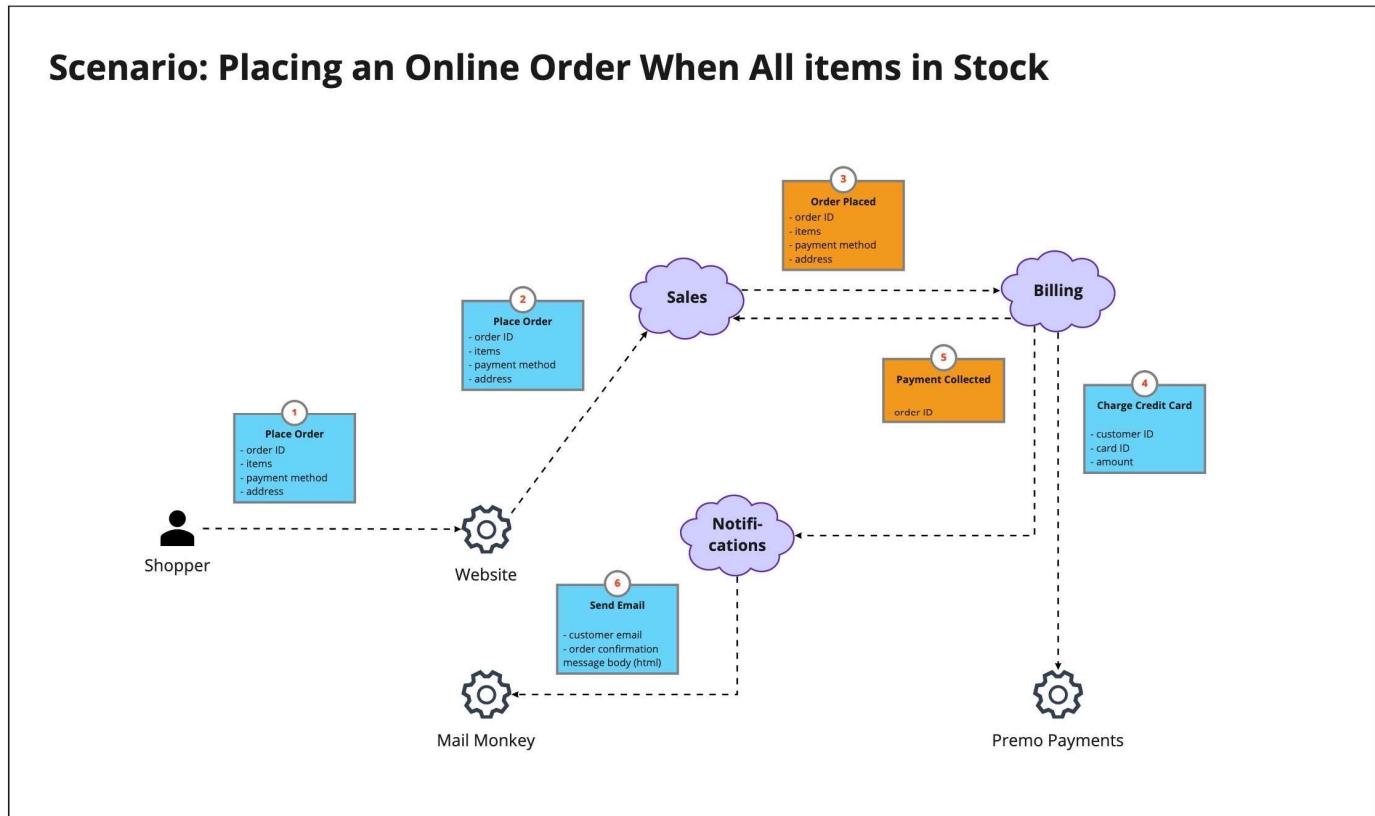
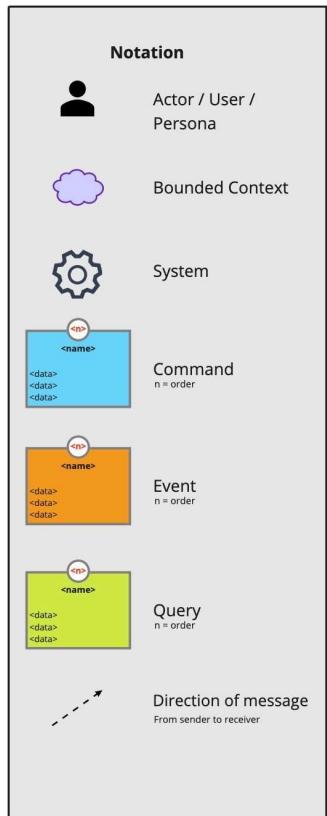
# Formats

