

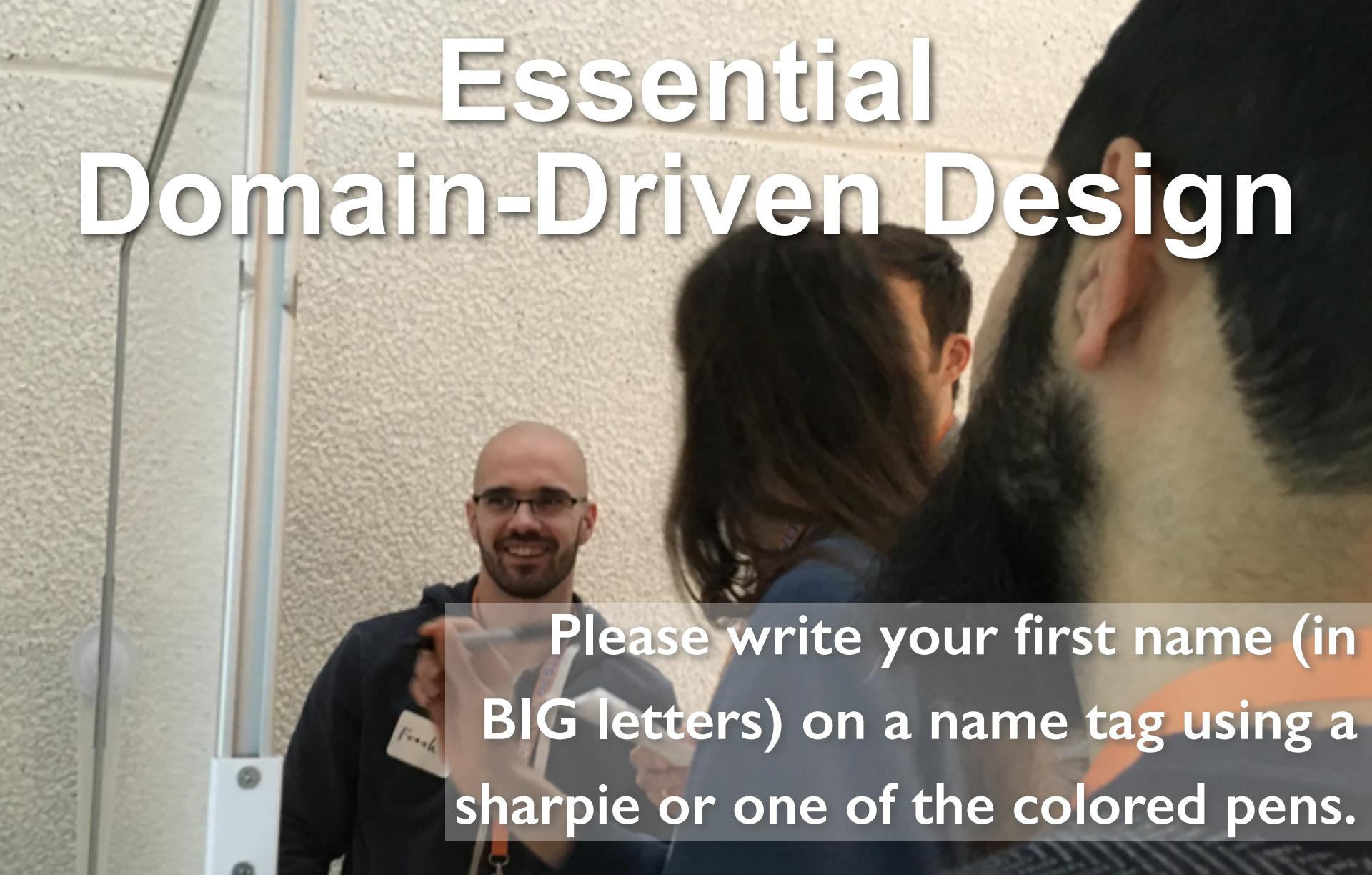
WELCOME!

Sit with different people today. Each person share something that you are good at outside of work.

Tear out the last page of your workbook (with the contact info) and fold it so it makes 8 equal-sized boxes.

Review yesterday's content and write a word or phrase to define/describe (but not the definition) in each box on the page.

Essential Domain-Driven Design

A photograph showing several people in what appears to be a conference or workshop setting. In the foreground, a man with glasses and a beard is smiling towards the camera. Behind him, other individuals are visible, some looking down at papers or devices. The background is a plain, light-colored wall.

Please write your first name (in BIG letters) on a name tag using a sharpie or one of the colored pens.

Paul Rayner
 [@the paul rayner](https://twitter.com/the paul rayner)



Design is Inevitable

“Questions about whether design is necessary or affordable are quite beside the point...The alternative to good design is bad design, not no design at all.”

— Douglas Martin

Adaptive, not Predictive

“A good design is not the one that correctly predicts the future, it’s one that *makes adapting to the future affordable*.

— Venkat Subramaniam

Extended Warranties



Technical staff learn to
speak the business
(domain) language...

John,

In the system for Client A, please tell me the source for populating the field Retail Price?

Pam

Accountant

Pam,

Retail Price = **ARRAMT** in **ARRTRXPF** ...

It comes out of the **BARCSTPF** ... either directly or as calculated percentage from **BARCSTPF**.

John

Developer

Claire <Business Analyst>,

**Can you please tell me what the @*^# field
John is referring to?**

I haven't memorized all of the column headers in the
tables in the system that I never see . . . :-)

I'm guessing he means the product maintenance menu,
and the screens 1 - 3 and the field marked "sale price" . .

.

Pam
Accountant



Tell us a story...



The Dishwasher



Domain Model

A set of concepts
expressing distilled
knowledge, rules,
assumptions, concepts
about a business domain

Domain Model

It's about usefulness,
not realism
usefulness relative to a
particular set of scenarios

Reference Scenario

A business example
that tells a story
about the business
need and what makes
the software worth
writing



Assertions about this Model

- Passengers may only access transport at stations or stops
- Passenger transport follows transit lines
- Transit line length does not convey relative distance or direction
- Transit lines terminate at last station



“...the *most significant complexity* of many applications is *not technical*. It is in the domain *itself*, the activity or business of the user.”

“A successful design

*must systematically deal with this business
domain complexity*

as the

central aspect of the software.”

- Eric Evans

Ubiquitous Language

A language structured around the domain model,
used by all team members
to connect all the activities of the team with the software

Context

The conditions under which a particular model is defined and applicable.

Bounded Context



Clues to Identifying Bounded Contexts

Look for:

- different teams
- differences in terminology
- Boundaries:
 - Subsystem
 - System
 - Component
 - Organizational

Ubiquitous Language

...within a
Bounded
Context

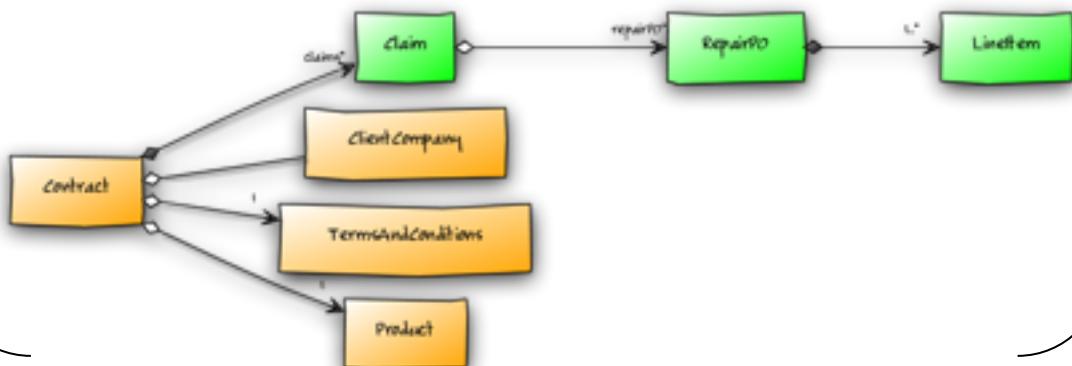
7. LIMIT OF LIABILITY:

The total amount that We will pay for repairs or replacement made in connection with all claims that You make pursuant to this Protection Plan shall not exceed the

```
public double limitOfLiability()
{
    double claimTotal = 0;
    List<Claim> claims = new ArrayList<Claim>();
    claims.addAll(this.getClaims());

    for (Claim claim:claims)
    {
        claimTotal += claim.amount;
    }

    return (this.purchasePrice - claimTotal) * 0.8;
}
```



Review the front side of the User Stories Reference Guide. Then...

talk in your table group about:

- **what you find interesting**
- **ways to better integrate design into your team process.**

How did diagramming a specific example together using an instance diagram help:

- surface hidden assumptions?
- clarify the domain?
- drive shared understanding?

