

# Strategic Design

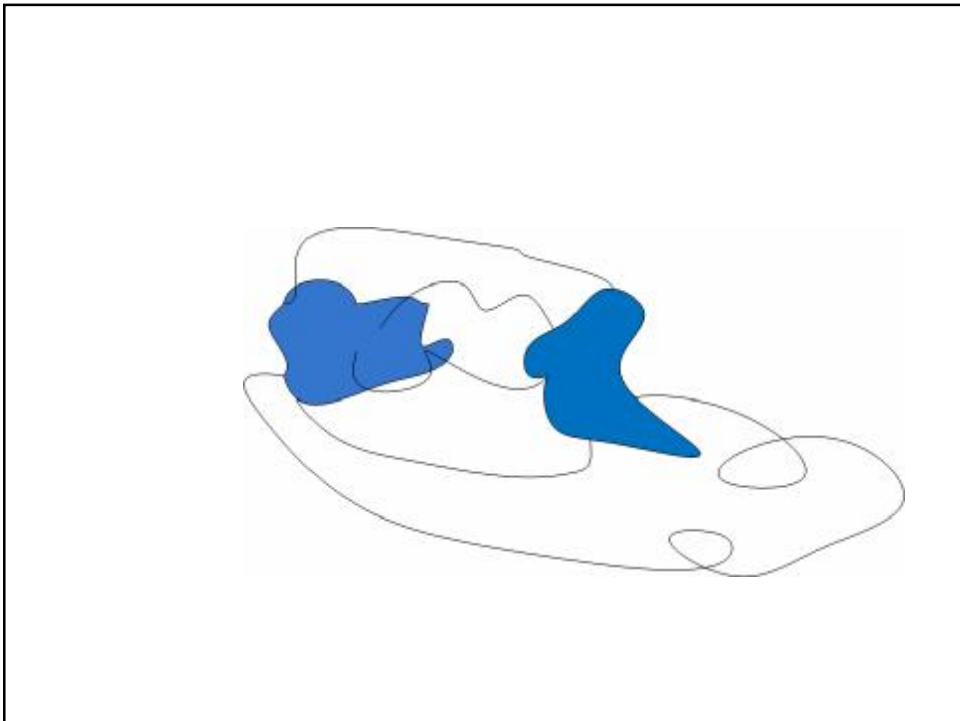
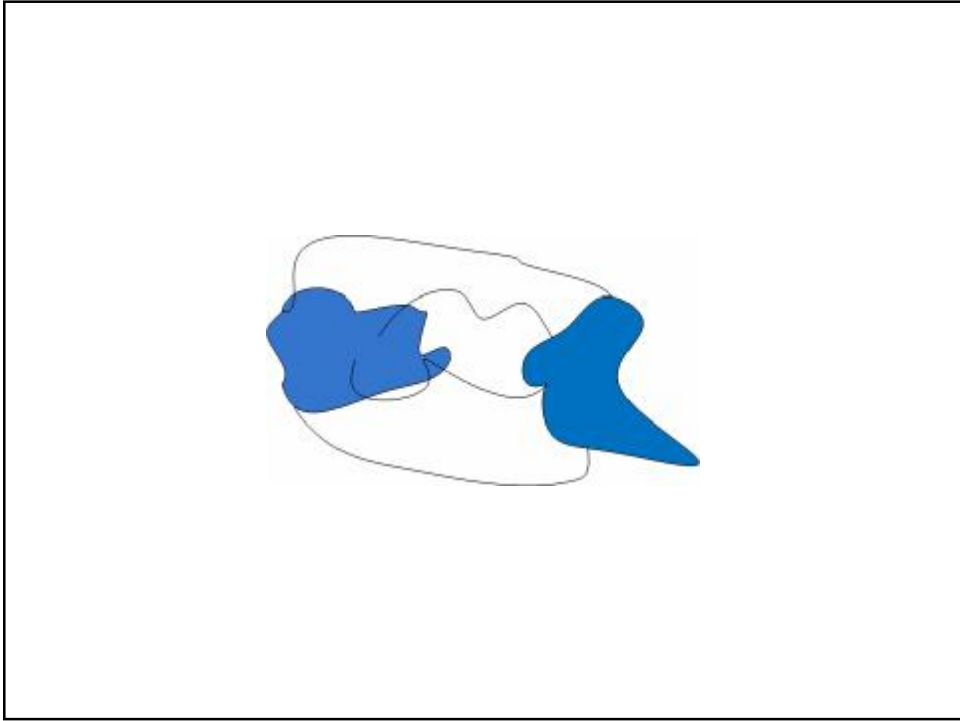
And Avoiding Traps That Snare  
Responsible People

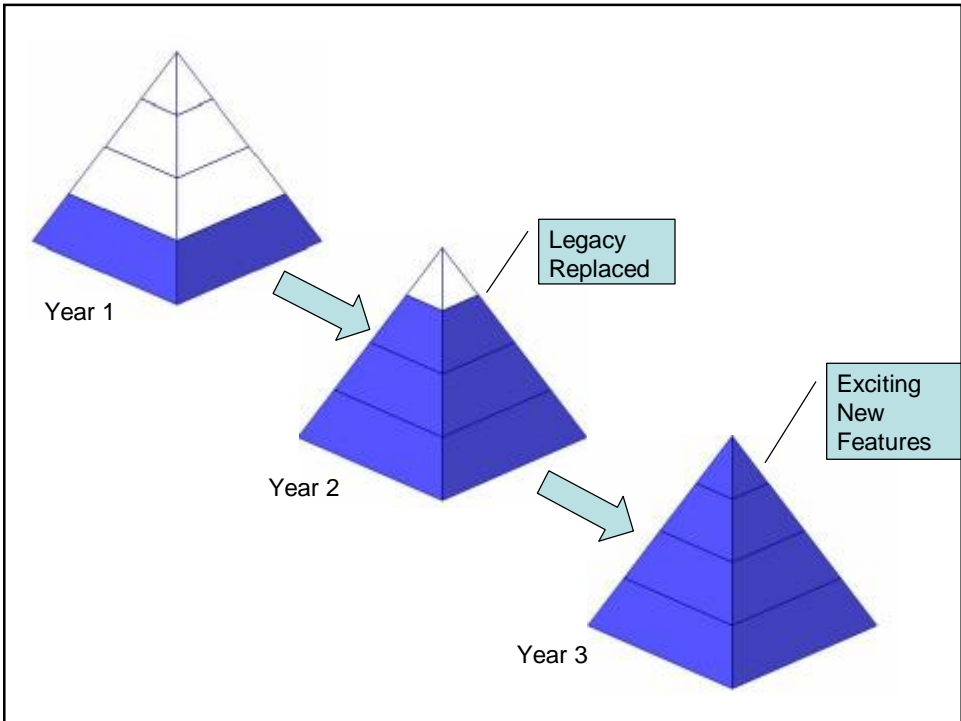
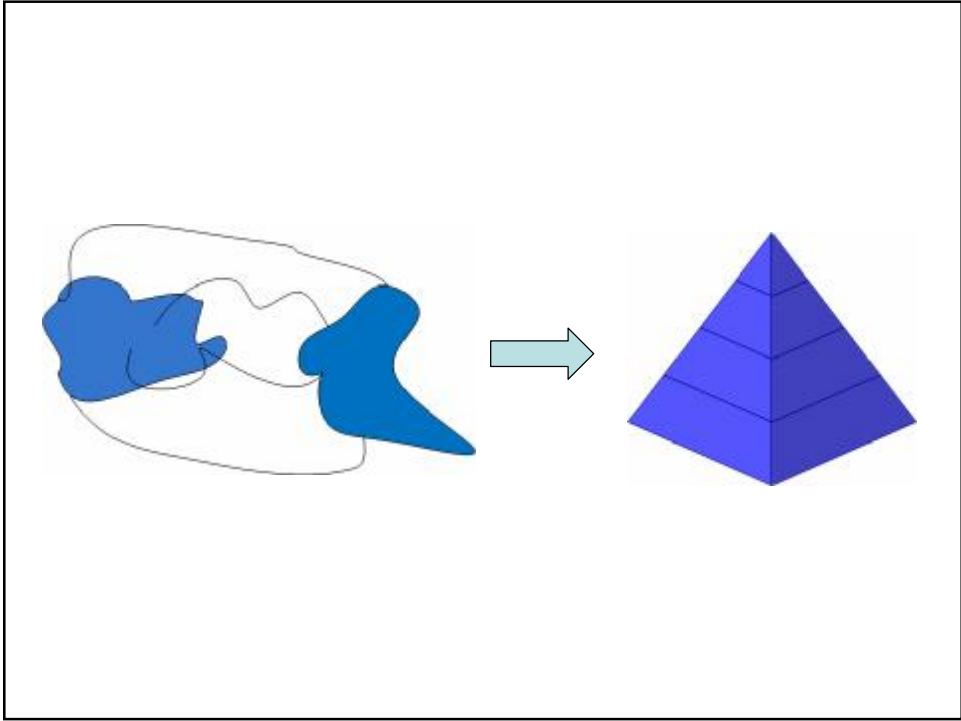
Eric Evans