## Format 1: Separate message and Contents



# Formats

## Format 2: Combined Message and Contents



# How to use

Crear una lista de escenarios. Para cada scenario crear un diagrama.

El flujo típico para crear un diagrama:

1. Iniciar con un actor /context /system
2. Crear el mensaje que desean enviar.
3. Agregar el receptor del mensaje y una línea conectando al emisor y receptor.
4. Ubique el mensaje cerca de la línea.

Repita los pasos 1-4 hasta que el scenario esté completo.

Elementos del mensaje :

*Nombre del mensaje*

*Datos significativos*

contenidos en el mensaje

*El orden* en el cual ocurre el mensaje dentro del flujo que se está modelando



—  
DA  
RO  
ENTSTORMING  
MAIN MESSAGE FLOW  
ODELLING  
OUNDED CONTEXT  
NVAS  
NTEXT MAPPING  
E RECIPE



# Bounded Context Canvas

Bounded Context Canvas es una herramienta colaborativa para diseñar y documentar el diseño de un único contexto delimitado.

Un contexto limitado es un subsistema en una arquitectura de software alineado con una parte de su dominio.

El lienzo lo guía a través del proceso de diseño de un contexto delimitado al solicitarle que considere y tome decisiones sobre los elementos clave de su diseño, desde la denominación hasta las responsabilidades, su interfaz pública y sus dependencias.

