

# Defining the Business Domain Using Event Storming

Barry S. Stahl  
Solution Architect – Carvana  
@bsstahl



# FAVORITE PHYSICISTS & MATHEMATICIANS

## FAVORITE PHYSICISTS

1. Harold "Hal" Stahl
2. Carl Sagan
3. Richard Feynman
4. Marie Curie
5. Nikola Tesla
6. Albert Einstein
7. Neil Degrasse Tyson
8. Niels Bohr
9. Galileo Galilei
10. Michael Faraday

## FAVORITE MATHEMATICIANS

1. Ada Lovelace
2. Alan Turing
3. Johannes Kepler
4. Rene Descartes
5. Isaac Newton
6. Leonardo Fibonacci
7. George Boole
8. Blaise Pascal
9. Johann Gauss
10. Grace Hopper

*Other notables:* Stephen Hawking, Edwin Hubble

*Other notables:* Daphne Koller, Benoit Mandelbrot

# SOME OSS PROJECTS I RUN

1. [Liquid Victor](#) : Media tracking and aggregation [used to assemble this presentation]
2. [Prehensile Pony-Tail](#) : A static site generator built in c#
3. [TestHelperExtensions](#) : A set of extension methods helpful when building unit tests
4. [Conference Scheduler](#) : A conference schedule optimizer
5. [IntentBot](#) - A microservices framework for creating conversational bots on top of Bot Framework
6. [LiquidNun](#) : Library of abstractions and implementations for loosely-coupled applications
7. [Toastmasters Agenda](#) : A c# library and website for generating agenda's for Toastmasters meetings

[HTTP://GIVECAMP.ORG](http://givecamp.org)





Achievement Unlocked  
100G - 100 Public Talks

# Come do Amazing things at Carvana!

---



[carvana.com](http://carvana.com)

# Thank You to the Code PaLOUsa Sponsors



<prosoft>

**CGI**

## Friends of Code PaLOUsa



Couchbase

DATASTAX



mongoDB



redis

**ROCKET**  
Mortgage

# EVENT STORMING

A PROCESS FOR MODELING A BUSINESS DOMAIN FROM THE PERSPECTIVE OF THE BUSINESS EXPERTS

# WHY EVENT STORMING?

- We Learn Things
  - Improved Understanding of the Problem
  - Better Naming of Objects and Operations
  - Separation of similar but different concepts
- Not Big Design Up-Front
  - Good Understanding Up-Front

# MODELING THE DOMAIN

- As the Business Sees It
- Implementation Agnostic
- Not storage-centric

# EVENT STORMING OUTPUT

1. Visual Model of the Domain
2. Ubiquitous Language Document (Dictionary)
3. Common Understanding Among Team Members

# DOMAIN DRIVEN DESIGN (DDD)

PRACTICES FOR DESIGNING SYSTEMS FROM COMPONENTS THAT ARE HIGHLY COHESIVE BUT LOOSELY COUPLED

# COHESION

A MEASUREMENT OF THE STRENGTH OF THE RELATIONSHIPS WITHIN A COMPONENT

- Operations
- Data

How closely the data and the operations on that data match within the context of the component

# COUPLING

A MEASUREMENT OF THE STRENGTH OF THE RELATIONSHIPS BETWEEN COMPONENTS

- Operations
- Data

The greater the coupling between components, the more likely we are to have to change one as a result of a change to the other

# APPLICATION TCO

- The Total Cost of Ownership of an application is the sum of all the costs of the application throughout its life-cycle.
- These costs include creating, deploying, extending, maintaining and decommissioning the application.