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



Not all of a large system will be  
well designed.



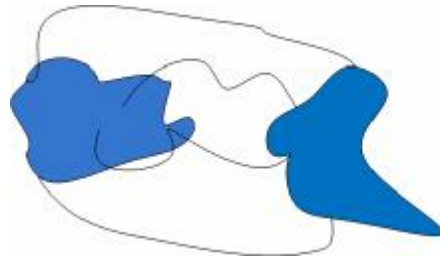


What's really going to happen?

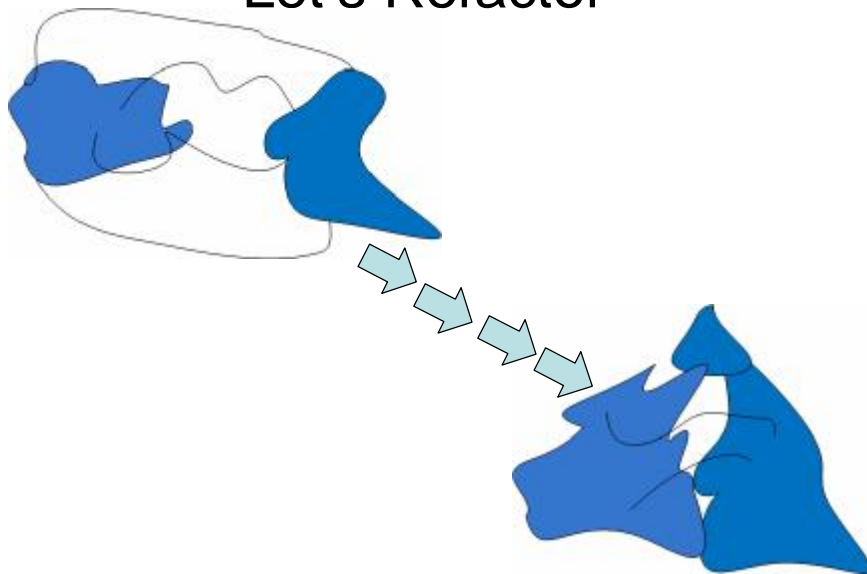
## Bad Strategic Goal: Shut Down Legacy

- Mundane functionality
- Perverse designs
- Steep the team in the old concepts
  
- The mad rush

Let's Refactor

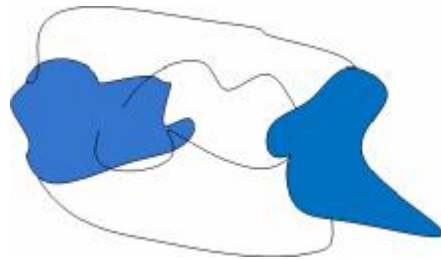


Let's Refactor

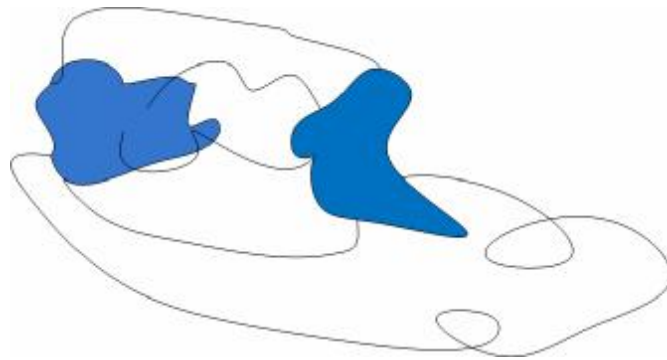


What's really going to happen?

Let's Hack



Let's Hack



Yes, that *is* really going to happen.

## More Bad Strategies

Building a platform to make other (lesser) programmers more productive.

→ *Design leverage turns out to be illusion.*

→ *You never deliver the sexy new capability.*

Cleaning up other people's mess.

→ *Sloppy hackers look even better.*

→ *Obscures feedback to management.*

What's a responsible designer  
to do?!!



# Strategic Design

- Distilling the Core Domain
- Context Mapping

**domain** A sphere of knowledge, influence, or activity.

The subject area to which the user applies a program is the *domain* of the software.

