What drives your design?

Stephan Hochdörfer, bitExpert AG

About me

- Stephan Hochdörfer, bitExpert AG
- Department Manager Research Labs
- enjoying PHP (and a little bit of Java) since 1999
- S.Hochdoerfer@bitExpert.de
- @shochdoerfer

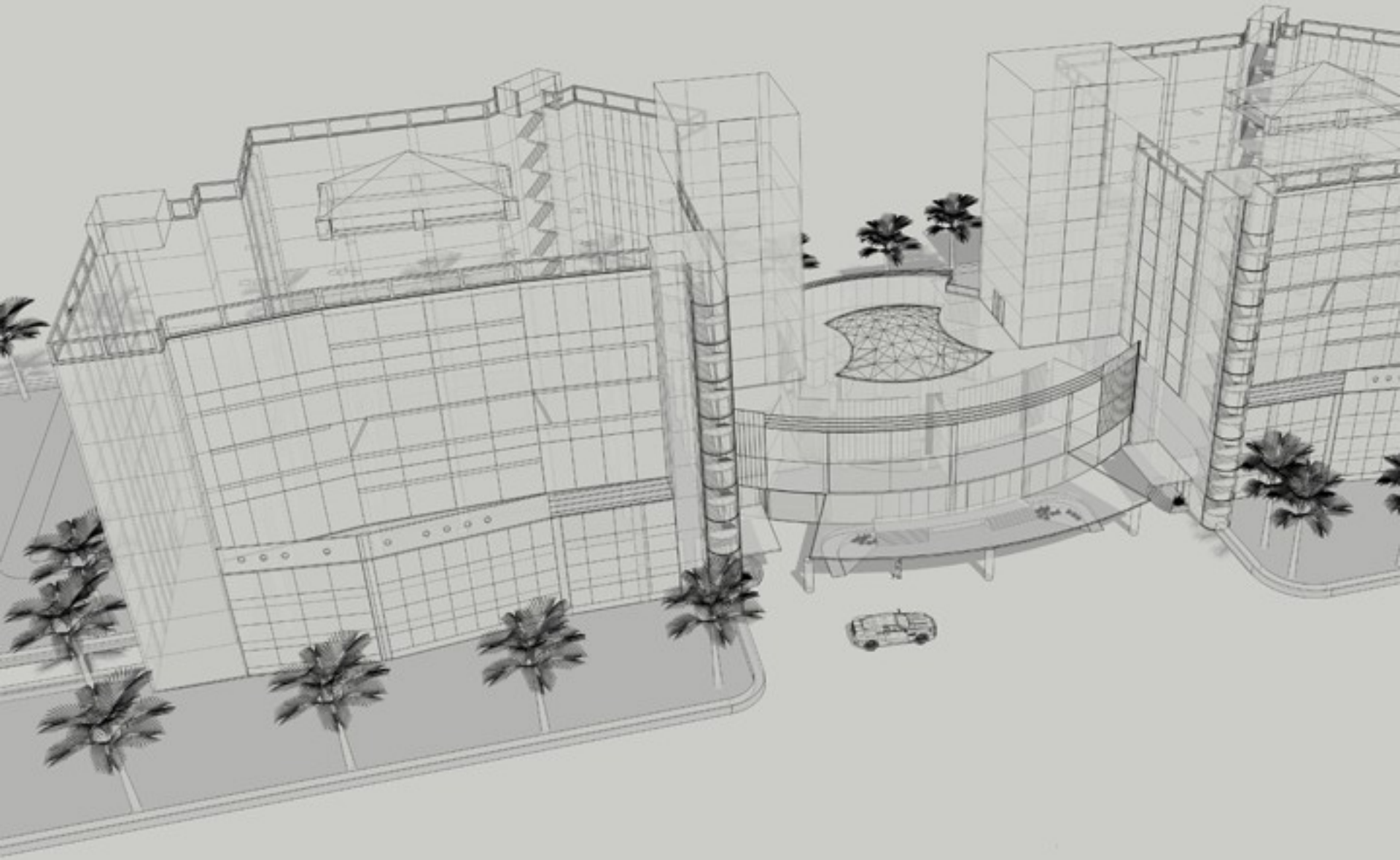
Remark: Sorry no code!

Code is not important, the domain is!