# LOOSE COUPLING REDUCES TCO

- Tight coupling makes it more difficult to modify one concern of the application without impacting others.
- Maintenance & Extension : more difficult == more expensive

# DOMAIN OBJECT

A REPRESENTATION OF A REAL-WORLD ENTITY IN SOFTWARE

- Meaningful to the business users
- Encapsulates both state and behavior

# BOUNDED CONTEXT

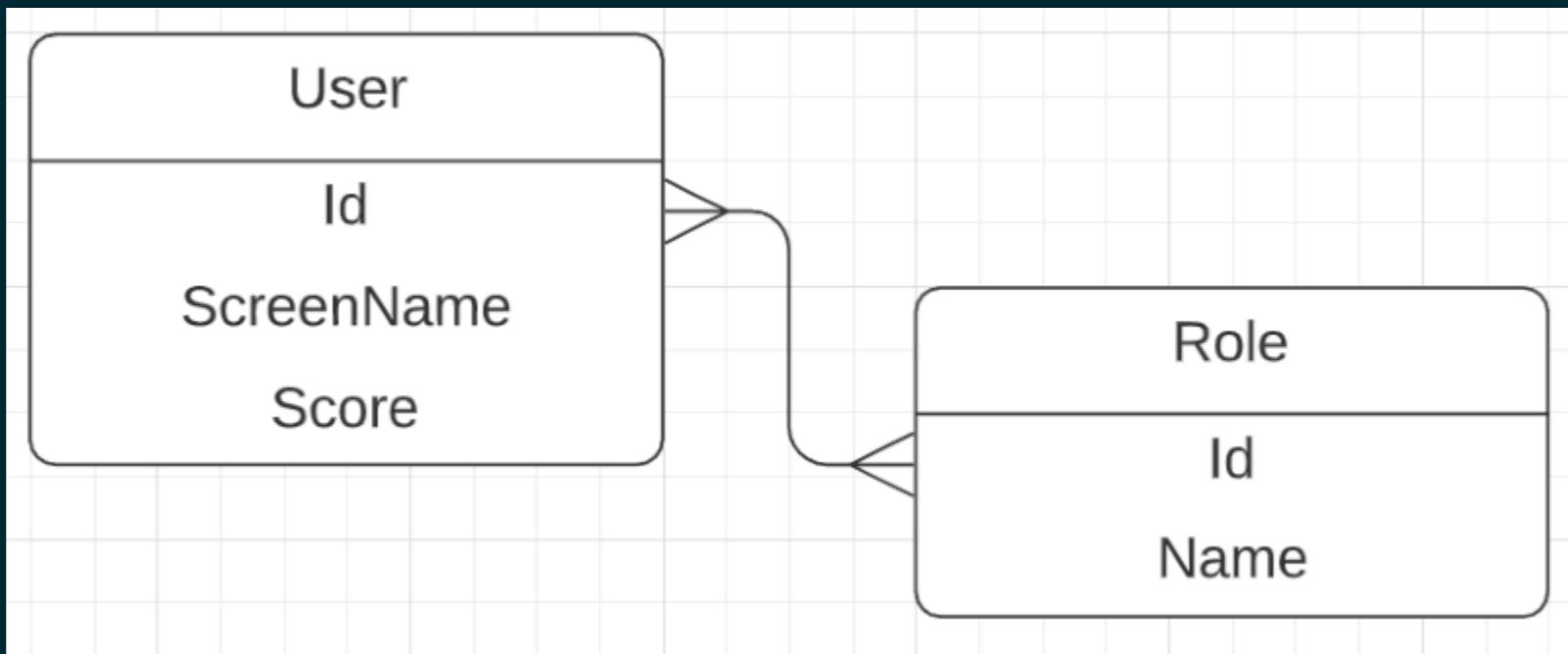
A SINGULAR VIEWPOINT ON THE FACTS AND CIRCUMSTANCES SURROUNDING A DOMAIN OBJECT

- Flight
  - Traveler's perspective
    - Departure & Arrival Times & Locations
    - Class of Service & Seat
    - Loyalty Miles
  - Gate Agent's perspective
    - Number of passengers
    - Boarding Time
    - Crew Members

# UBIQUITOUS LANGUAGE

| Term              | Definition  | Comments   |
|-------------------|---|--|
| Answer            | A response to a question that attempts to resolve the posed query such that the goal of the user in posting the query is satisfied  |  |
| Comment           | Text that can be attached to either a question or an answer, usually to ask about the question or answer in order to clarify the subject  |  |
| Domain            | The high-level subject of the Q & A site  | Examples: "Event Storming", "Python", "Buying a Used Car"                                      |
| Flagged           | The item has been marked to indicate that the flagging person feels the item is inappropriate or that some action may need to be taken by a Moderator   | Applies to Questions, Comments or Answers  |
| Moderator         | A persona with the ability to take certain actions on behalf of other users. This may include responding to a flagged item, marking an answer as accepted and other actions around protecting the integrity of the conversation |  |
| Question          | A query that is appropriate to the domain   |  |
| Question Searched | An indication that a particular question was returned in a set of search results  |  |
| Question Viewed   | An indication that the detail of the question was shown to a user   | Usually this means the user clicked on the title of the question from a list of search results |