# Bounded Context Canvas

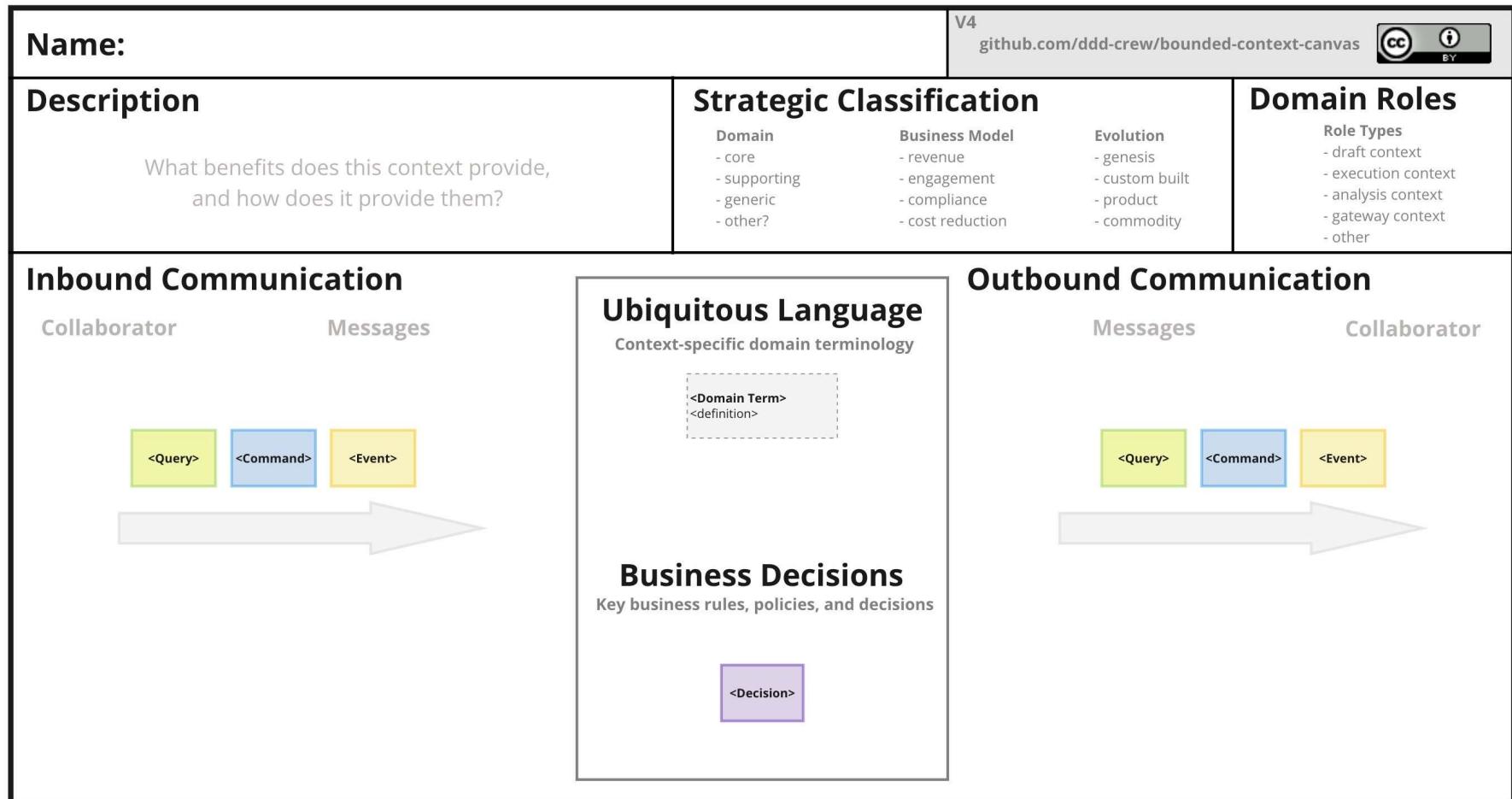
Herramienta para diseñar Bounded Contexts

<b>Name</b>	<b>Model Traits</b> draft, execute, audit, enforcer, interchange, gateway, etc.																
<b>Description</b> Summary of purpose and responsibilities	<b>Messages Consumed and Produced</b>  <b>Messages Consumed</b> <b>Messages Produced</b>																
<b>Strategic Classification</b>  <table><tr><td>Domain</td><td>Business Model</td><td>Evaluation</td></tr><tr><td>- Core</td><td>- Revenue</td><td>- Genesis</td></tr><tr><td>- Supporting</td><td>- Engagement</td><td>- Custom built</td></tr><tr><td>- Generic</td><td>- Compliance</td><td>- Product</td></tr><tr><td>- Other</td><td>- Cost reduction</td><td>- Commodity</td></tr></table>	Domain	Business Model	Evaluation	- Core	- Revenue	- Genesis	- Supporting	- Engagement	- Custom built	- Generic	- Compliance	- Product	- Other	- Cost reduction	- Commodity	Commands Handled	Commands Sent
Domain	Business Model	Evaluation															
- Core	- Revenue	- Genesis															
- Supporting	- Engagement	- Custom built															
- Generic	- Compliance	- Product															
- Other	- Cost reduction	- Commodity															
Events Handled	Events Published																
Queries Handled	Queries Invoked																
<b>Business Decisions</b> Key business rules, policies, and decisions	<b>Dependencies and Relationships</b>  <b>Message Suppliers</b> Name      Relationship  <b>Message Consumers</b> Name      Relationship																
<b>Ubiquitous Language</b> Key domain terminology	Bounded Context Canvas V3 Nick Tune   ntcoding.co.uk																

Bounded Context Canvas v3

# Bounded Context Canvas

Herramienta para diseñar Bounded Contexts



Bounded Context Canvas v4

# Definiciones

## Name

Nombrar es difícil. Anotar el nombre de su contexto y llegar a un acuerdo como equipo enmarcará la forma en que diseña el contexto.