# Distilling the Core Domain



Generic Subdomains

Supporting Subdomains

Core Domain

amazon.com [Hello, sign in to get personalized recommendations. New customer? Start here.](#) [Shop Valentine's Gifts at 50% Off. You'll Love!](#)

Your Amazon.com [Today's Deals](#) [Gifts & Wish Lists](#) [Gift Cards](#) [Your Account](#) | [Help](#)

Shop All Departments Search Books [Go](#) [Cart](#) [Your Lists](#)

Desktop [Advanced Search](#) [Browse Subjects](#) [Not Now Recommended](#) [Best Sellers](#) [The New York Times Best Sellers](#) [Links in Español](#) [Bargain Books](#) [Textbooks](#)

Click to **LOOK INSIDE!**

**Domain-Driven Design: Tackling Complexity in the Heart of Software (Hardcover)**  
by Eric Evans (Author)  
[Read this for \(13 customer reviews\)](#)

List Price: ~~\$64.99~~  
Price: **\$51.99** & this item ships for **FREE** with Super Saver Shipping.  
[Details](#)  
You Save: \$13.00 (20%)

**In Stock.**  
Ships from and sold by Amazon.com. Gift-wrap available.

**Want it delivered Thursday, February 12?** Order it in the next 5 hours and 2 minutes, and choose **One-Day Shipping** at checkout. [Details](#)  
\$9 new from \$43.10 10 used from \$49.95

**Get Free Two-Day Shipping**  
Let free two-day shipping for three months with a special extended free trial of Amazon Prime™. Add this eligible textbook to your cart to qualify. Sign up at checkout. [See details.](#)

Quantity: 1  
[Add to Shopping Cart](#)  
or  
Subscribe to free one-time ordering  
or  
[Add to Cart with FREE Two-Day Shipping](#)  
Amazon Prime Free Trial required. Sign up when you check out. [Learn More](#)

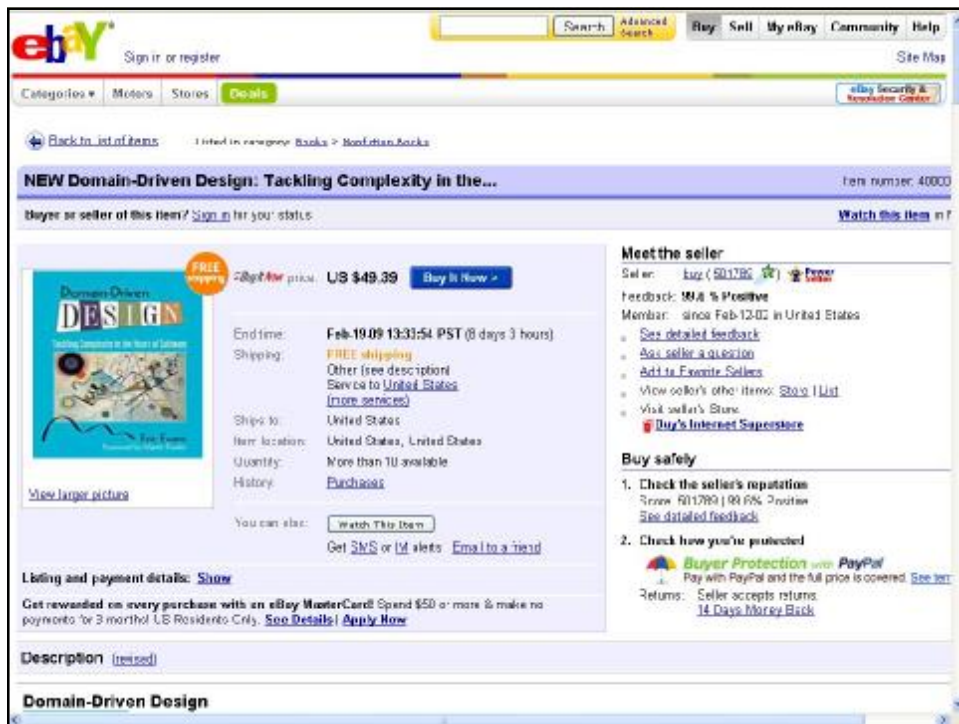
**More Buying Choices**  
**40 used & new from \$43.10**  
Have one to sell? [Sell your here](#)

[Add to Wish List](#)  
[Add to Shopping List](#)  
[Add to Wedding Registry](#)  
[Add to Baby Registry](#)  
 [Share with Friends](#)

**Frequently Bought Together**

[LOOK INSIDE!](#) [LOOK INSIDE!](#) [LOOK INSIDE!](#) **Price For All Three: \$149.47**  
[Add all three to Cart](#)

**This item:** Domain-Driven Design: Tackling Complexity in the Heart of Software by Eric Evans  
 **Patterns of Enterprise Application Architecture (Addison-Wesley Signature Series)** by Martin Fowler  
 **Refactoring: Improving the Design of Existing Code (Addison-Wesley Object Technology Series)** by Martin Fowler



# Distilling the Core Domain

– What makes your system worth writing?

## Distilling the Core Domain

- What makes your system worth writing?
- Why not buy it off the shelf?

## Distilling the Core Domain

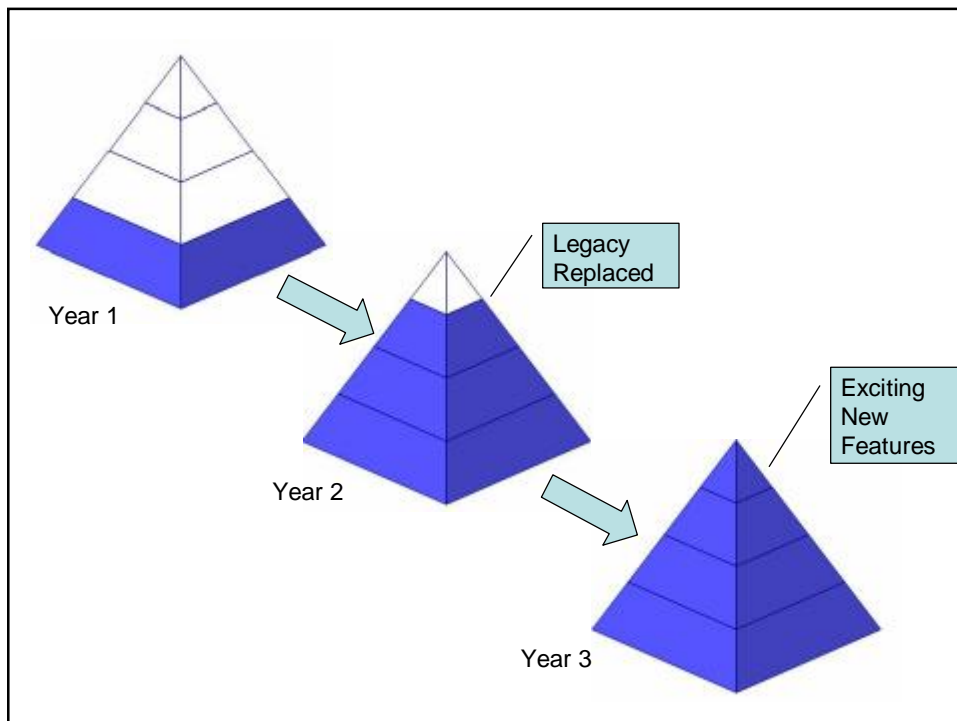
- What makes your system worth writing?
- Why not buy it off the shelf?
- Why not outsource it?

The heroes are working in the  
core domain.

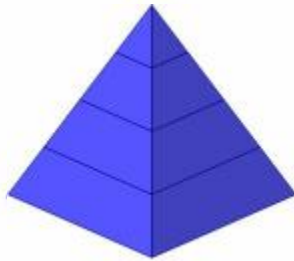
The heroes are working in the  
core domain.