# AGGREGATE (N)



A CLUSTER OF DOMAIN OBJECTS THAT CAN BE TREATED AS A SINGLE UNIT

# DOMAIN EVENTS

- A fact of significance to the business
- Described in the past tense
- Immutable (can't change history)
- Implementation Agnostic
- Domain Event ≠ Message

# SOME DOMAIN EVENTS

- Item Added to Inventory
- Flight Scheduled
- Video Uploaded
- User Logged-In



# THE 4 STEPS IN EVENT STORMING

1. Collect Domain Events
2. Refine Domain Events
3. Track Causes (Use-Case Modeling)
4. Aggregation (Sub-System Modeling)

# OUR PROBLEM DOMAIN - Q&A



# THE Q&A DOMAIN

- Access from Behind the Corporate Firewall only
  - Using the Company SSO solution only
- Users Can:
  - Search for Questions and Answers
  - View details of a Question and its Answers
  - Ask a Question
  - Answer an existing Question
  - Comment on existing questions and answers
  - Report an existing Question, Answer or Comment
- Activity can impact reputation

stack**overflow** Products  Ask Question

Home PUBLIC Questions Tags Users Companies COLLECTIVES Explore Collectives TEAMS Stack Overflow for Teams – Start collaborating and sharing organizational knowledge.

## Search Results

Results for what is domain driven design  
Search options **not deleted**

1,293 results Relevance Newest More ▾

**What is Domain Driven Design?** 214 votes 9 answers 105k views domain-driven-design Calanus 24.7k asked Mar 16, 2011 at 13:15

Can somebody please explain (in succinct terms) what exactly is domain driven design? I see the term quite a lot but really don't understand what it is or what it looks like. ... How does it differ from non-...

**What is Domain Driven Design (DDD)? [closed]** 305 votes 2 answers 77k views domain-driven-design YodasMyDad 8,998 asked Aug 3, 2009 at 13:33

I keep seeing DDD (Domain Driven Design) being used a lot in articles - I have read the Wikipedia entry about DDD but still can't figure out what it actually is and how I would go about implementing it ...

stackoverflow Products Search...  
Home PUBLIC Questions Tags Users Companies COLLECTIVES Explore Collectives TEAMS Stack Overflow for Teams – Start collaborating and sharing organizational knowledge. Create a free Team Why Teams? 9 Answers EDIT: 117 As this seem to be a top result on Google and my answer below is not, please refer to this much better answer: <https://stackoverflow.com/a/1222488/1240557>  
OLD ANSWER (not so complete :))  
In order to create good software, you have to know what that software is all about. You cannot create a banking software system unless you have a good understanding of what banking is all about, one must understand the domain of banking.  
From: Domain Driven Design by Eric Evans.  
This book does a pretty good job of describing DDD.

# THE Q&A DOMAIN

- Access from Behind the Corporate Firewall only
  - Using the Company SSO solution only
- Users Can:
  - Search for Questions and Answers
  - View details of a Question and its Answers
  - Ask a Question
  - Answer an existing Question
  - Comment on existing questions and answers
  - Report an existing Question, Answer or Comment
- Activity can impact reputation

# STEP 1 - COLLECT DOMAIN EVENTS

- Brainstorm Domain Events
  - Use orange stickies for each one
  - Conversations around events are encouraged
- Identify questions or concerns
  - Use pink stickies liberally

# "INFINITE" DESIGN SURFACE



# EVENTS COLLECTED

- Notable Events
  - User Logged-in
  - Question Asked
  - Question Answered
  - Answer Upvoted
  - Comment Submitted
- Interesting Issues
  - Can a question be edited?
  - Can a comment be upvoted?
  - What happens with duplicates?
  - What is actually searched for?

