

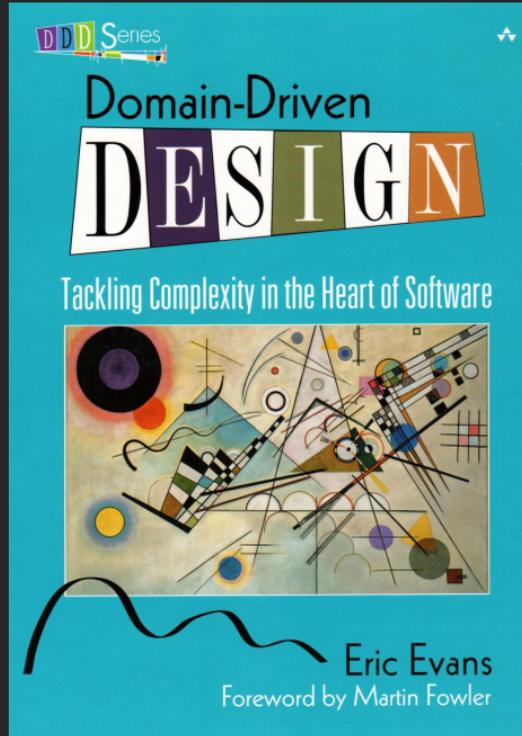
An introduction to Domain Driven Design



Rob Allen, January 2024



It started with a book



Tackling Complexity in the Heart of Software



DDD provides for the
Strategic and Tactical



Domains & Models



24 BRANCHES



CHANCERY LANE STATION

A Domain is

something in the real-world



A Domain is

the subject of our project



A Domain is

understood by experts in the space



A Domain Model is

a representation of the Domain



A Domain Model is

an abstraction of reality



A Domain Model is

expressed as diagrams, words, code



Building domain knowledge

*"We have really everything in common
with America nowadays except, of
course, language"*

Oscar Wilde



Ubiquitous Language



Ubiquitous Language is

*The agreed concepts, meanings and
terms for the project*



You have to talk!



Ubiquitous Language is

*Foundational to implementing
Domain Driven Design*



In Domain-Driven Design

*Everything revolves around
Ubiquitous Language*



Observations about creating a domain model



Focus effort
where it matters



Knowledge Crunching



Knowledge Crunching

Event Storming



Knowledge Crunching

User Story Mapping





Knowledge Crunching

Event Modelling



```
Todo           Done
1) Buy Milk    1) Wash Car
2) Buy Bread

(C)reate (F)inish (Q)uit:
C
Vaccuum House
```

```
Todo           Done
1) Buy Milk    1) Wash Car
2) Buy Bread
3) Vaccuum House

(C)reate (F)inish (Q)uit:
```