Domain Driven Design

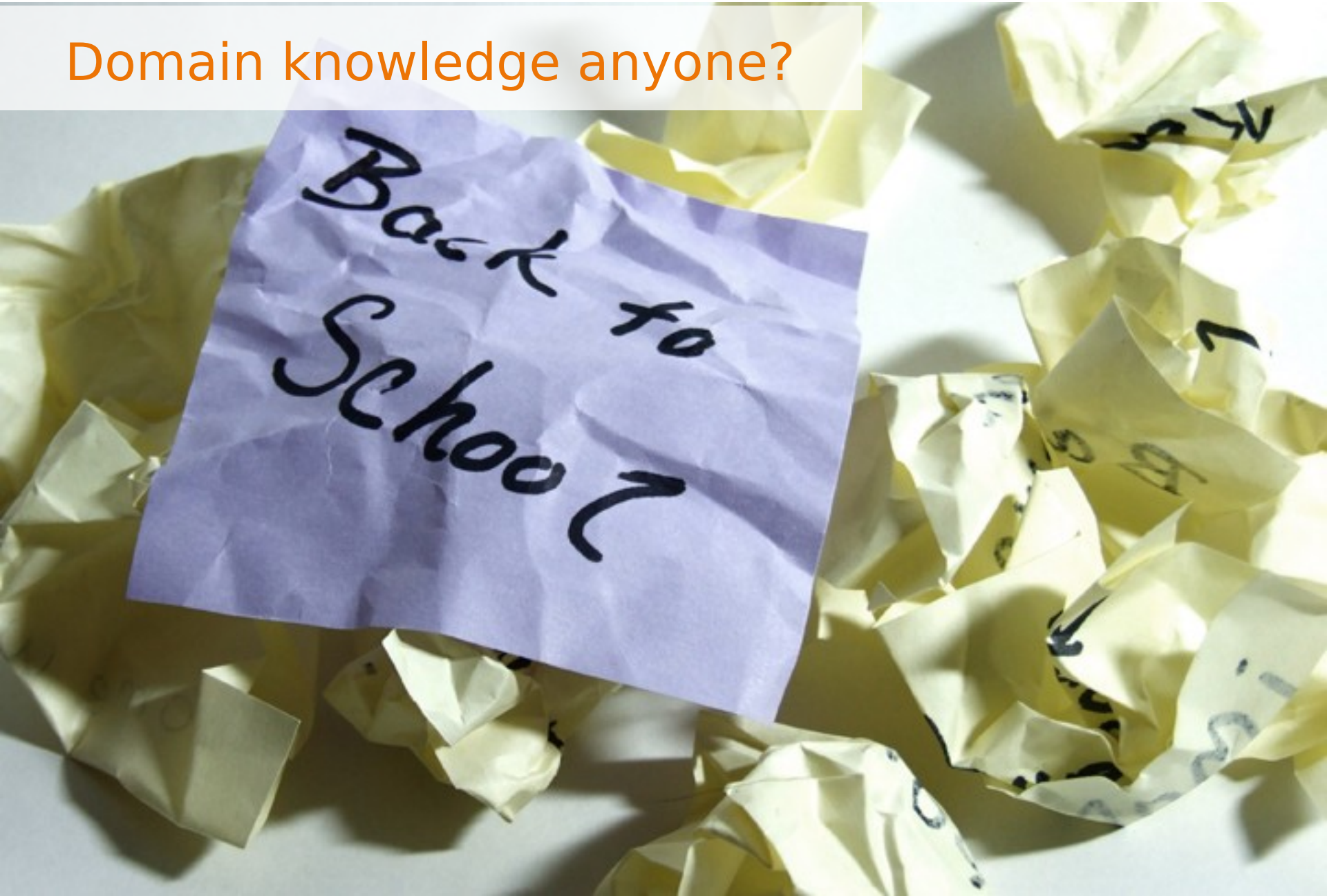
What`s a domain?

What`s a domain?



What drives your design?

Domain knowledge anyone?



What drives your design?

Will die without care!

How to build domain knowledge?