This is where your *best*  
people should be!

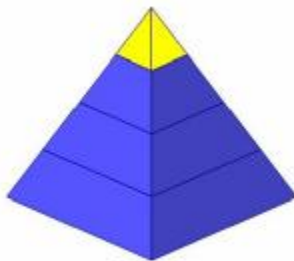
# Not Just Core Features; *Core Domain*



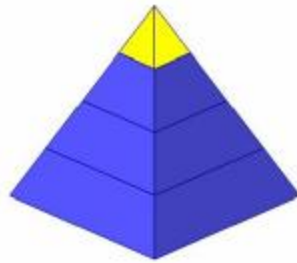
Which Parts are Core?



Which Parts are Core?



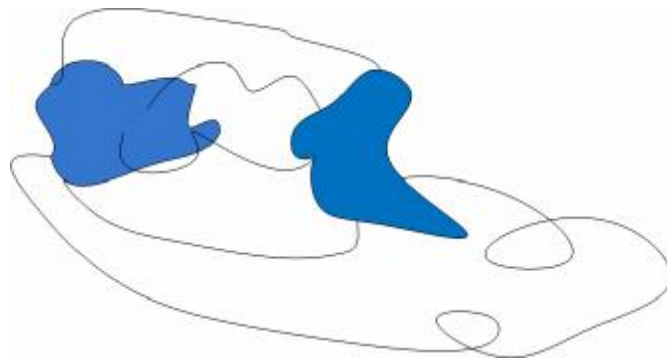
Which Parts are Core?



or

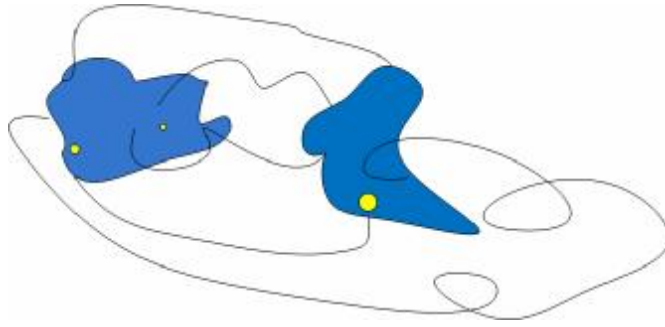


Which Parts are Core?

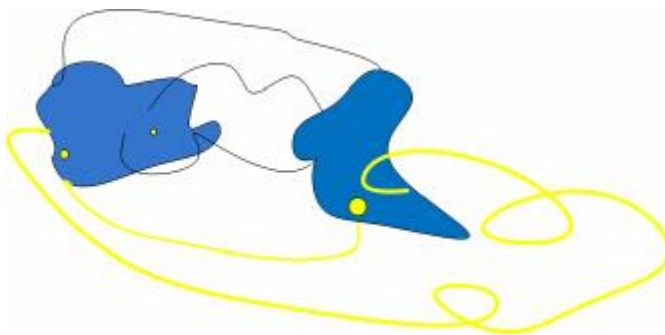




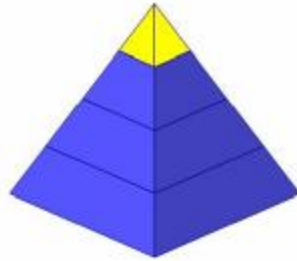
Which Parts are Core?



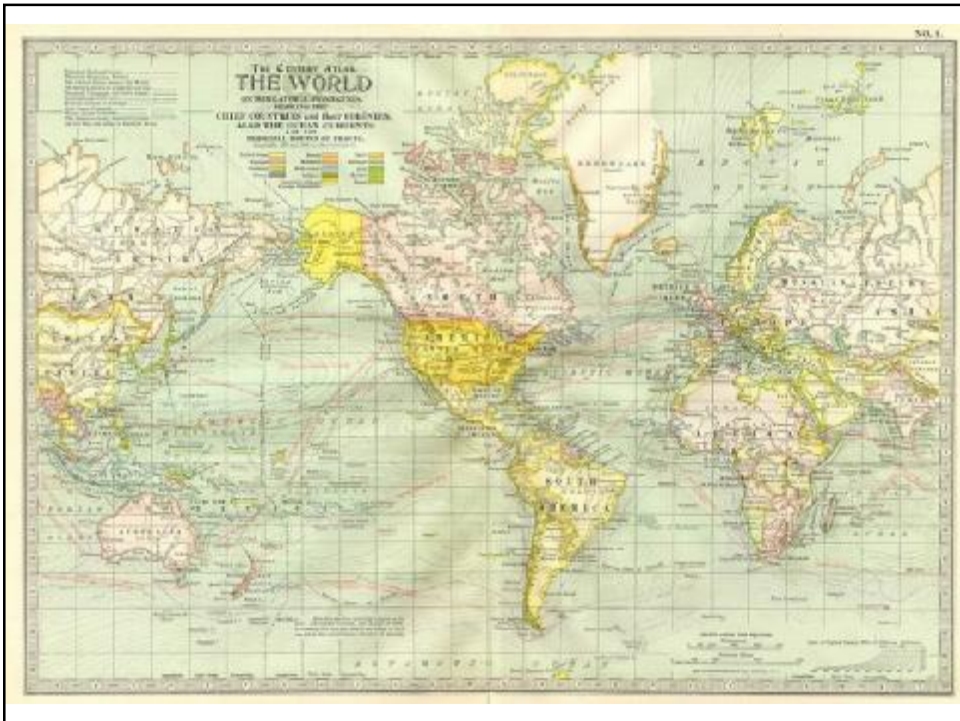
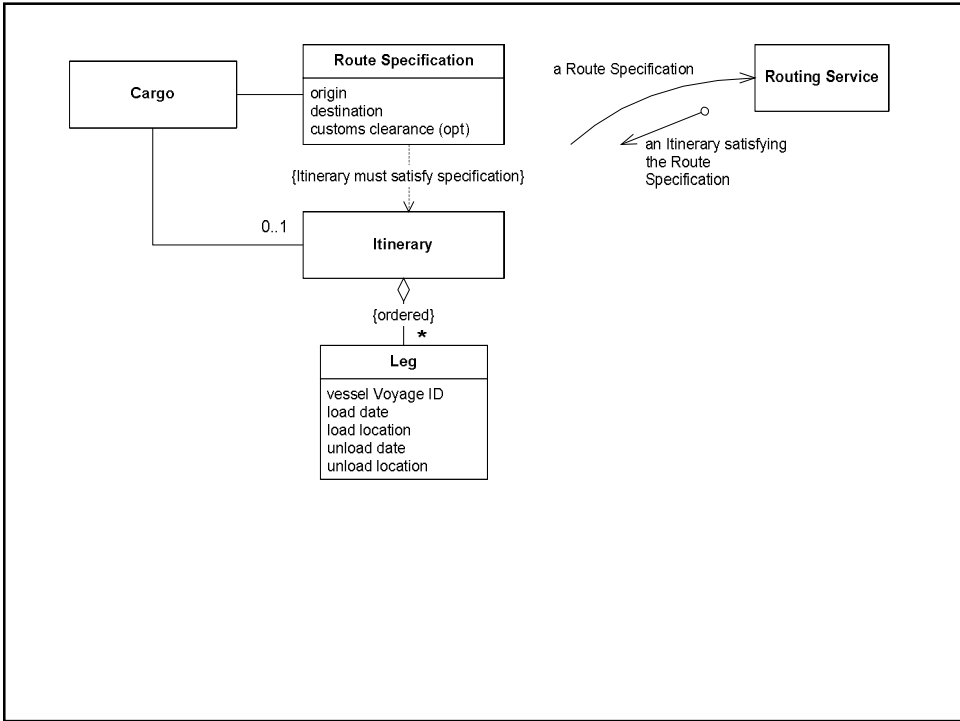
Which Parts are Core?