How did EventStorming help:

- surface hidden assumptions?
- clarify the domain?
- drive shared understanding?

How might you apply this technique in your work?

- A. Outsource development to an offshore team [Core | Generic | **Supporting**]
- B. Buy an off-the-shelf software package [Core | **Generic** | Supporting]
- C. Do model exploration [**Core** | Generic | Supporting]
- D. Do lots of experiments and make many mistakes [**Core** | Generic | Supporting]
- E. *Never* buy an off-the-shelf software package [**Core** | Generic | Supporting]
- F. Use an open source framework [Core | **Generic** | Supporting]
- G. Collaborate intensively with the domain experts [**Core** | Generic | Supporting]
- H. Bring in a team of contractors [Core | Generic | **Supporting**]
- I. Hire an intern to do the work [Core | Generic | **Supporting**]
- J. Isolate in an explicitly bounded context [**Core** | Generic | Supporting]

- Product Replacements only happen for certain types of Covered Products. Depends on the Warranty type.
- Not always a store credit.
- Purchase Orders are used as a tracking mechanism for us, and a communication mechanism with the Servicer Management Company



Flesh out Customer Payment

Walk through the model together...

Technical and domain experts

Be realistic/concrete

Open-ended questions

Ask for examples

Be specific

Go broad if necessary

**Update the ubiquitous
language**

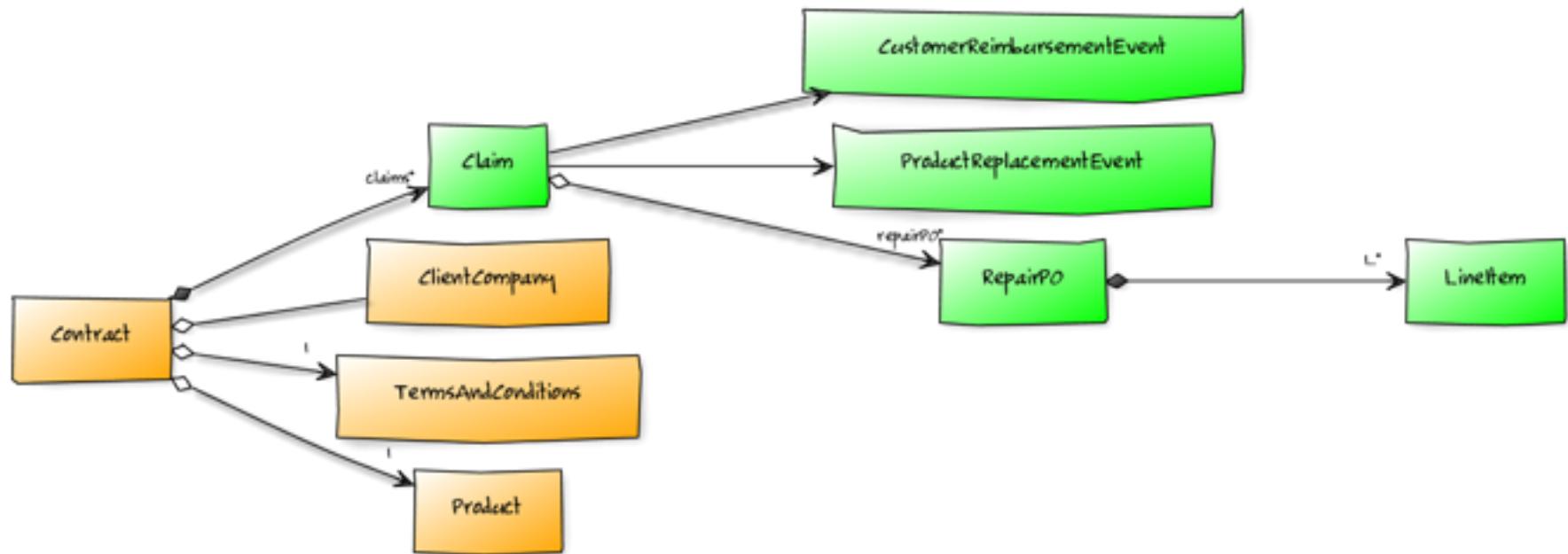
Make concepts explicit

**It's *exploration*, so be
humble and make
mistakes**

Stress Points

- “Reimbursement,” not “Payment”
- Creation of (unnecessary) Purchase Orders and Line Items
- Contract Fulfillment should be triggered by (one-time only) Customer Reimbursement/Product Replacement
- Not a check, a *store credit*. But would like to support at check later on.

Incorporate the Learning



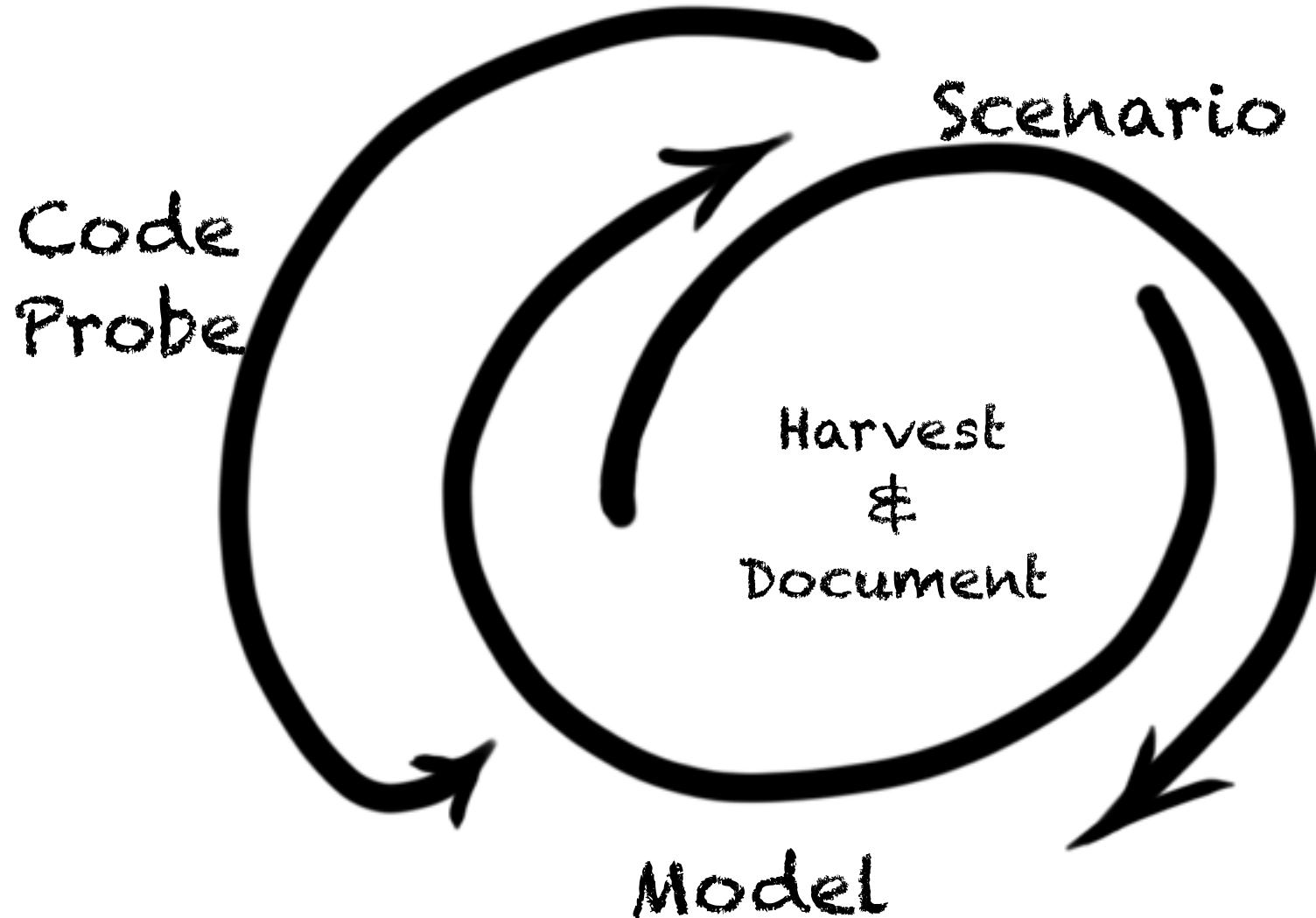
WELCOME!