## Description

Algunas frases que describen el por qué y el qué del contexto en el lenguaje comercial. No hay detalles técnicos aquí.

Escribir la descripción lo obliga a articular claramente pensamientos confusos y asegurarse de que todos en el equipo estén en la misma página.

# Definiciones

Strategic Classification

*¿Qué importancia tiene este contexto para el éxito de su organización?:*

*Core domain:* una iniciativa estratégica clave

*Supporting domain:* necesario pero no un diferenciador

*Generic:* una capacidad común que se encuentra en muchos dominios

# Definiciones

Strategic Classification

*¿Qué papel juega el contexto en su modelo de negocio?*

*Revenue generator:* la gente paga directamente por esto

*Engagement creator:* a los usuarios les gusta, pero no pagan por ello.

*Compliance enforcer:* protege la reputación y la existencia de su negocio

# Definiciones

Strategic Classification

*Qué tan evolucionado es el concepto:*

*Genesis:* nuevo dominio inexplorado

*Custom build:* las empresas están creando sus propias versiones

*Product:* existen versiones listas para usar con diferenciación

*Commodity:* existen versiones altamente estandarizadas

# Definiciones

## Inbound Communication

La comunicación entrante representa colaboraciones iniciadas por otros colaboradores.

ffessages:

Los mensajes son la información que un colaborador envía a otro.

Hay tres tipos de conversación que pueden ocurrir entre contextos delimitados. Una solicitud para hacer algo (command),

una solicitud de información (query) o una notificación de que algo ha sucedido (event).

Collaborators:

Los colaboradores son otros sistemas o subsistemas que envían mensajes a este contexto. Pueden ser otros contextos delimitados, frontends (web o móviles) o cualquier otra cosa.

Relationship Type:

El tipo de relación entre dos contextos delimitados indica cómo los modelos y los equipos se influyen entre sí.

Organizing into

Swimlanes:

Los colaboradores se pueden organizar en carriles de natación horizontales que muestran los mensajes que envían.

## Collaborator Types



Bounded Context



External System



Frontend



Direct User Interaction

## Other



Relationship Type

# Inbound Communication

Collaborator



company website

C  
F

Messages

Find nearest  
showroom

Get  
showroom  
details

Close  
showroom

O  
H  
S

P  
N  
R



Sale  
Complete

Refund  
Requested

Quotation  
Produced

K

# Definiciones

## Outbound Communication

La comunicación saliente representa colaboraciones que se inician en este contexto para interactuar con otros colaboradores. Se aplican los mismos tipos de mensajes y notaciones que la comunicación entrante.

## Ubiquitous Language

¿Cuáles son los términos de dominio clave que existen dentro de este contexto y qué significan?

## Business Decisions

¿Cuáles son las reglas y políticas comerciales clave en este contexto?

# Strategic Classification

Documentar el rol que juega el context. Es importante considerar el rol que juega el Contexts en el modelo de negocio y considerar cómo el valor del negocio evolucionará.

## Strategic Classification

### Domain

- Core
- Supporting
- Generic
- Other

### Business Model

- Revenue
- Engagement
- Compliance
- Cost reduction

### Evolution

- Genesis
- Custom built
- Product
- Commodity

# Model Traits

Articular las características de este contexto del business model.

## Model Traits

draft, execute, audit, enforcer, interchange, gateway, etc.