How could we just build  
the core part?

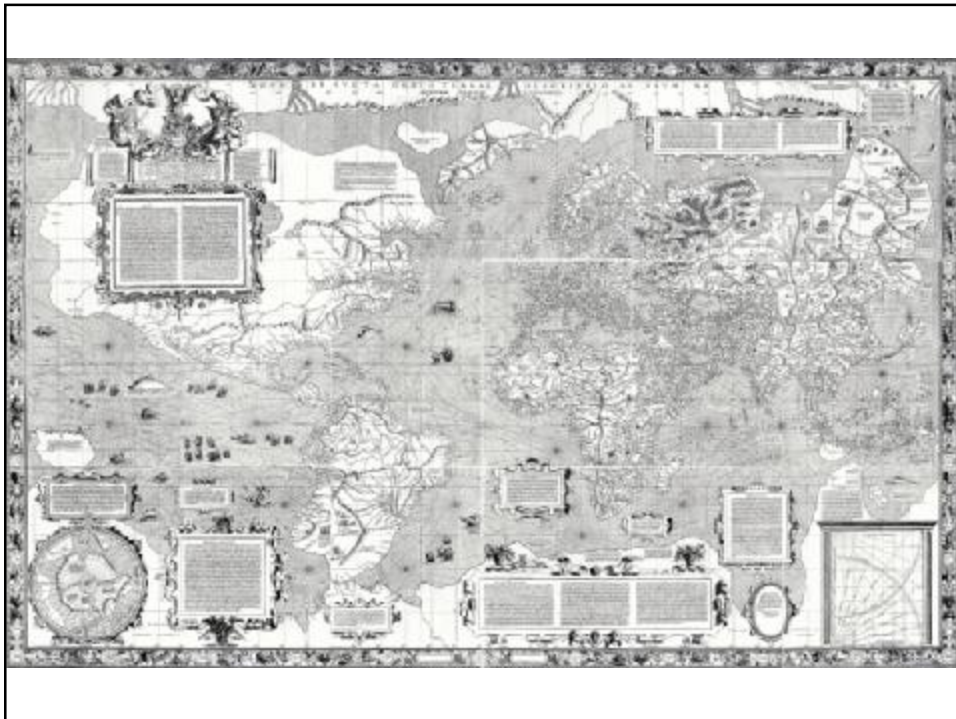


What is a model?



## Some statements about this model

- The Earth is approximated as a sphere.
- There are functions that map points on the surface of a sphere onto a cylinder (projections).
- Mercator's Projection preserves cardinal direction of points relative to each other.



**domain** A sphere of knowledge or activity.

**model** A system of abstractions representing selected aspects of the domain.

*A model is a distilled form of domain knowledge, assumptions, rules and choices.*

## It's not about "realism"

A model serves some *use*.

Usefulness is specific to *particular scenarios*.

There are always *multiple models* of a domain.

There are always multiple models.

It was six men of Indostan  
To learning much inclined,  
Who went to see the Elephant  
(Though all of them were blind),  
That each by observation  
Might satisfy his mind.

The *First* approached the Elephant,  
And happening to fall  
Against his broad and sturdy side,  
At once began to bawl:  
"God bless me! but the Elephant  
Is very like a wall!"

...

The *Third* approached the animal,  
And happening to take  
The squirming trunk within his hands,  
Thus boldly up and spake:  
"I see," quoth he, "the Elephant  
Is very like a snake."

The *Fourth* reached out his eager hand,  
And felt about the knee.  
"What most this wondrous beast is like  
Is mighty plain," quoth he;  
"'Tis clear enough the Elephant  
Is very like a tree!"

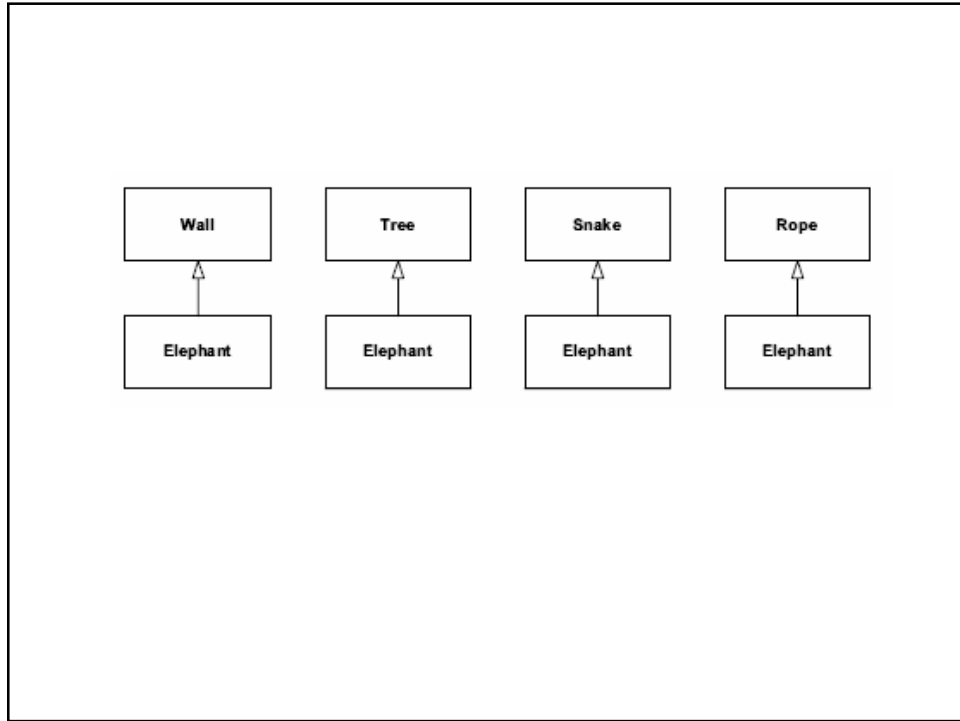
...

The *Sixth* no sooner had begun  
About the beast to grope,  
Than, seizing on the swinging tail  
That fell within his scope,  
"I see," quoth he, "the Elephant  
Is very like a rope!"