Sit with different people today.

Review yesterday's content and come up with at least 5 short quiz questions (one per sticky) based on what you learned.

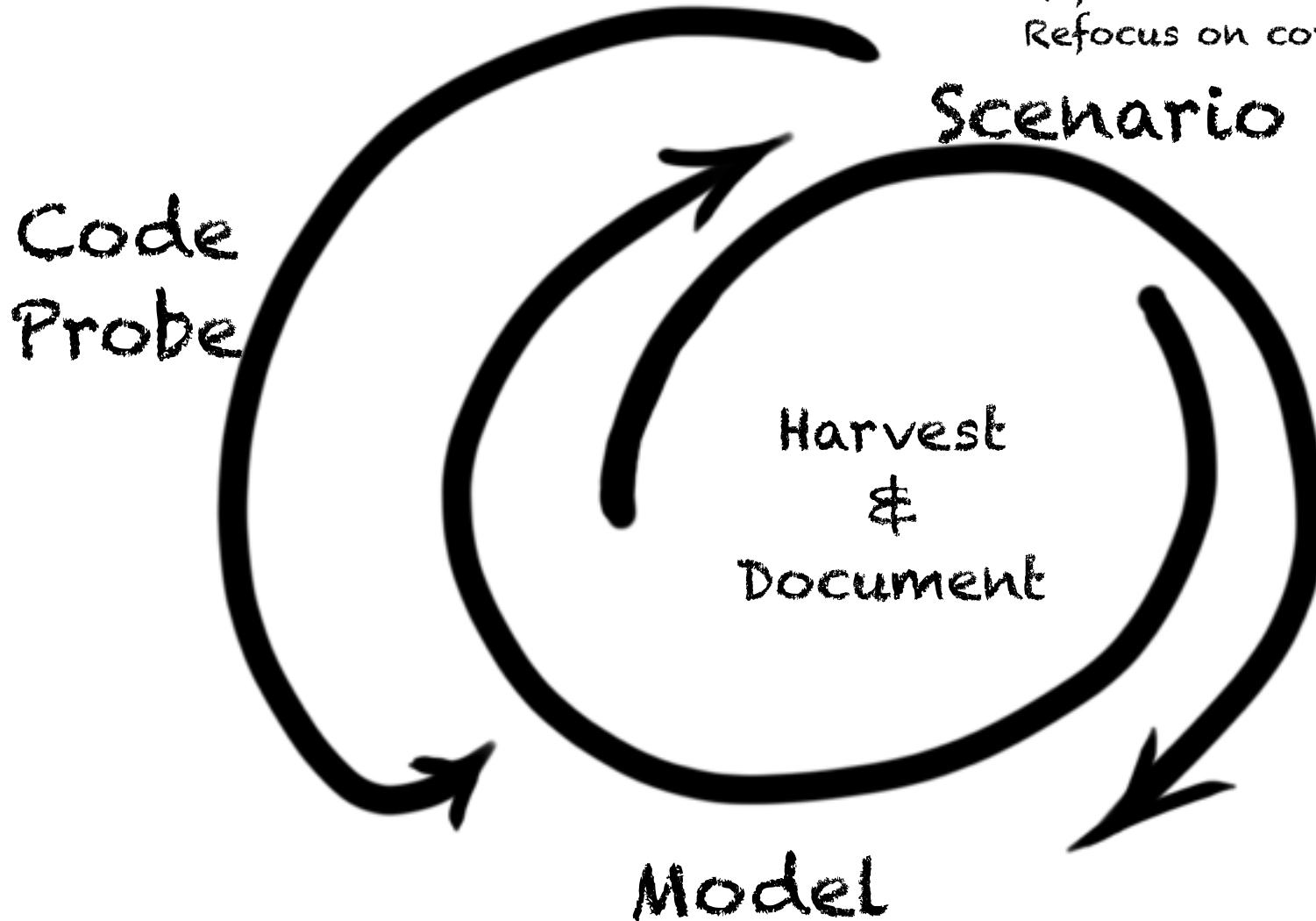
Model Exploration Whirlpool

domainlanguage.com/ddd/whirlpool

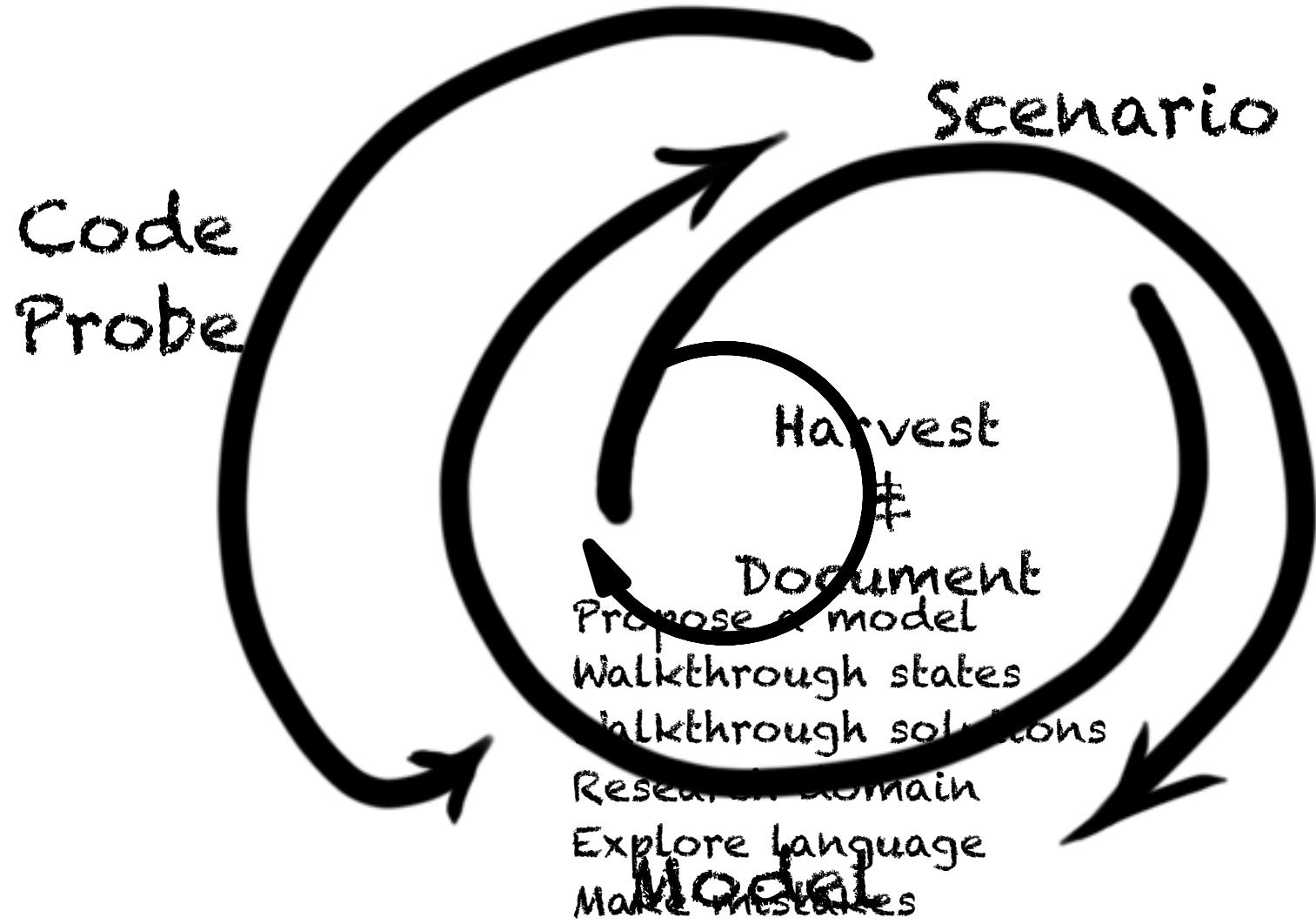


Model Exploration Whirlpool

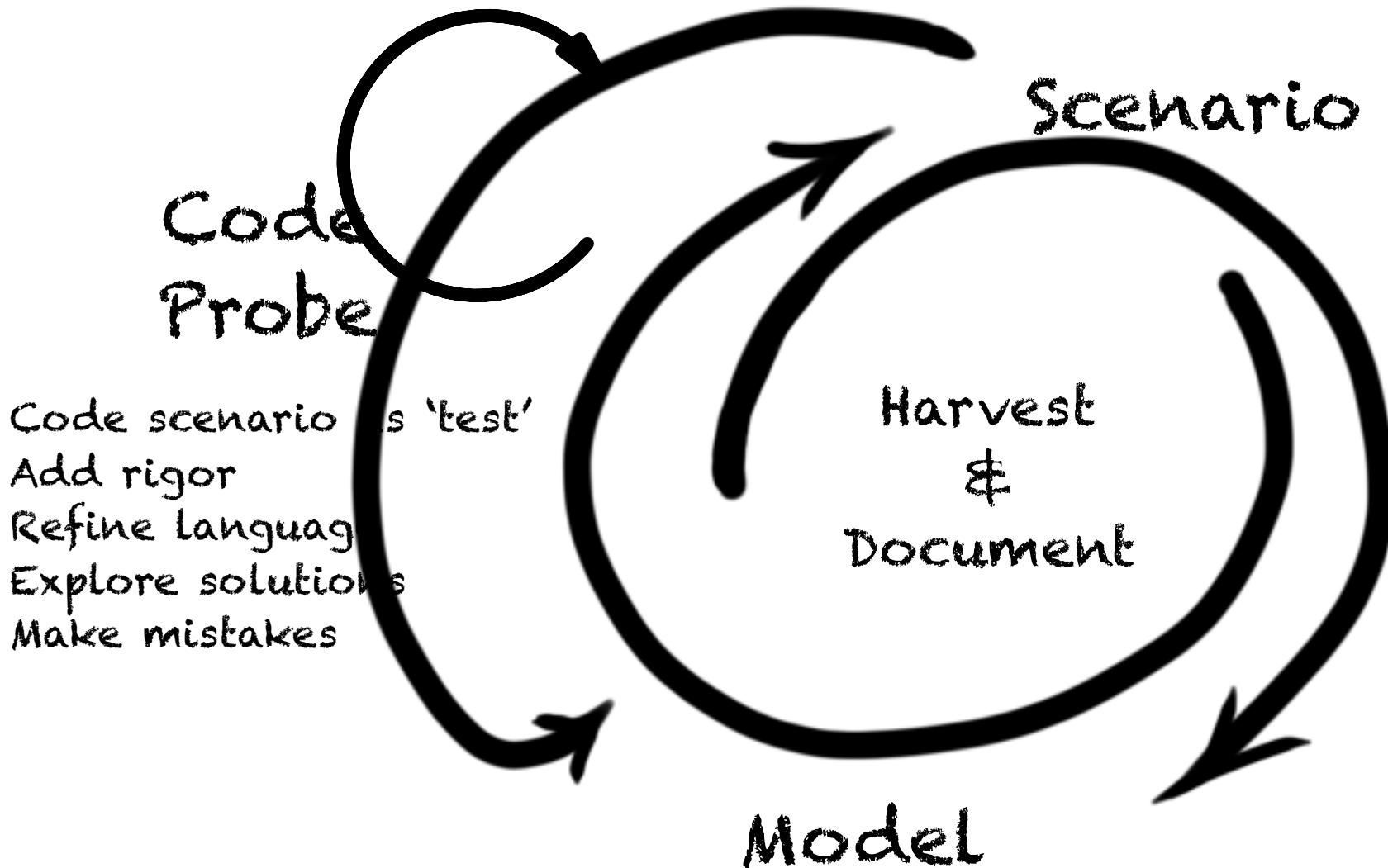
Tell us a story
Flesh it out
Refocus on the hard part
Refocus on core domain