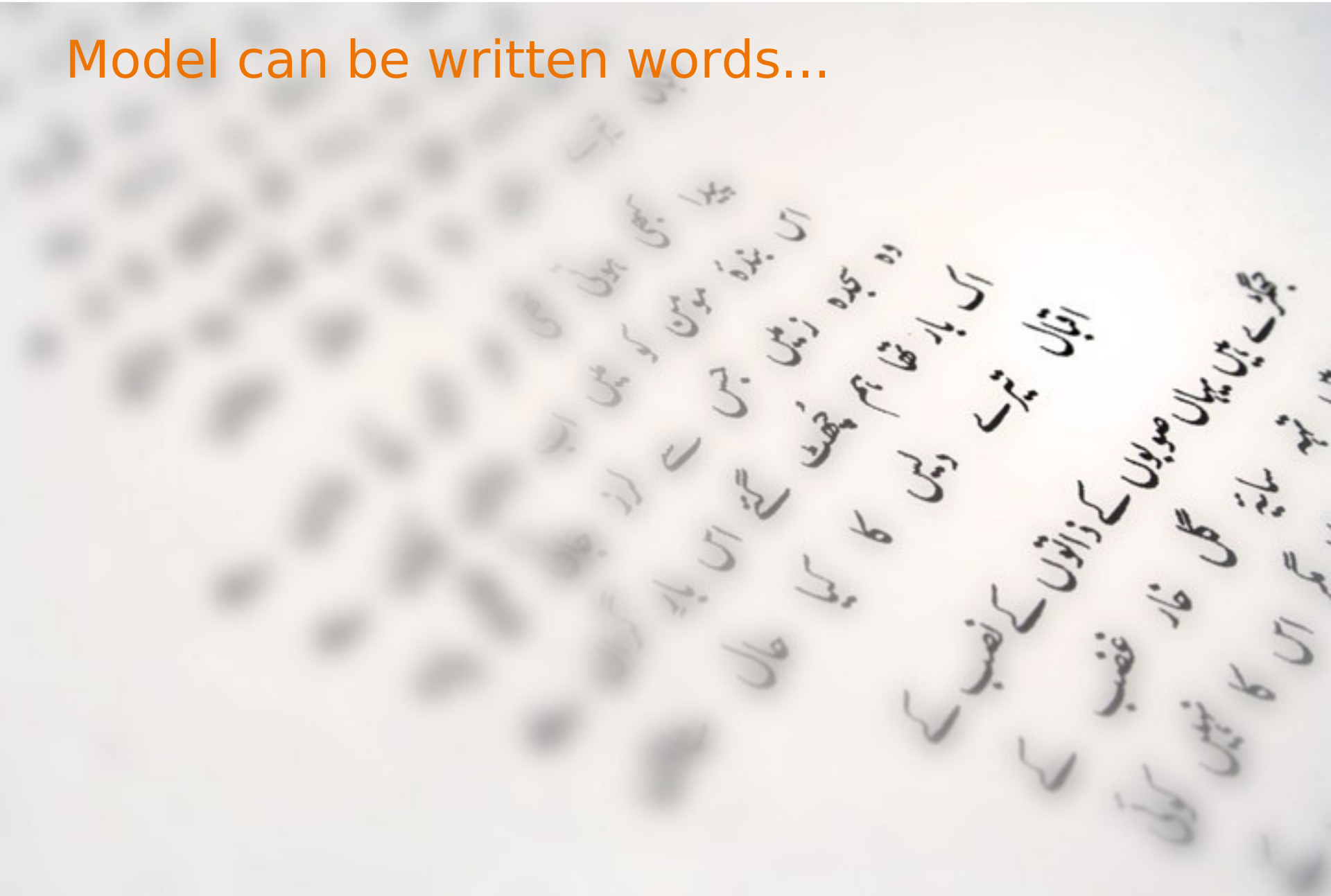
What drives your design?

Talk to the experts!

How to express
domain knowledge?