User  
Logged-In

Question  
Asked

Can a  
question be  
edited? By  
whom?

Comment  
Submitted

Can you  
upvote or  
like a  
comment?

Question  
Answered

What  
happens if  
it is a  
duplicate?

Answer  
Upvoted

Can an  
answer be  
downvoted?

What do we  
actually  
search for?  
Keywords?

Question  
Searched  
for

Question  
Viewed

Report  
Submitted

What  
happens to  
reported  
items?

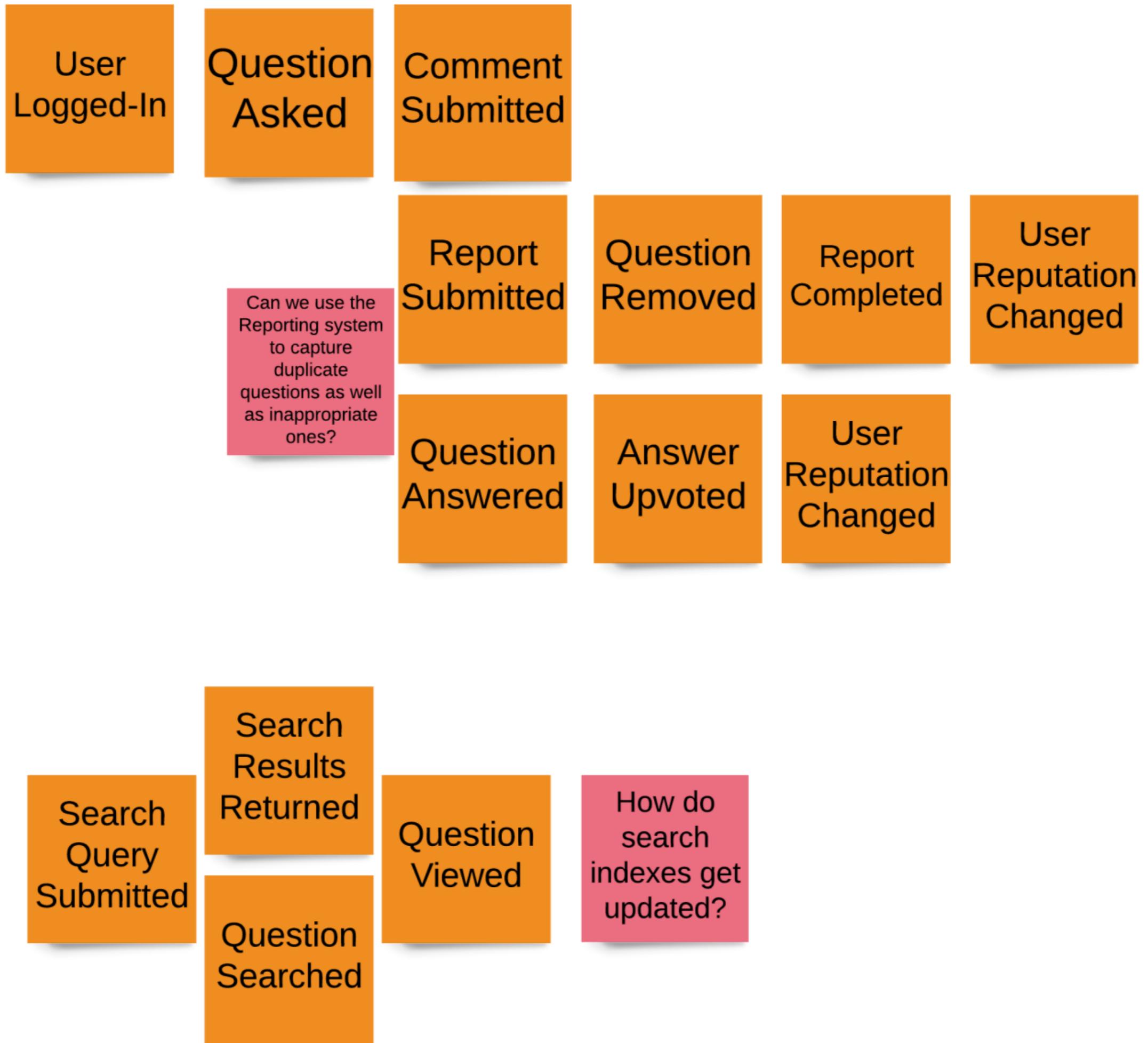
User  
Reputation  
Changed

# STEP 2 - REFINER EVENTS

- Revisit each event as a team
  - Grow shared understanding for each
  - Add any additional Events that are discovered
  - Resolve pink stickies if possible
  - Leave pink stickies that are unresolved
  - Add new pink stickies for additional questions