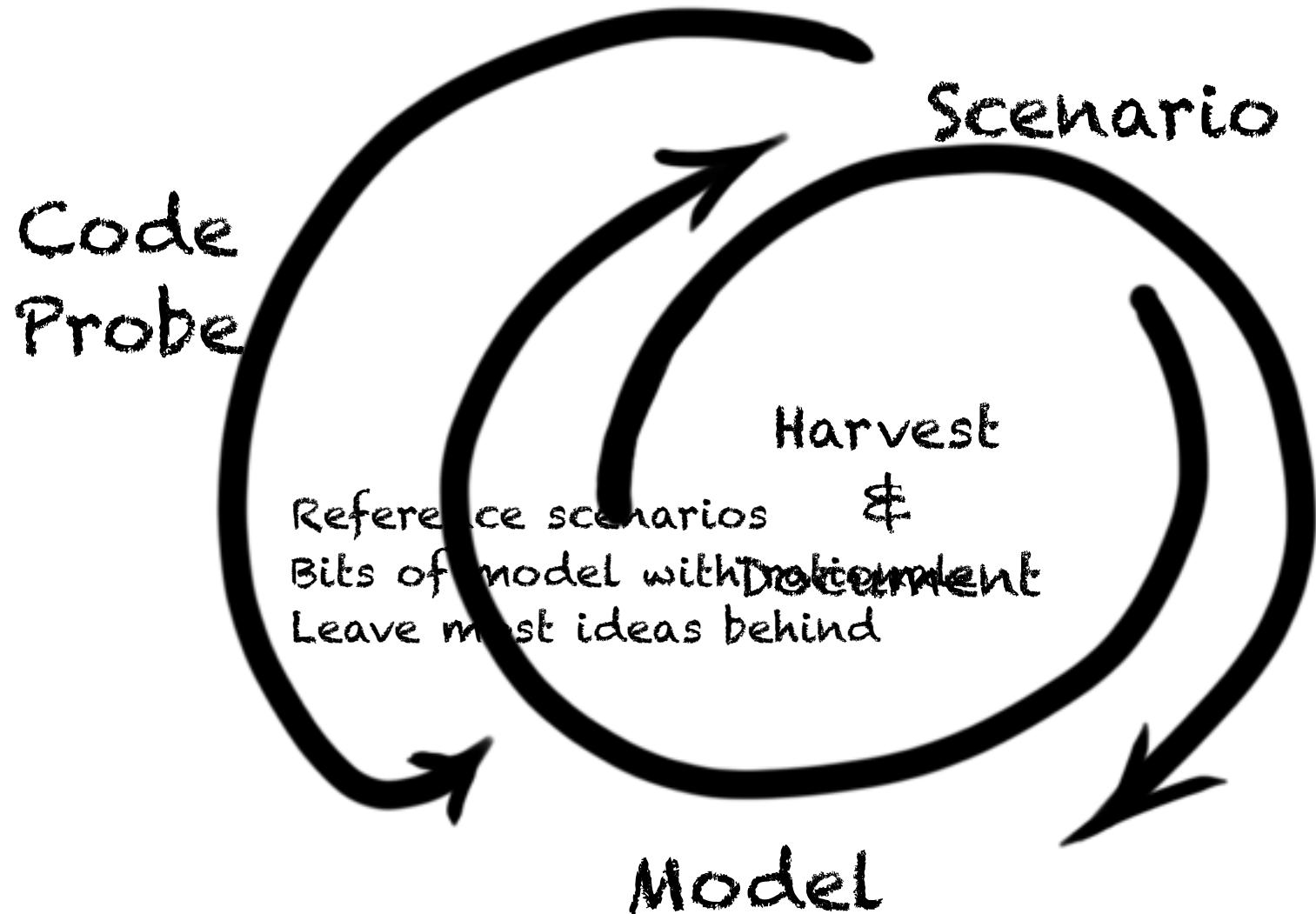
Model Exploration Whirlpool



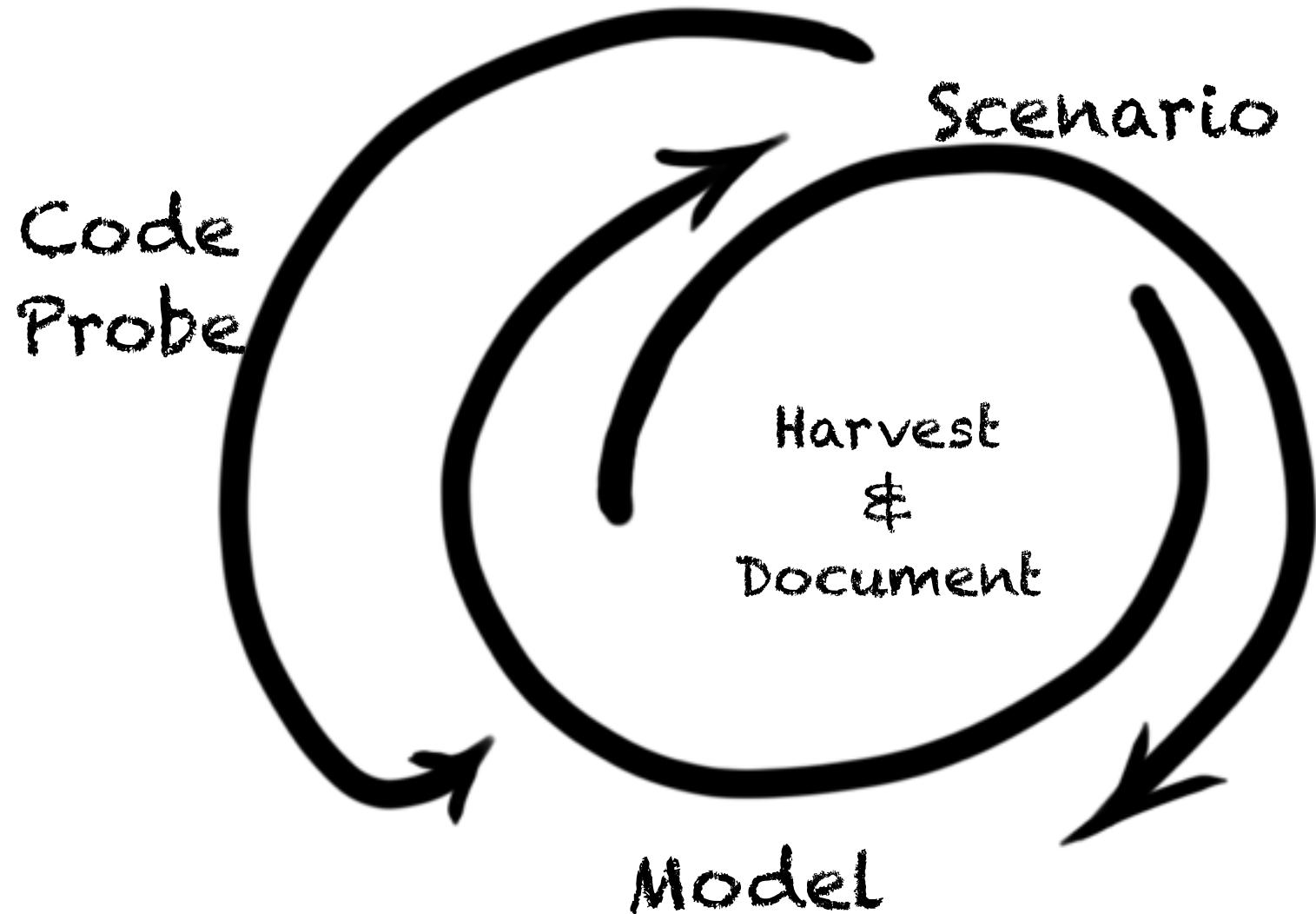
Model Exploration Whirlpool



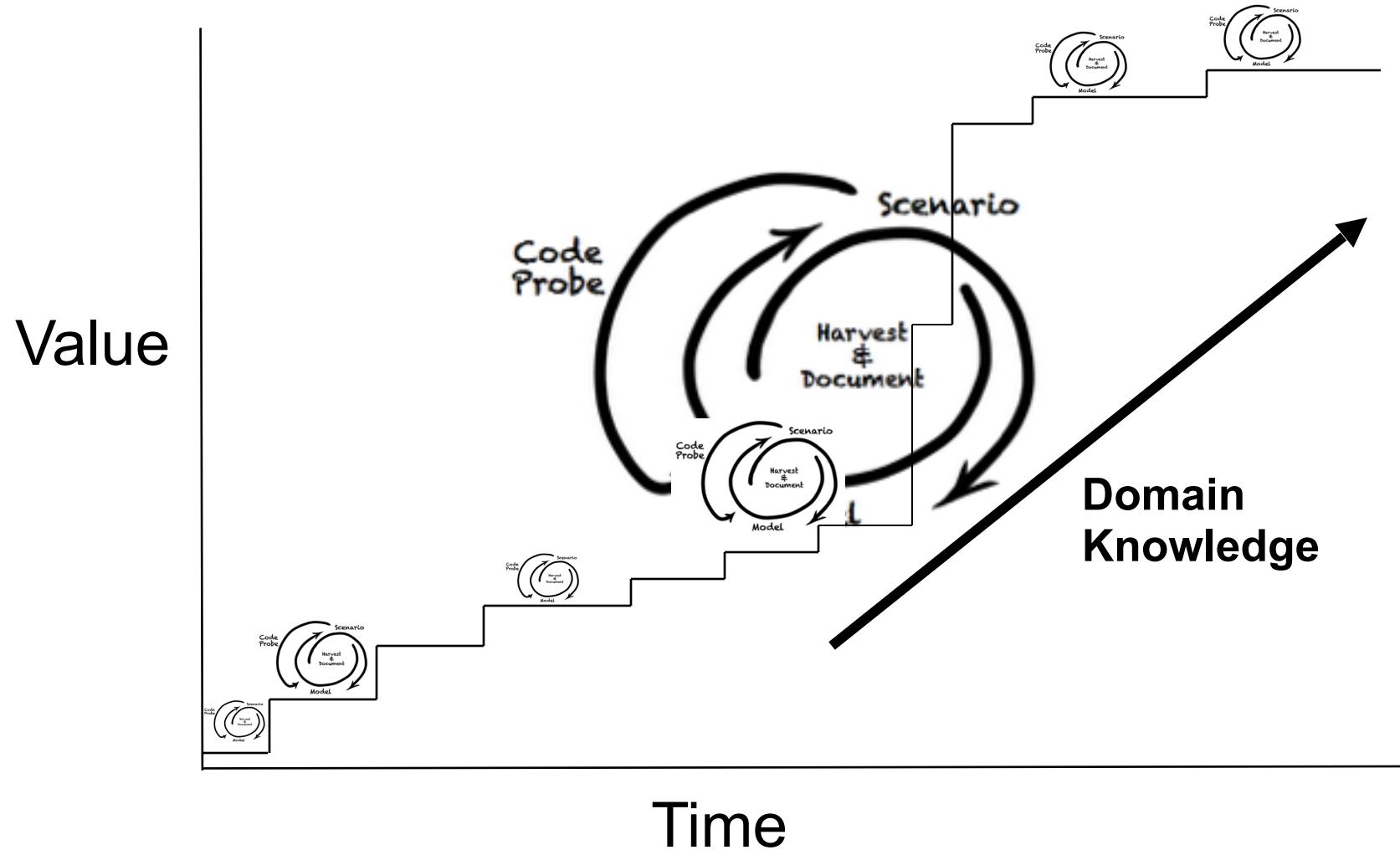
Model Exploration Whirlpool