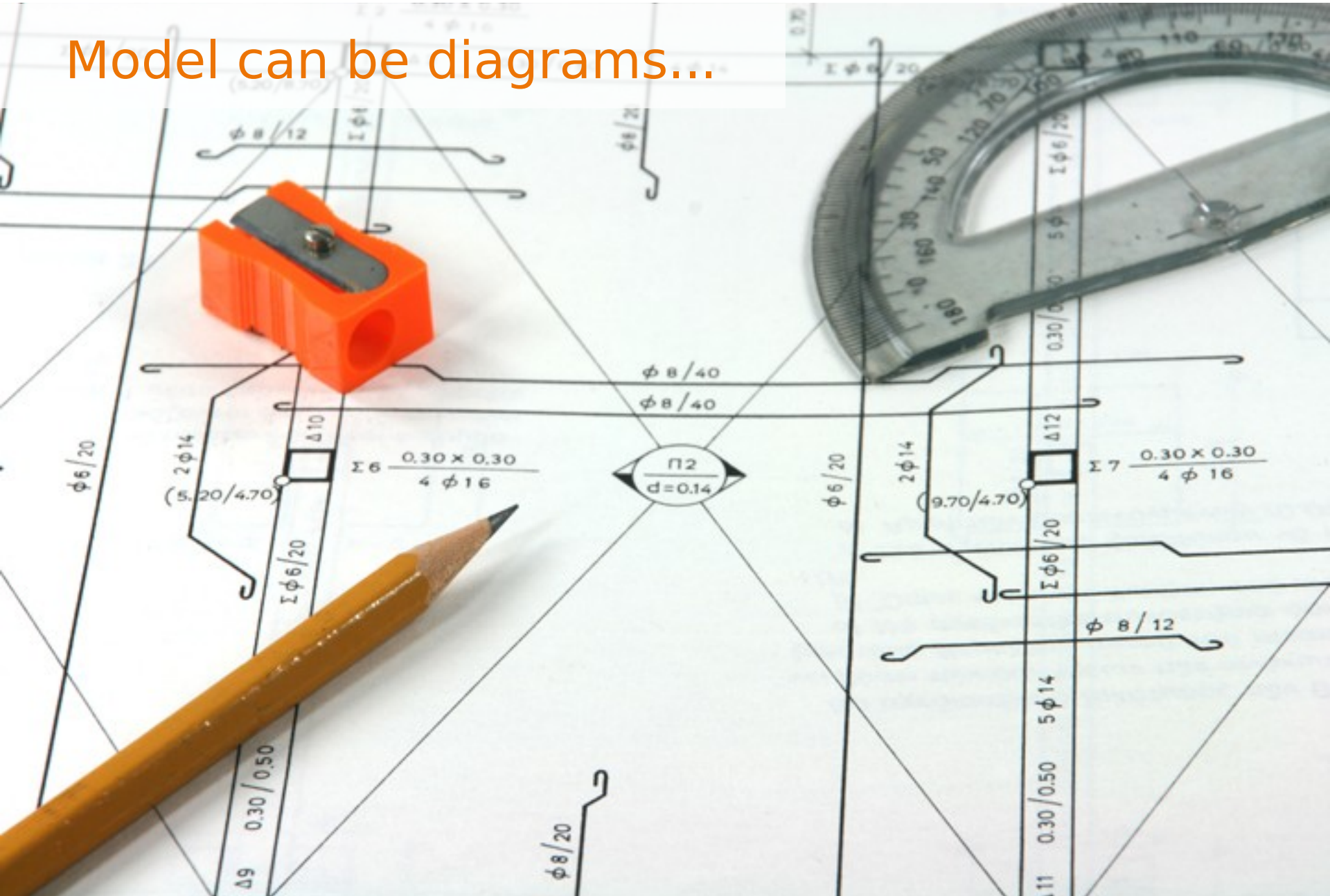
Models please!

Model can be written words...

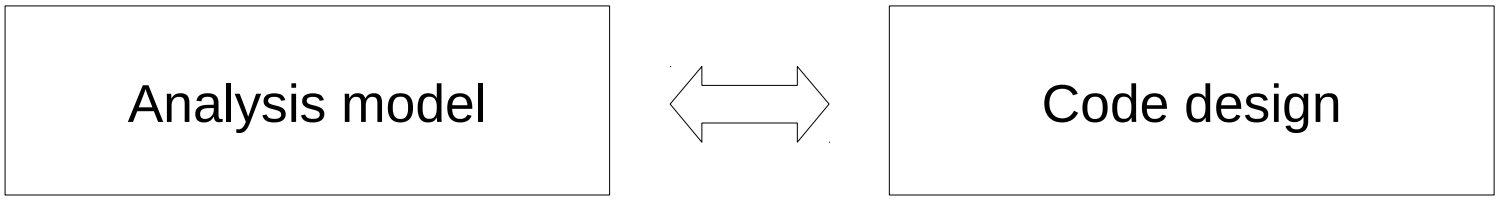


What drives your design?