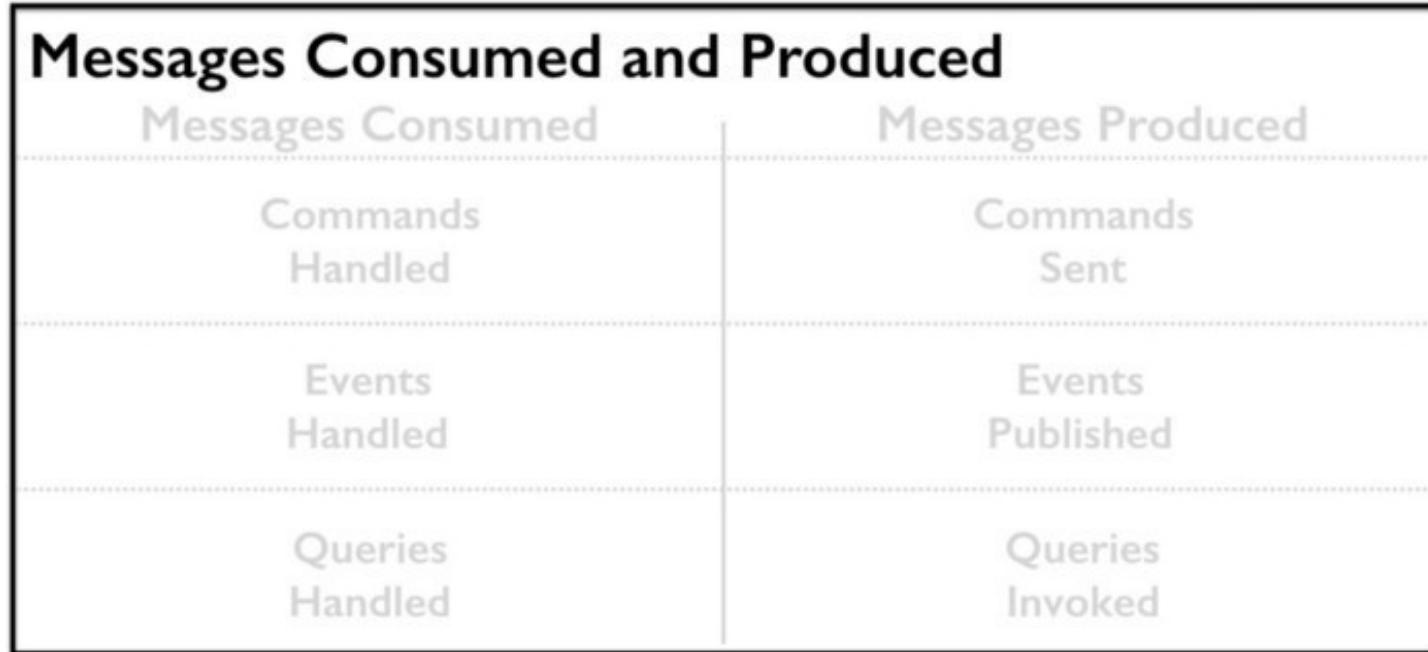
# Messages Consumed and Produced

Documenta las interfaces públicas para un bounded context. En DDD se usa tres tipos de mensajes:

Command. El producer le indica al recipient hacer algo.

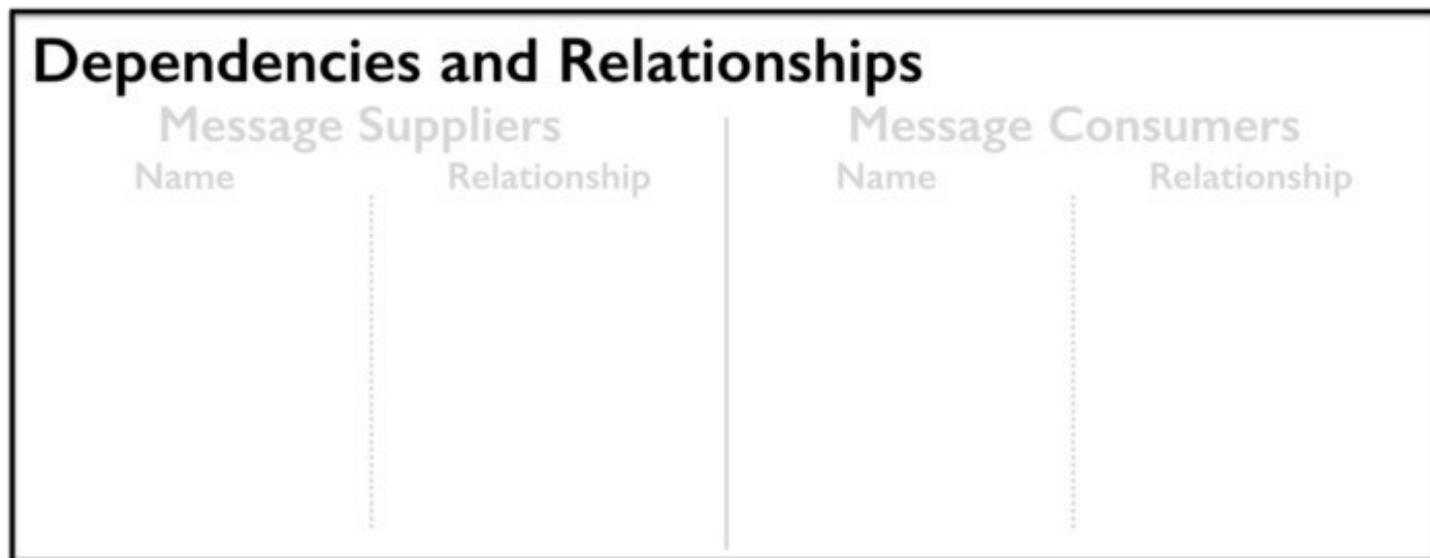
Domain Event. Evento significativo en el dominio que sucede dentro de un solo bounded context. Otros bounded contexts podrían necesitar enterarse de él o responder a él. A diferencia de un Command, no hay obligación de cuándo o cómo responder a un evento.