And so these men of Indostan  
Disputed loud and long,  
Each in his own opinion  
Exceeding stiff and strong,  
Though each was partly in the right,  
And all were in the wrong!

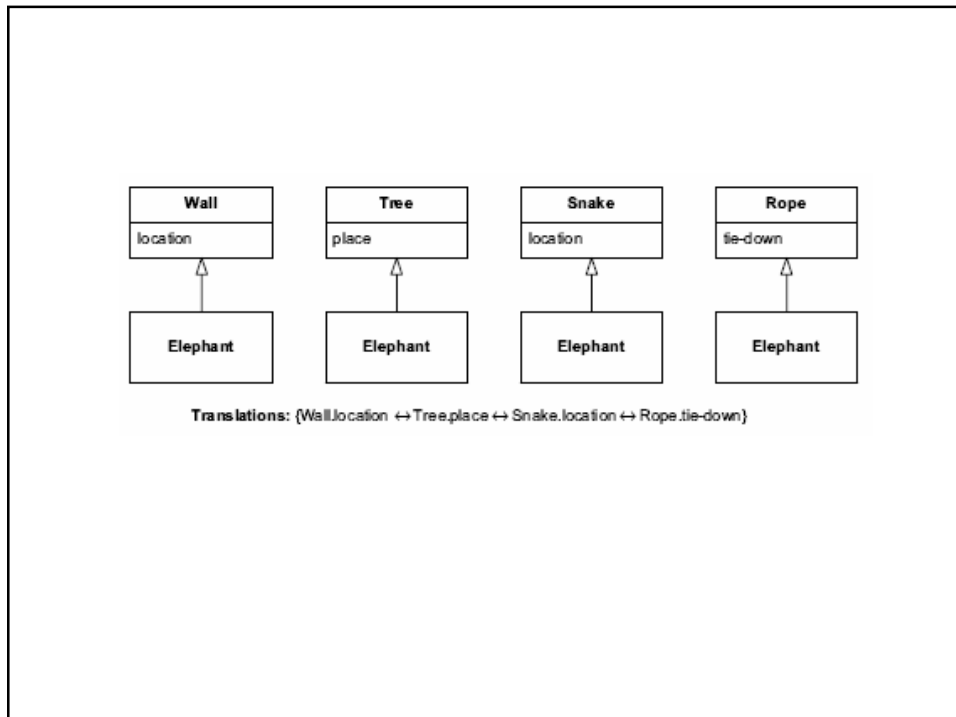
...

—From "The Blind Men and the Elephant," by John Godfrey Saxe  
(1816-1887), based on a story in the *Udana*, a Hindu text

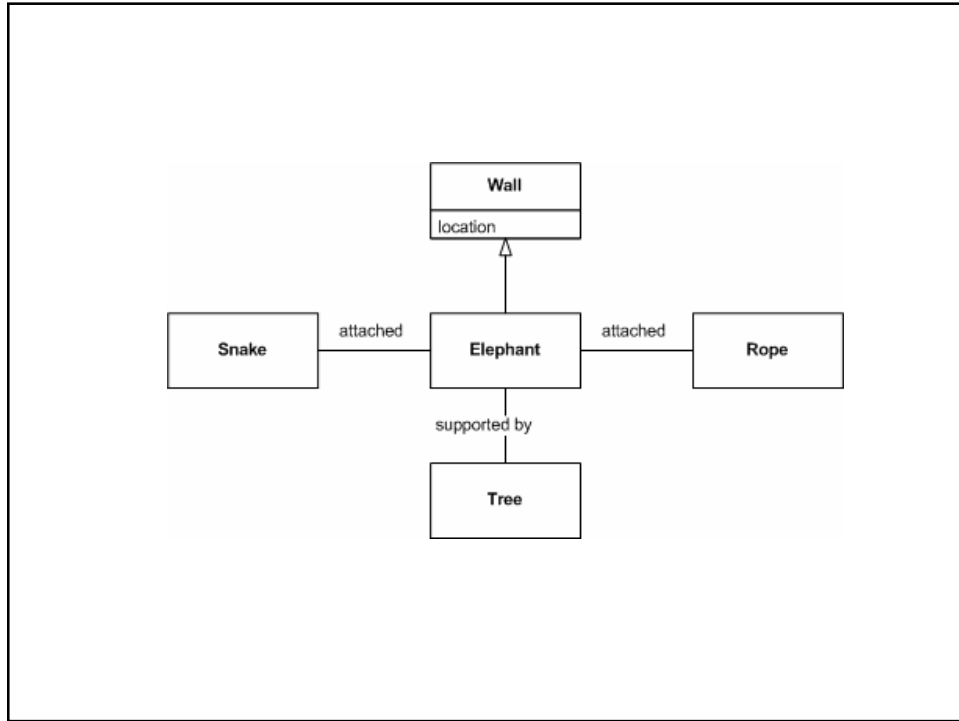


**context** The setting in which a word or statement appears that determines its meaning.

**Bounded Context** An operational definition of where a particular model is well-defined and applicable. (Typically a subsystem, or the work owned by a particular team).



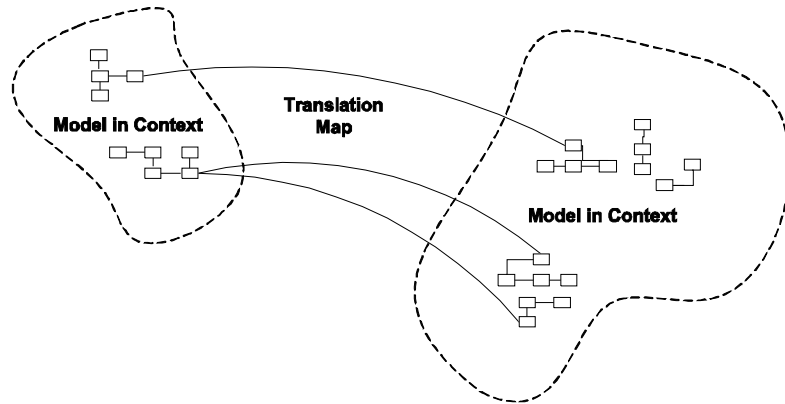




## A Bad Strategy: The Enterprise Model

*One Ring to rule them all, One  
Ring to find them, One Ring to  
bring them all, and in the darkness  
bind them*