Model can be diagrams...



Many models?



How to describe a model?

Ubiquitous Language

"Britain and America are two nations divided by a common language."

George Bernard Shaw / Oscar Wilde

Ubiquitous Language

"Business and Developers are two nations divided by a common language."

Casey Charlton

Ubiquitous Language

A language specific
to a certain domain.

Ubiquitous Language

Who is the „Customer“?

Ubiquitous Language

Please, use it in your code as well!

Domain specific language

```
if(1 == node.getLeft()) {
  // handle root node seperatly

  // [...]
}
else {
  // handle all other nodes

  // [...]
}
```

Domain specific language

```
if(node.isRoot()) {
  // [...]
}
else {
  // [...]
}
```

Domain specific language

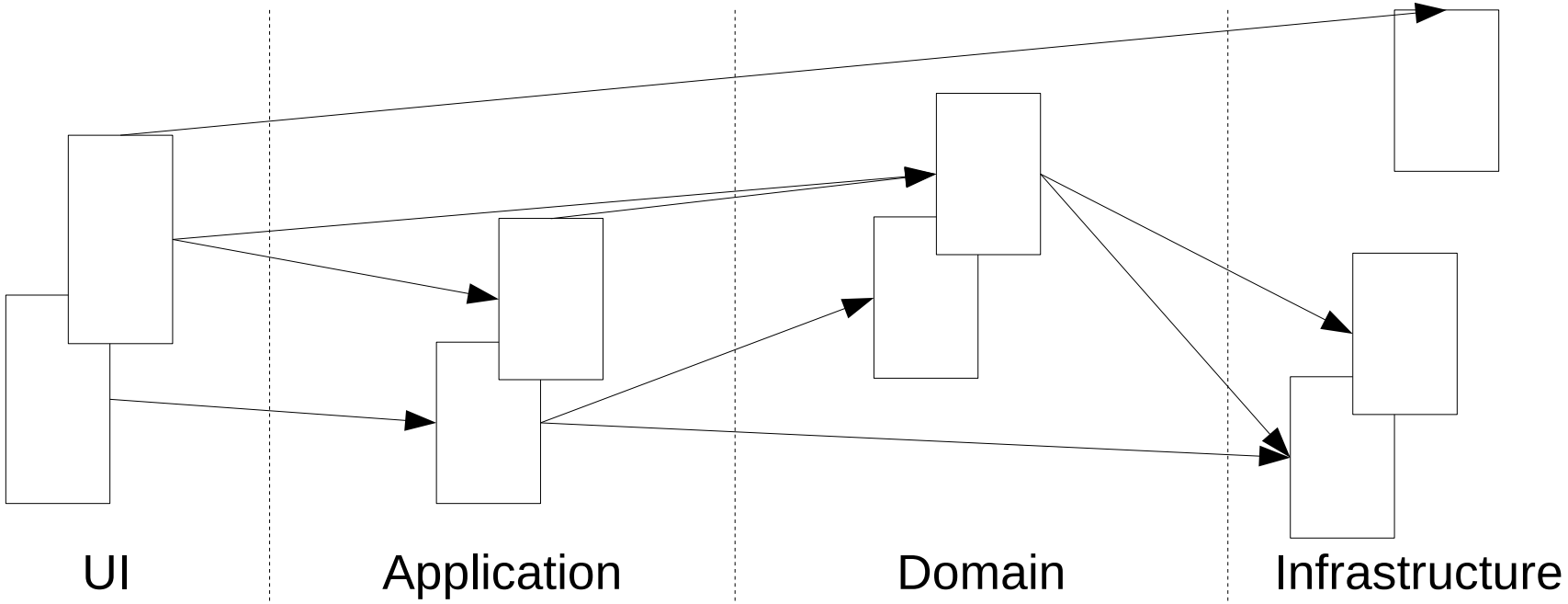
```
if(category.isStart()) {
  // [...]
}
else {
  // [...]
}
```

Domain specific language

Bonus point: Discuss
domain code with your client!

Layered architecture

Layered architecture



Layered architecture

Tightly coupled code will lead to unmaintainable code.