# EVENTS REFINED

- Notable Changes
  - Events roughly ordered
  - Added Question Removed event
  - Added Report Completed event
  - Filled-In Ubiquitous Language document
- Issues Resolved
  - Questions can not be edited -- comment instead
  - No upvotes of comments or questions (perhaps future)
  - Searches are for keywords and phrases



# STEP 3 - TRACK CAUSES

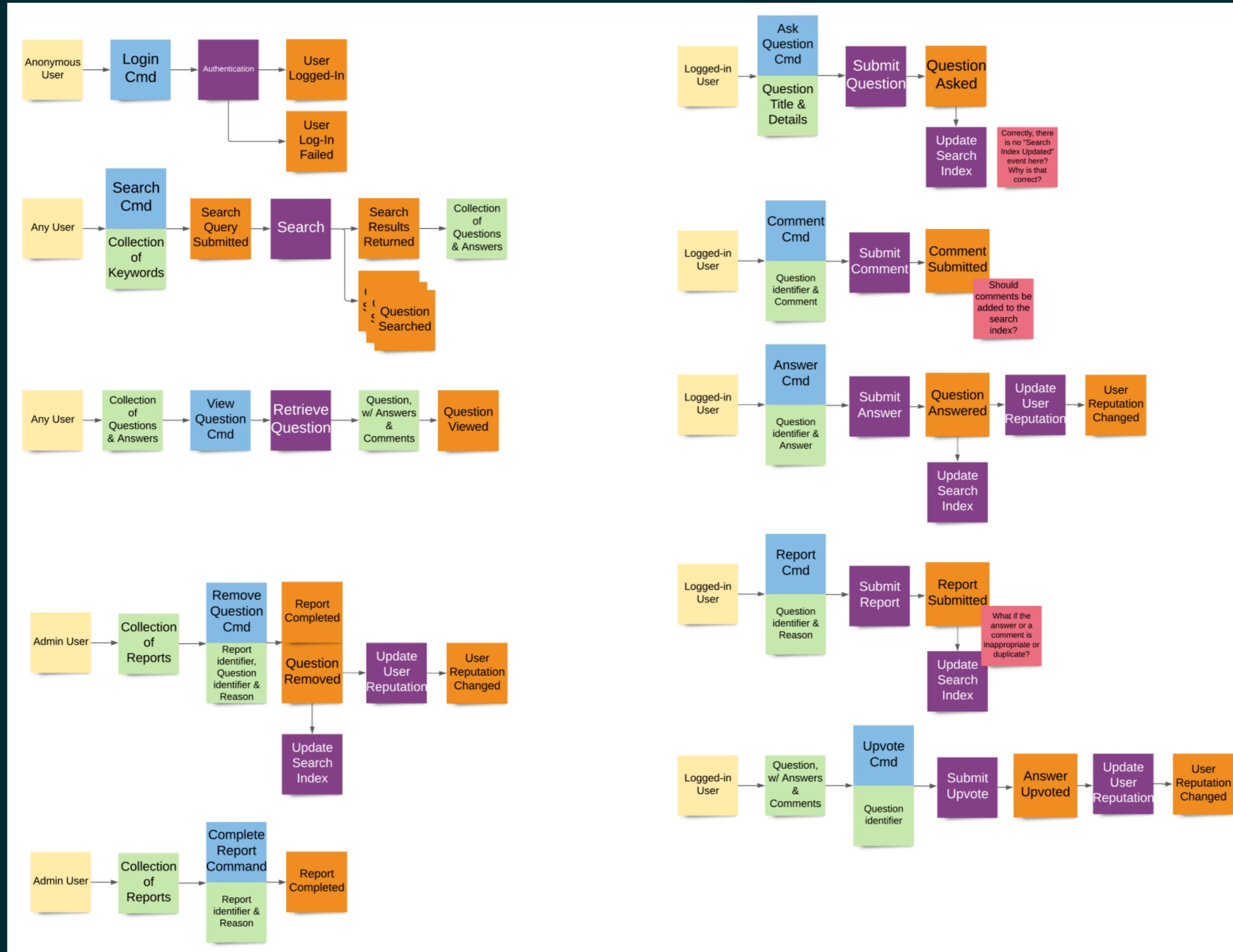
- User or Actor
  - A human interacting with the system
  - May utilize View Models
  - May issue Commands
- Business Process
  - Logic internal to our systems
  - May utilize Read Models
  - May trigger other Events
- External Systems
  - Logic from outside our organization
  - May utilize View Models
  - May trigger Business Processes