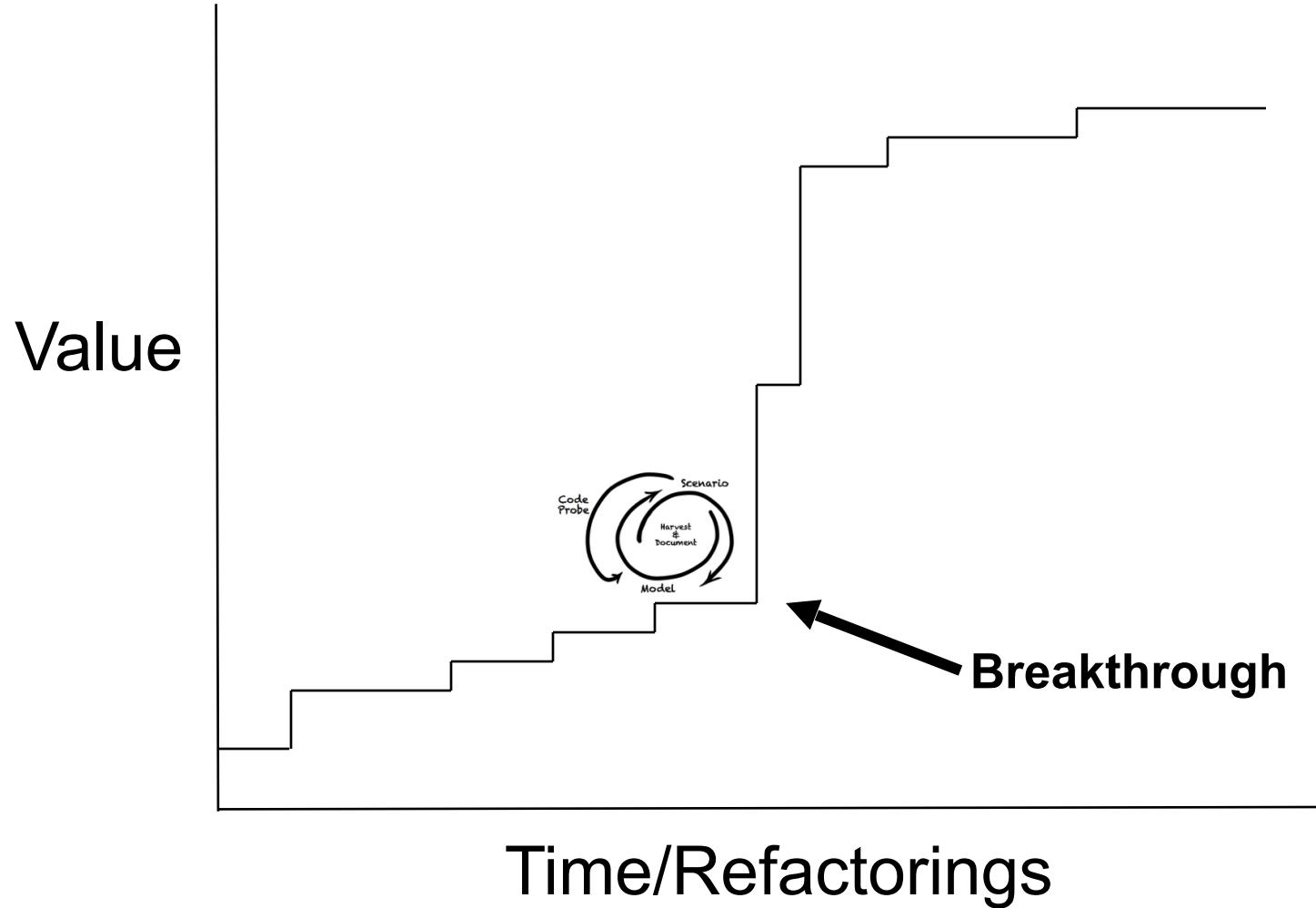
Model Exploration Whirlpool



Iterative Design + Incremental Development = Accelerated Product Delivery



Refactoring to a Deeper Model



4 strategies for getting started with DDD when you have a big commitment to legacy systems