Entities

- Objects with an identity
- Spanning the life of the system
- The model must define what it means to the same thing
- e.g Person object

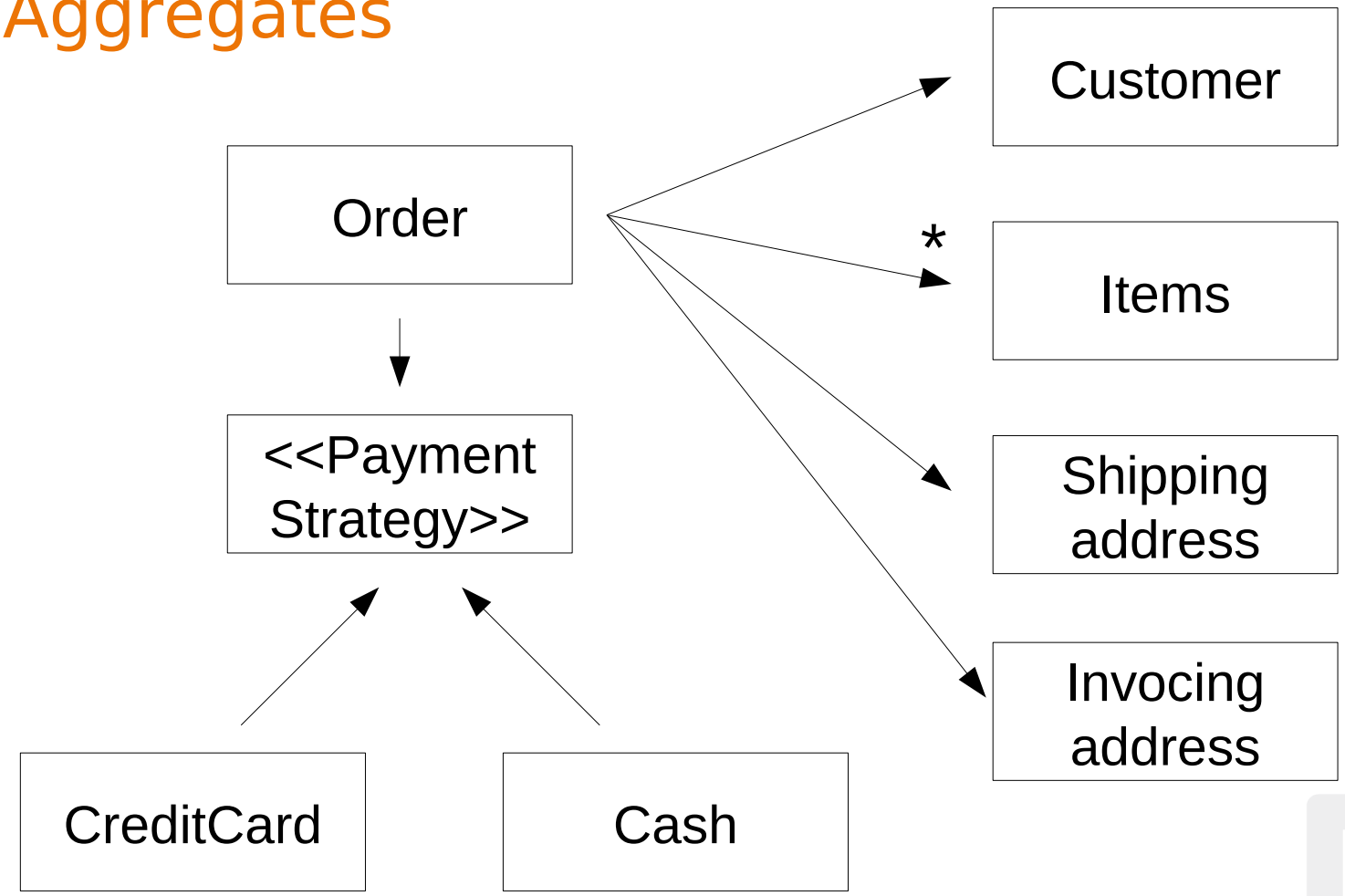
Value Objects

- Objects without an identity
- Can be easily created or discarded
- We do not care what the object is, but what it has
- References other Value Objects or Entities

Aggregates

Too many objects?

Aggregates


Order
<<Payment Strategy>>
CreditCard
Cash
Customer
*
Items
Shipping address
Invocing address

Service objects

- Services are verbs of the ubiquitious language.
- Provides functionality for the domain
- A service is not an store, does not provide persistency!
- Performs operations

Growing too big?