# AFTER TRACKING CAUSES

- Identify Sources
  - Actor
  - Business Process
  - External System
- Identify Interactions
  - Which Users trigger Commands?
  - Which elements trigger Business Processes?
  - Which elements utilize Read Models?
  - Which Business Processes create new Events?

# CAUSES TRACKED

- Notable Flows
  - User log-in
  - Search
  - View Question
  - Ask Question
  - Answer Question
  - Submit Comment
  - Submit Answer
  - Upvote Answer
  - Report Question
  - Remove Question
  - Close Report

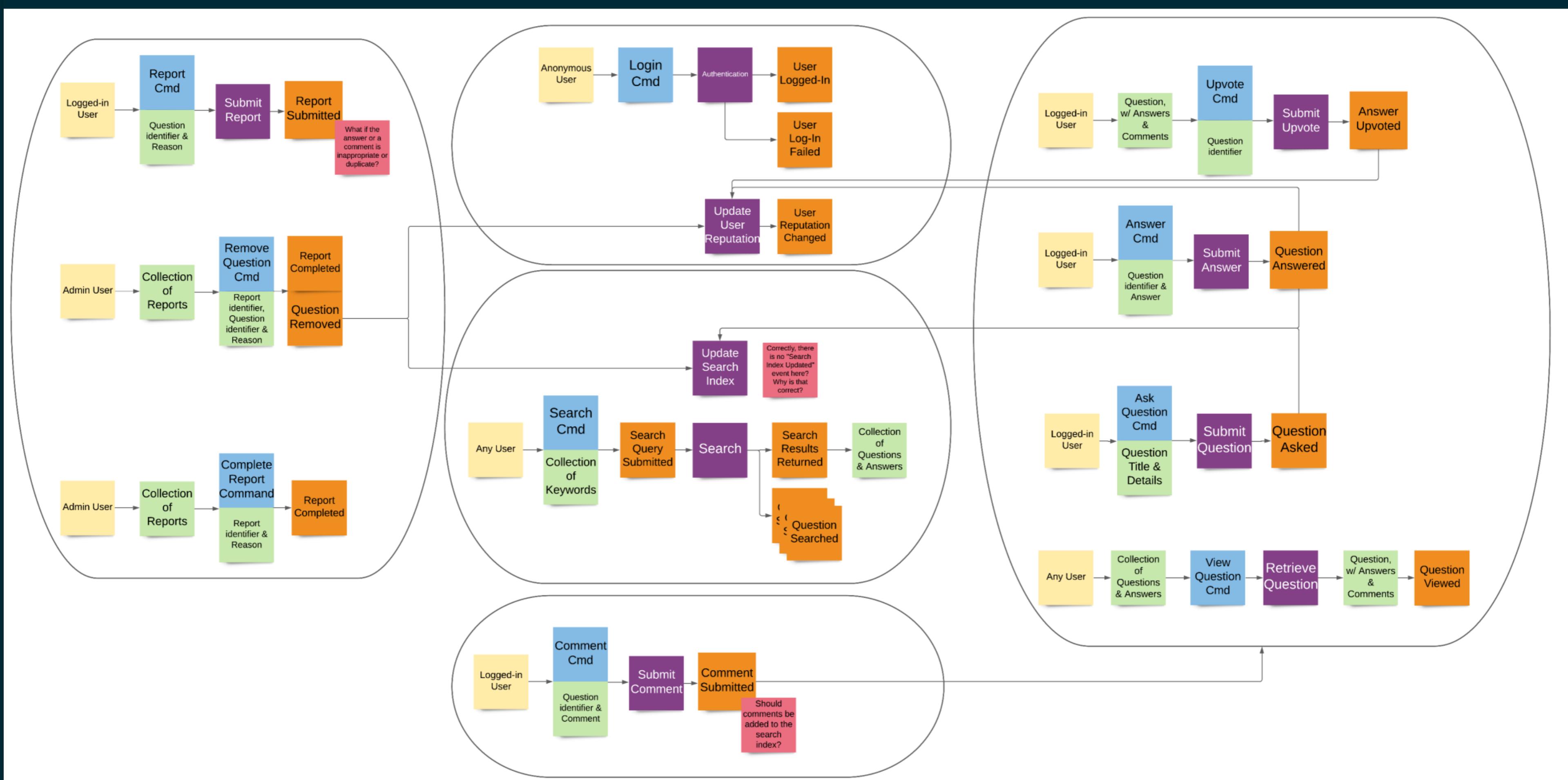


# STEP 4 - AGGREGATE (V)

- Group the modeled objects by their Aggregate Root
  - Identify the key subsystems in the model
    - Subsystem is often a good proxy for μservice
  - Err on the side of greater separation
    - It is easier to combine than separate later

# AGGREGATES AGGREGATED

- Notable Aggregates
  - User
  - Report (moderation)
  - Search
  - Comment
  - Question



# RECOMMENDATIONS



# BLOCK THE TIME UP-FRONT

- Size and complexity of the domain
- Number and the assertiveness of participants

# START IN-PERSON THEN GO VIRTUAL

- Step 1 = many conversations
- Steps 2-4 = 1 conversation

You'll need to be able to transfer the physical surface to a virtual whiteboard

# HAVE A FACILITATOR

- The loudest voices should be from the business