Query. Un context solicita información de otro, o hace una consulta.



# Dependencies and Relationships

Explica qué bounded context proporcionan mensajes y por qué. Quién consume los mensajes que éste bounded context produce y por qué.



# Information and Services Provided

Documentar la interfaz pública del bounded context. Aquí va lo que otros contexts pueden consumir.

Queryable Information. ¿Qué información pueden otros contexts consultar sobre otros contexts? ¿Qué preguntas pueden hacer?

Invokable Commands. ¿Qué commands pueden invocar otros bounded contexts de este context? ¿Qué pueden pedir ellos?

Published Events. ¿Qué eventos publica este context públicamente al que otros contexts se pueden suscribir?

Reactive Jobs. Qué Jobs realiza este context que se inician internamente – por ejemplo scheduled Jobs, o iniciar un business process cuando se responde a un evento publicado por otro context.

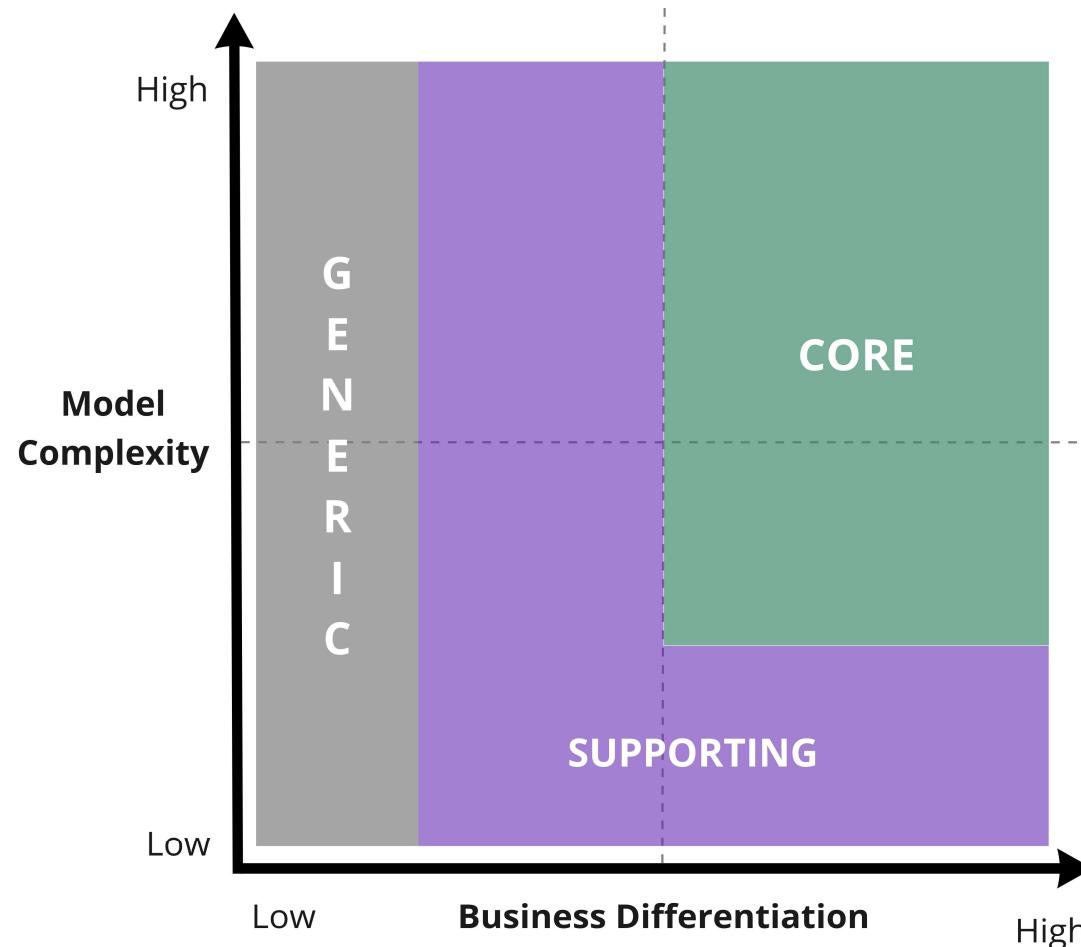


—  
DA  
RO  
ENTSTORMING  
MAIN MESSAGE FLOW  
ODELLING  
UNDDED CONTEXT  
NVAS  
NTEXT MAPPING  
E RECIPE



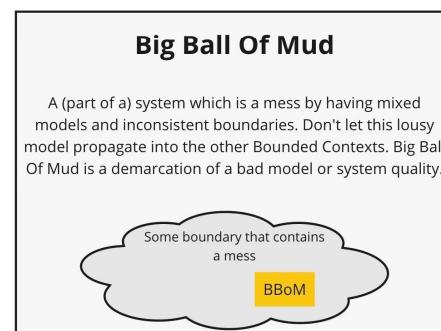
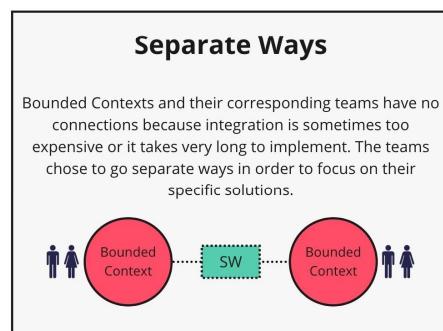
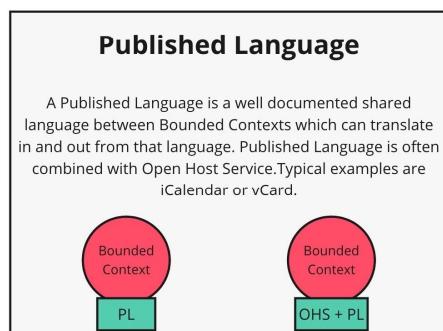
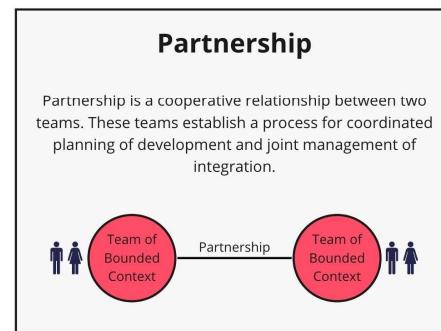
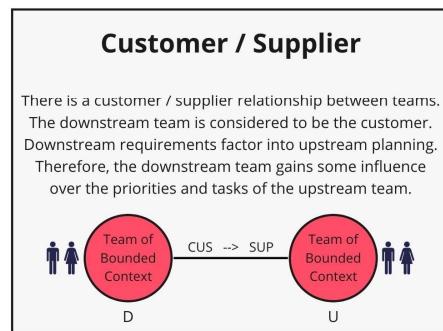
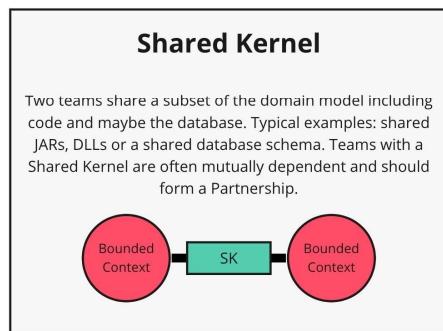
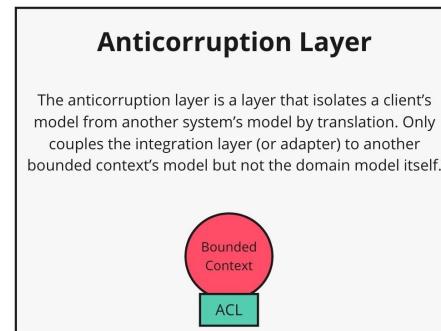
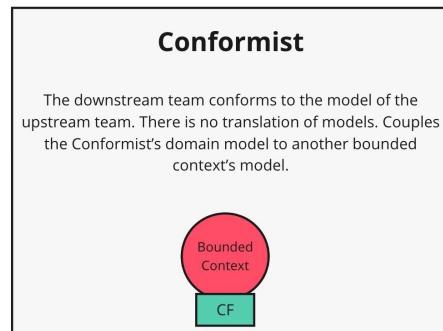
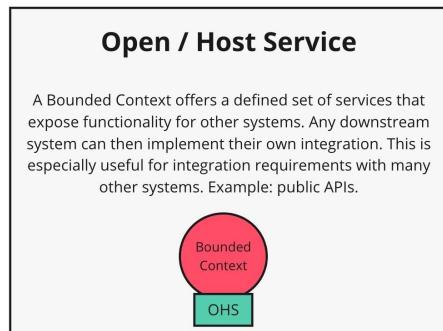
# Context Mapping