Based on <https://domainlanguage.com/ddd/surrounded-by-legacy-software/>

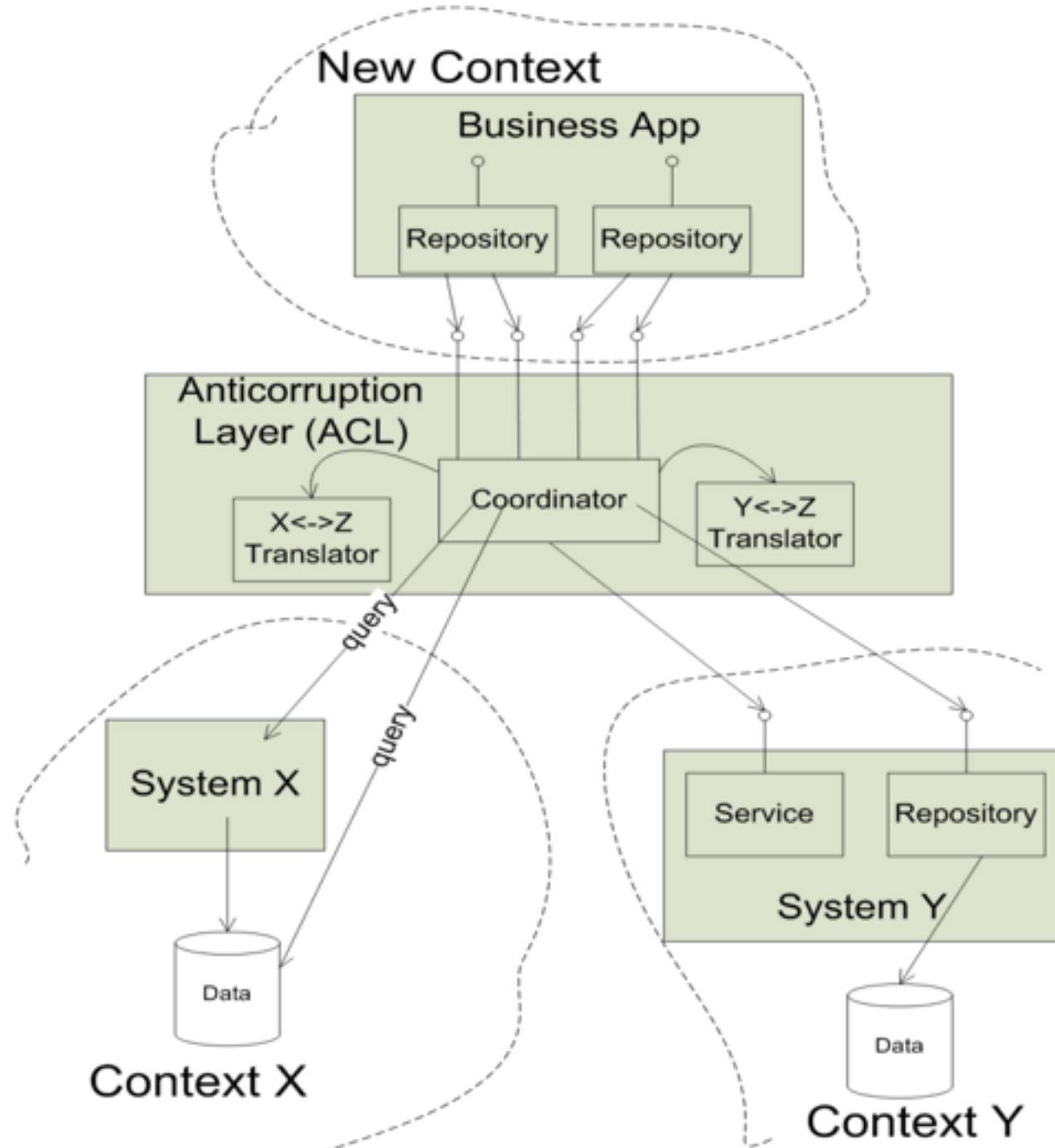
1. Bubble Context*
2. Autonomous Bubble
3. Exposing Legacy Assets as Services
4. Expanding a Bubble