- Technical voices should not be authoritative

# DON'T FALL IN LOVE WITH THE MODEL

- It will hopefully be improved on
- It will almost certainly change

# UPDATE THE UBIQUITOUS LANGUAGE OFTEN

- Facilitators - call attention to the UL often
- Use a spreadsheet for sortability

# DRAWING SEMANTICS ARE NOT CRITICAL

- There is no "right way" to draw these pictures
- Understanding is the important thing

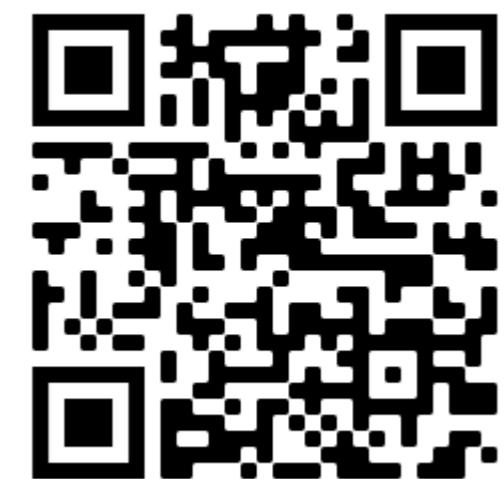
# SUMMARY OF RECOMMENDATIONS

- Block the time up-front
- Start in-person, then go virtual
- Have a facilitator
- Don't Fall in Love with the Model
- Update the Ubiquitous Language often
- Drawing semantics are not critical

# SUMMARY

- Modeling process from the perspective of the business experts
  1. Collect Events
  2. Refine Events
  3. Track Causes
  4. Aggregation (Sub-System Modeling)
- Powered by Domain Events
  - Triggered by Users, Business Processes and External Systems
  - Supported by Commands and Read Models
- Produces
  - Visual Model of the Domain
  - Ubiquitous Language
  - Shared Understanding

# RESOURCES



- <http://IntroToEventStorming.azurewebsites.net>
- Printable Version of these slides
- Eric Evans on Domain Models
- Alberto Brandolini on Event Storming
- Q&A Domain - Sample Event Storming
- Lucid Chart