Growing too big?

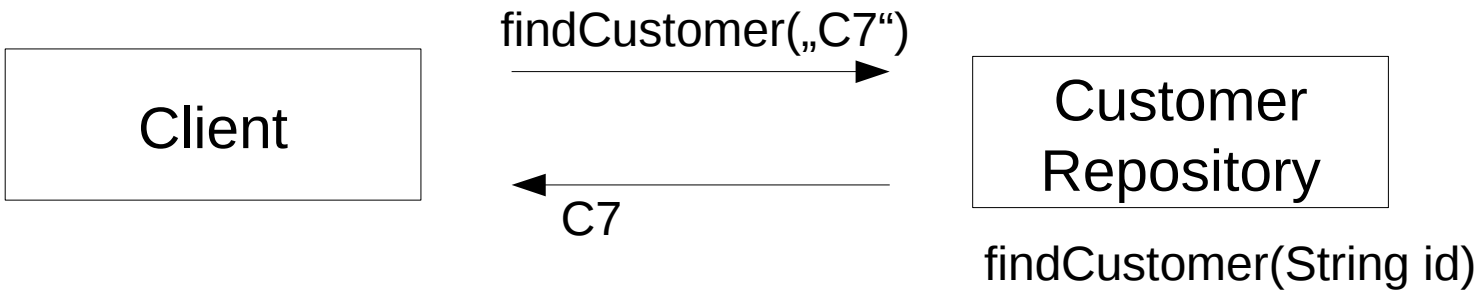
Modularize!

Where to retrieve the data from?