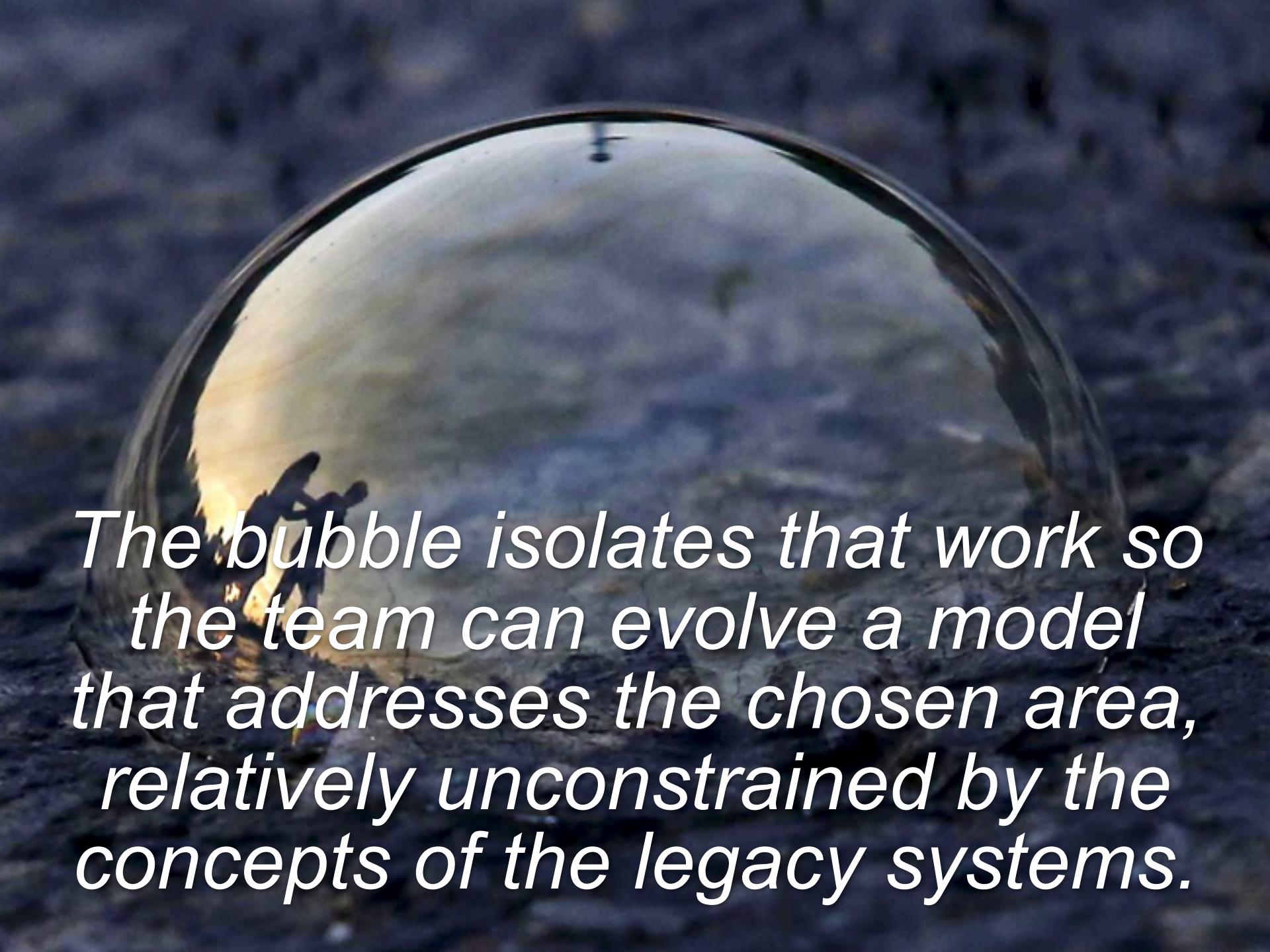
* Introducing a difficult new set of development principles and techniques is best done incrementally, as in a pilot project, in a way that allows members of the team to gain experience and allows the organization to assess the approach. Bubble Context is a nice way to do this.

- 1. Bubble Context***
- 2. Autonomous Bubble**
- 3. Exposing Legacy
Assets as Services**
- 4. Expanding a Bubble**

How we form the Bubble:

- Choose an important, yet modest-sized, business-related problem with some intricacy.
- Small, self-disciplined team with control over its code.
- Establish context boundary for the bubble using an ACL to isolate from the legacy BBOM.



A large, translucent bubble floats over a dark, stormy sea at sunset. The sky is filled with dramatic, orange and yellow clouds. The horizon shows a bright sun setting behind the clouds. The water below is dark and choppy. The bubble is perfectly clear, reflecting the light from the sunset and the surrounding clouds.

*The bubble isolates that work so
the team can evolve a model
that addresses the chosen area,
relatively unconstrained by the
concepts of the legacy systems.*

Introducing a difficult new set of development principles and techniques *is best done incrementally, as in a pilot project*, in a way that allows members of the team to gain experience and allows the organization to assess the approach. Bubble Context is a nice way to do this.

- Eric Evans

Main Characteristics of the Bubble Context Strategy:

- Modest commitment to DDD
- No synchronization risk (uses legacy database)
- Works when there is a limited range of data needed from legacy
- At some point the bubble bursts