# “Context Map”

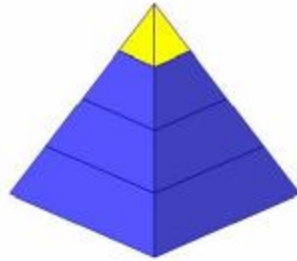


Precision designs are fragile

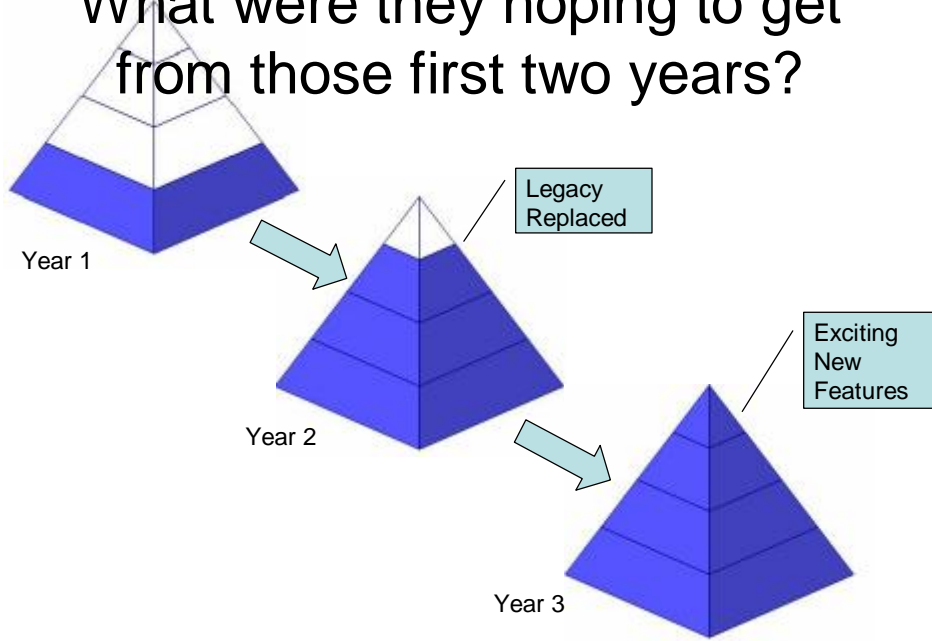
Sophisticated Design &  
Modelling Require:

A Bounded Context containing a  
Unified Model

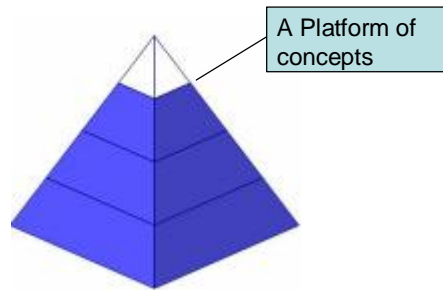
How could we just build the core part?



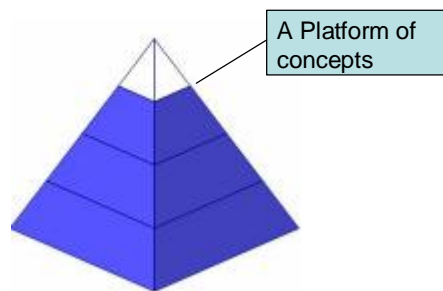
What were they hoping to get from those first two years?



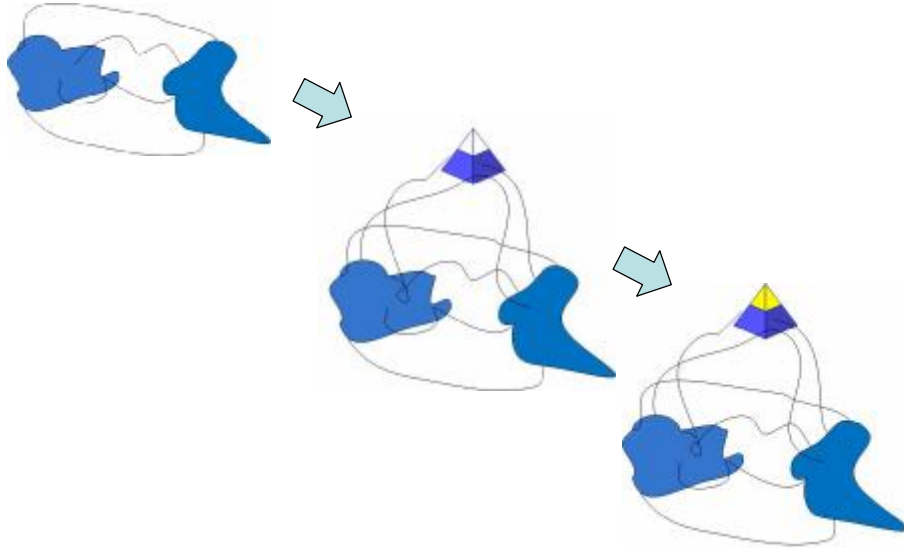
What were they hoping to get from those first two years?



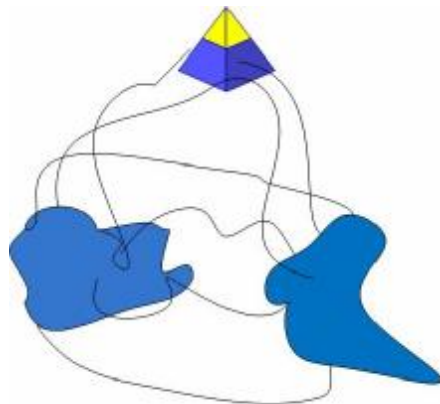
We only *really* care about the platform *interface*.

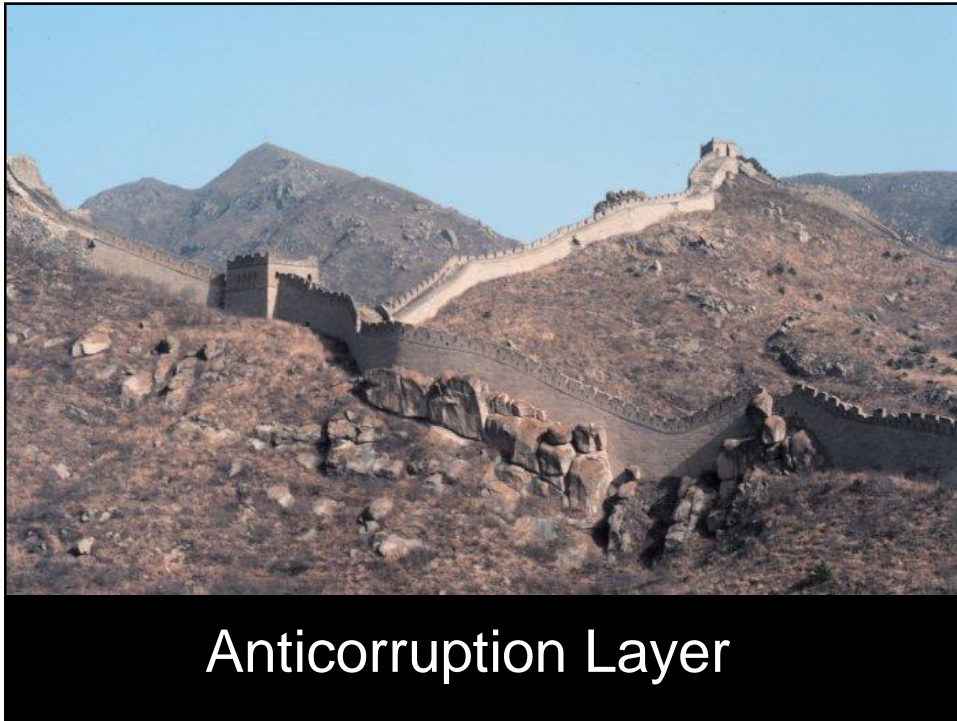


## Build Platform Based on Model



## Build Platform Based on Model





dddcommunity.org/experiencereports

The screenshot shows the website interface for dddcommunity.org. The main content area features an article titled "Anticorruption - A Domain-Driven Design Approach To More Robust Integration" by Sara Healy and Ming He. The article includes a diagram illustrating the "Anticorruption" layer, which is highlighted with a red box. The diagram shows a central box labeled "Anticorruption" connected to various external systems and data sources. The text of the article discusses how this layer helps in integrating with legacy systems and managing data consistency. The website also includes a navigation menu, a search bar, and a user login section.