Comenzar por core sub-domains: las partes del domain que tienen el. fflaor potencia. Para la diferenciación del negocio o significancia estratégica.



# Context Mapping

## Context ffiap Patterns



# Context Mapping

## Team Relationships

### Mutually Dependent

Two software artifacts or systems in two bounded contexts need to be delivered together to be successful and work. There is often a close, reciprocal link between data and functions between the two systems.



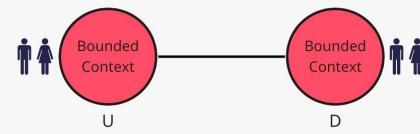
### Free

Changes in one bounded context do not influence success or failure in other bounded contexts. There is, therefore, no organizational or technical link of any kind between the teams.



### Upstream / Downstream

Actions of an upstream team will influence the downstream counterpart while the opposite might not be true. This influence can apply to code but also on less technical factors such as schedule or responsiveness to external requests.



—  
DA  
RO  
ENTSTORMING  
MAIN MESSAGE FLOW  
ODELLING  
UNDDED CONTEXT  
NVAS  
NTEXT MAPPING  
E RECIPE



# The Recipe

Big Picture EventStorming

Candidate Context ffiodelling

Domain ffiessage Flow ffiodelling

Bounded Context Canvas

Refined Context Exploration

---

**RESUMEN**

Recordemos

EventStorming

Bounded Context Canvas



# REFERENCIAS

Para profundizar

ffiodelling Bounded Contexts with the Bounded Context Canvas: A Workshop Recipe

<https://medium.com/nick-tune-tech-strategy-blog/modelling-bounded-contexts-with-the-bounded-context-design-canvas-a-workshop-recipe-1f123e592ab>

The Bounded Context Canvas

<https://github.com/ddd-crew/bounded-context-canvas>

Domain-Driven Design Starter ffiodelling Process

<https://github.com/ddd-crew/ddd-starter-modelling-process>