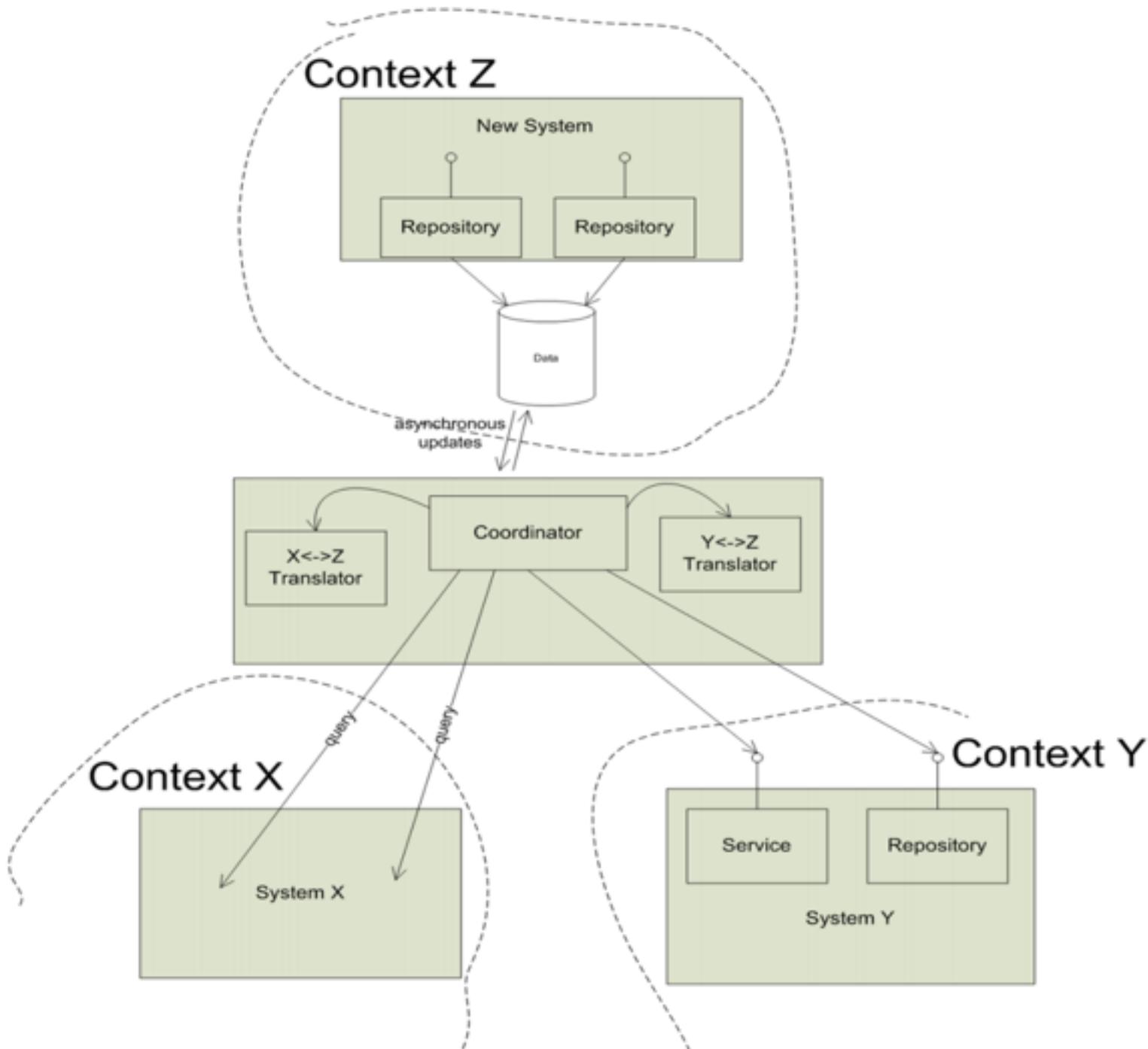
1. Bubble Context*
2. Autonomous Bubble
3. Exposing Legacy
Assets as Services
4. Expanding a Bubble

3:36
HOME 0 PERIOD 1 GUE



Main Characteristics of the Autonomous Bubble Strategy:

- Needs organization committed to significant new development with new design approach.
- Allows new data to be collected and used without involving legacy.
- Allows decoupled "product" to evolve.
- May progress from Bubble Context as development is scaled up on a successful pilot project.



Synchronizing ACL

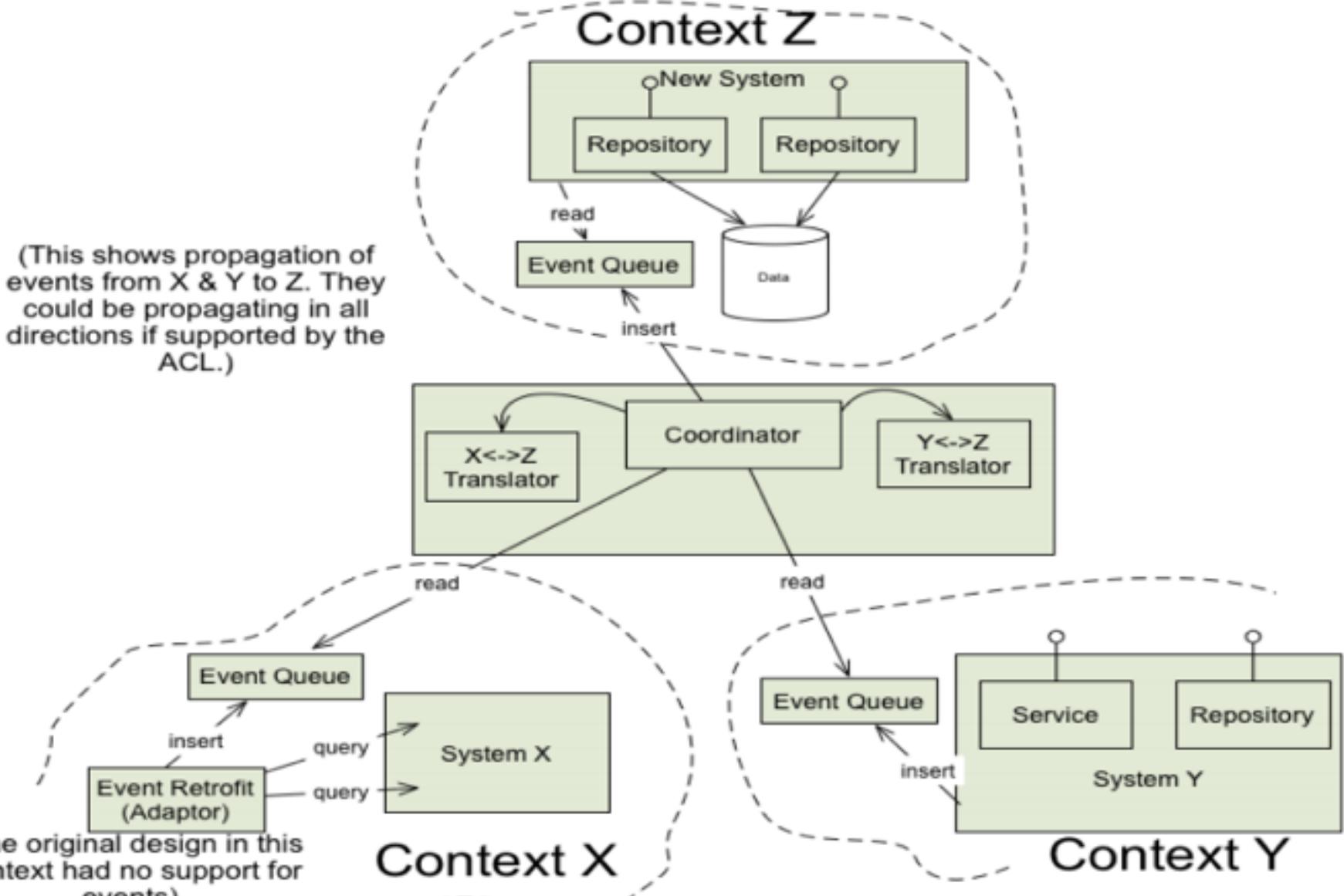
The ACL takes on the responsibility of synchronization between data stores in two contexts (which do not depend directly on each other).

This ACL activity is asynchronous with any activity in either context.

Define the service level agreement (SLA) regarding the freshness of the translated data.

Synchronizing ACL Approaches

- **Simple:** Nightly batch job
- **More Flexible:** Messages carrying domain events



1. Bubble Context*
2. Autonomous Bubble
3. Exposing Legacy Assets as Services
4. Expanding a Bubble

Don't waste time
recreating capabilities
that already exist,
especially in generic
and supporting
subdomains.

Use Open Host
Services* to make
legacy capabilities
more accessible

*Can be ACL-backed or synchronizing (with its
own data store)

1. Bubble Context*
2. Autonomous Bubble
3. Exposing Legacy
Assets as Services
4. Expanding a Bubble

Expanding a Bubble

1. Coevolving model and ACL
2. Only bring in data you use!
3. Adding data is a modeling job