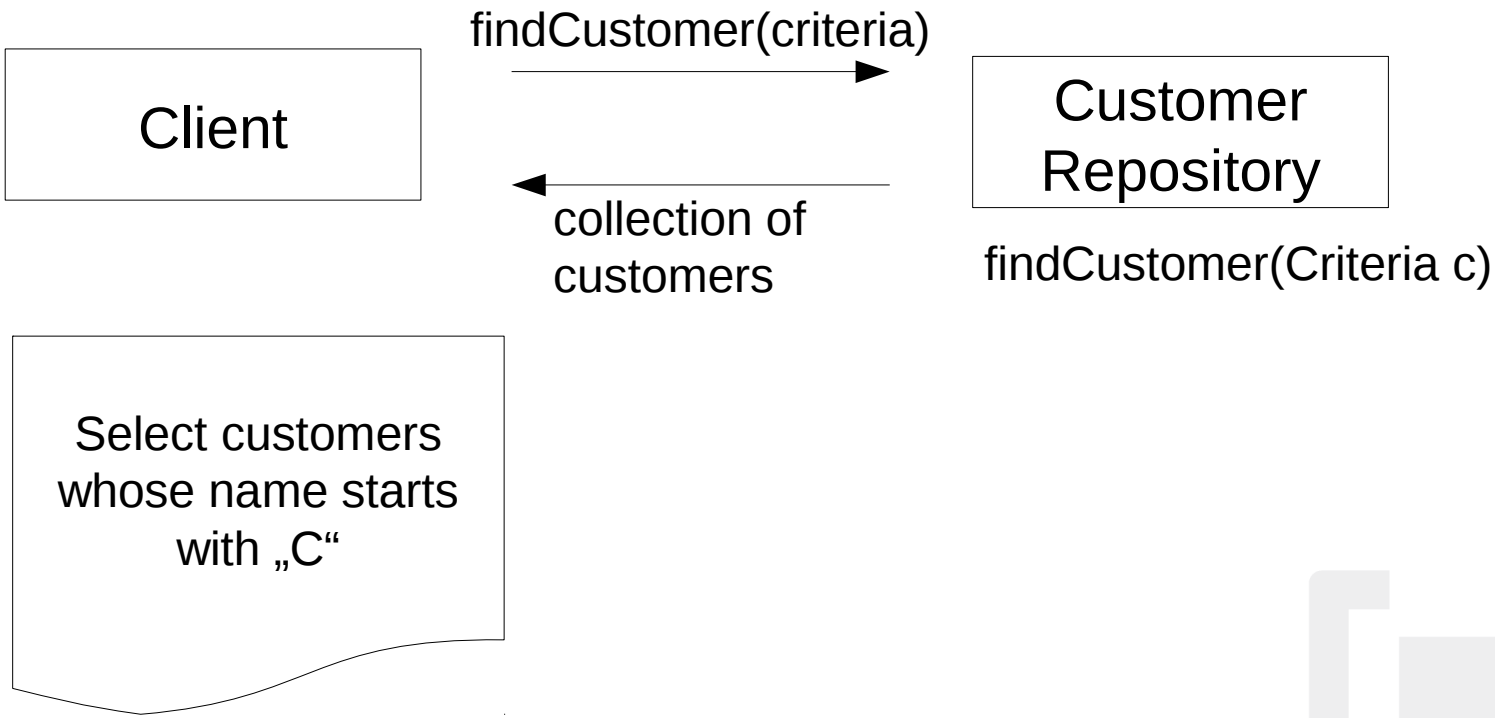
Repositories

- No coupling to persitence infrastructure in domain code!
- Encapsulates the logic to retrieve object references
- Communicates with the infrastructure
- Does not care where the entries are stored
- One central place where the data communication happens

Repositories



Repositories - Specifications



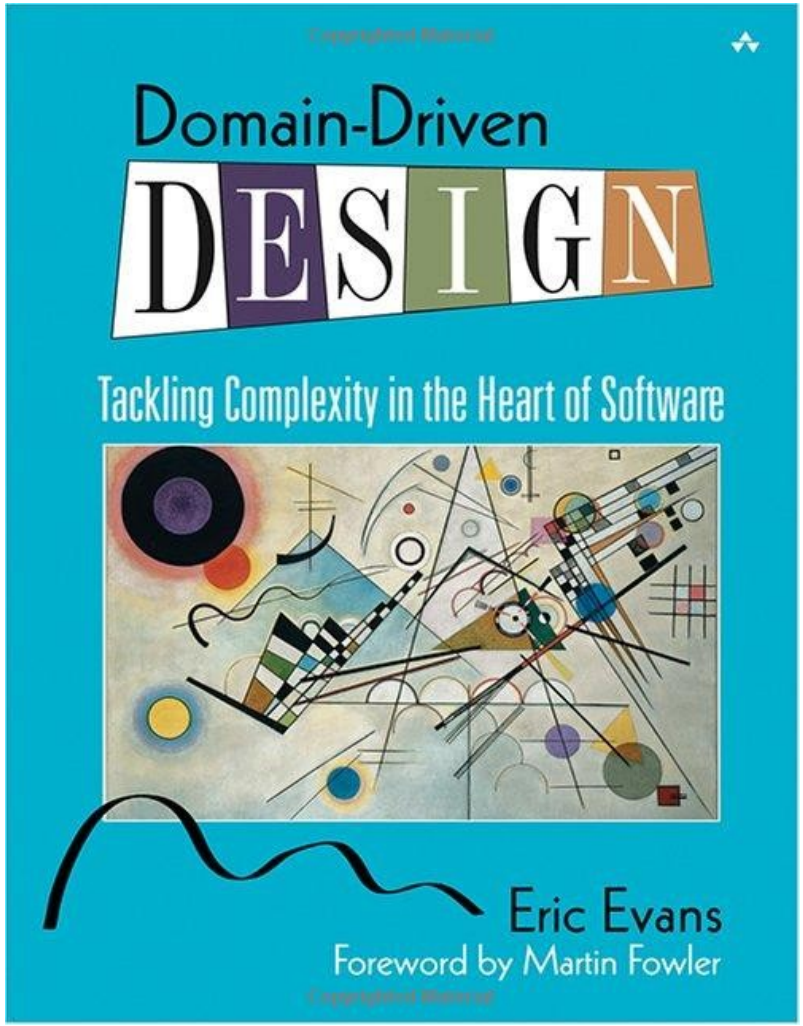
Repositories - Specifications

```
var spec = newPizzaSpecification()
   .BasedOn(newMargaritaPizzaSpecification())
   .WithThickCrust()
   .WithExtraCheese();

var pizza = new PizzaFactory().
    CreatePizzaFrom(spec);
```

Repositories – Specifications for Queries

```
var margaritaPizzaCustomers =
newMargaritaPizzaSpecification();

var customers = this.customerRepository
  .GetCustomersSatisfying(margaritaPizzaCustomers);
```

What drives your design?



bitExpert

Thank you!

http://joind.in/3184