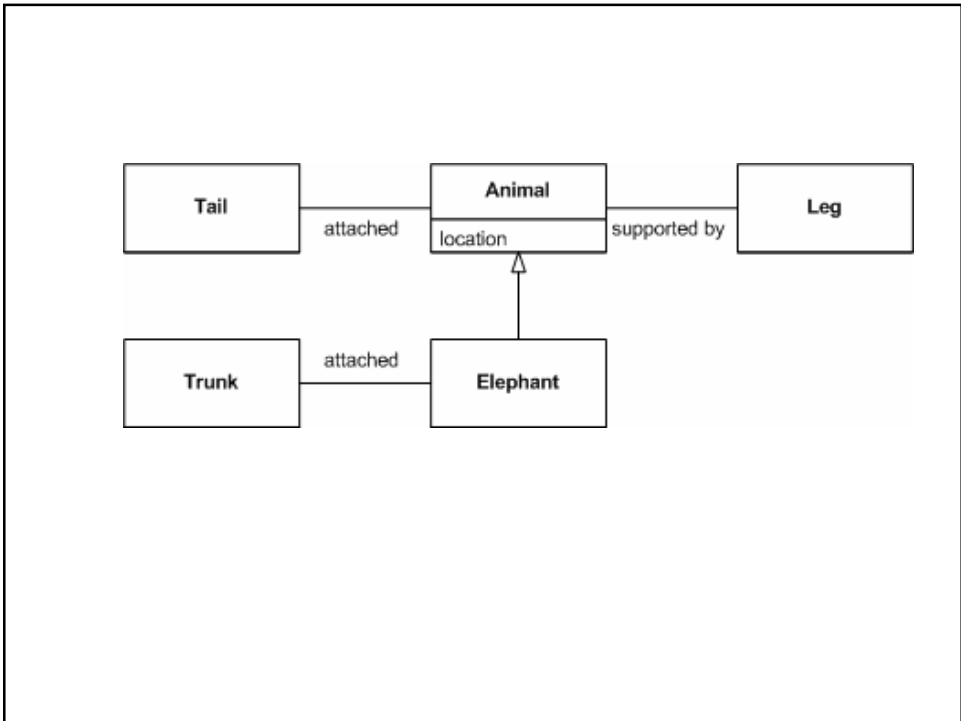
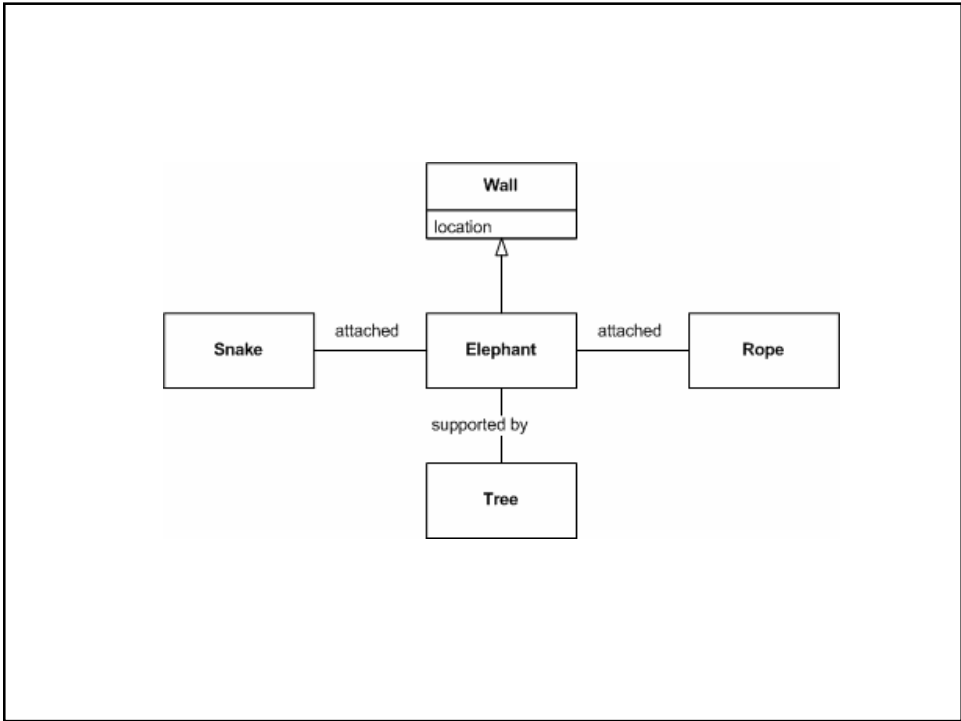
## Good Strategic Goals

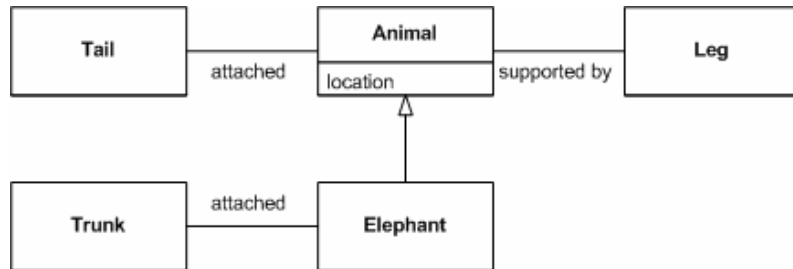
- Clarify context map
- Stabilize legacy systems
- Focus on core domain
- Core domain development in clean context
- Create a platform
- Deliver early
- Produce enthusiastic business sponsors

## Ugly (but Acceptable) Consequences

- Thick translation layers
- Inelegant (but stable) legacy
- Inelegant (but stable) supporting domains







*Should we do this?*

Look for assets in the legacy.

Look for assets in the legacy.

Work in the core domain.

Look for assets in the legacy.

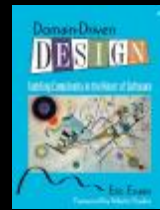
Work in the core domain.

**Be the hero for a  
change!**

# Strategic DDD Resources

[www.dddcommunity.org](http://www.dddcommunity.org)

*Domain-Driven Design*  
by Eric Evans  
*PART 4!*



[www.domainlanguage.com](http://www.domainlanguage.com)