List Items

name	status
Buy Bread	todo
Buy Milk	todo
Wash car	done
vaccuum	todo

Add a ToDo Item
- name: vaccuum
house

Finish a ToDo It
- name: vaccuum
house

Added a ToDo Item
- name: vaccuum
house

Finished
Item
- vaccuum

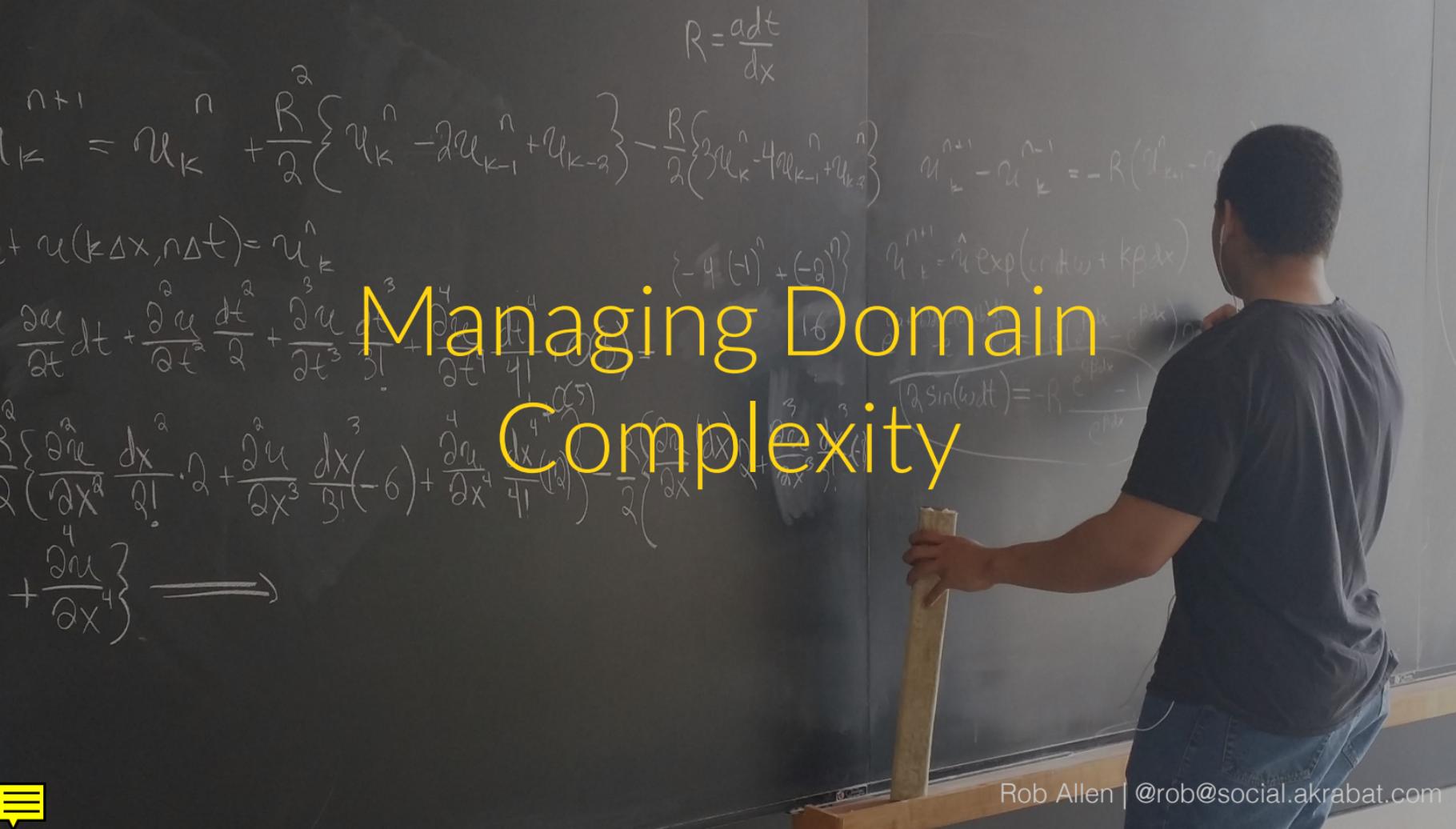
sends

is displayed on

marks success
by storing

populates





$$R = \frac{adt}{dx}$$

$$u_{k+1} = u_k + \frac{R}{2} \{ u_k - 2u_{k-1} + u_{k-2} \} - \frac{R}{2} \{ u_k - 4u_{k-1} + u_{k-2} \}$$

$$u_k^{n+1} - u_k^{n-1} = -R(u_{k+1}^n - u_{k-1}^n)$$

$$u(k\Delta x, n\Delta t) = u_k^n$$

$$\{ -4(-1)^n + (-2)^n \}$$

$$u_k^{n+1} = \hat{u} \exp(\dots)$$

Managing Domain Complexity

$$\frac{\partial u}{\partial t} + \frac{\partial^2 u}{\partial t^2} \frac{dt}{2} + \frac{\partial^3 u}{\partial t^3} \frac{dt^3}{6} + \dots$$

$$\frac{\partial^2 u}{\partial x^2} \frac{dx^2}{2!} \cdot 2 + \frac{\partial^3 u}{\partial x^3} \frac{dx^3}{3!} (-6) + \frac{\partial^4 u}{\partial x^4} \frac{dx^4}{4!} (24) - \frac{1}{2} \left\{ \frac{\partial u}{\partial x} \right\}$$

$$+ \frac{\partial u}{\partial x} \Bigg\} \implies$$



Bounded Contexts

define the boundary for a model



Bounded Contexts

*protect the domain model from
dilution*

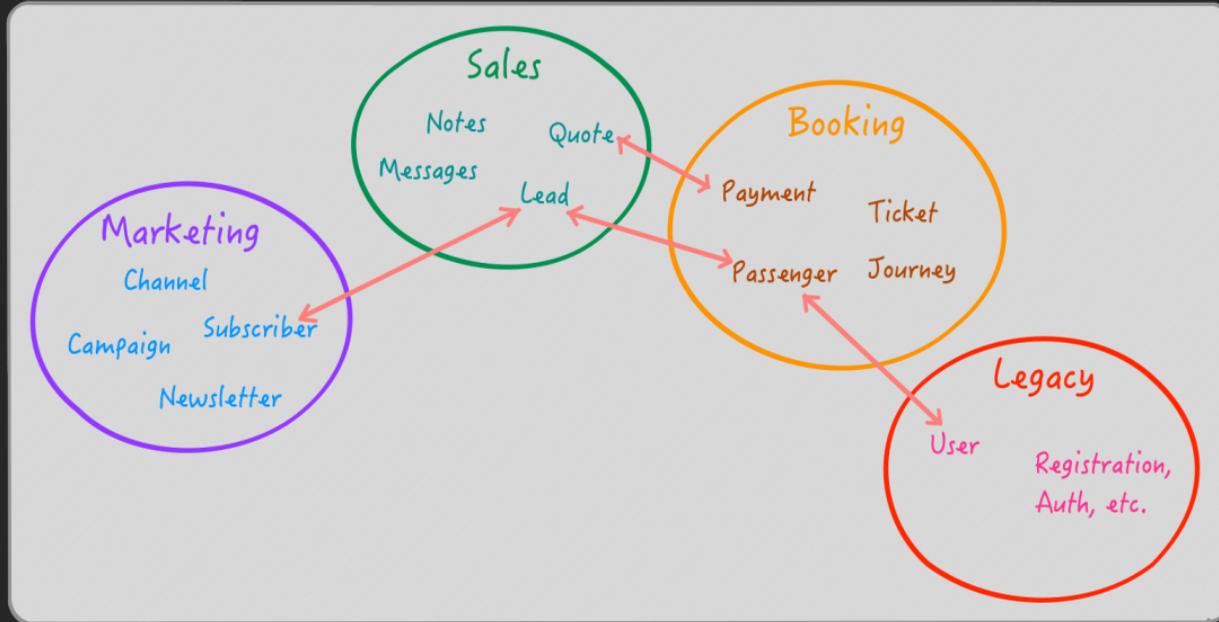


Bounded Contexts

are composed into an application



Context maps



Tactical design



Tactical Design

*Entities, Value Objects &
Aggregates*



Tactical Design

Storing State



Tactical Design

Services



Tactical Design

Architecture



To sum up



"Domain-Driven Design is about creating shared understanding of the problem space that is reinforced ubiquitously via conversations, code and diagrams."

Nick Tune





Thank you!

Photo Credits

- Event Storming: H Schwentner, <https://commons.wikimedia.org/w/index.php?curid=57766348>
- Story Mapping: D Fujihara, <https://www.flickr.com/photos/49942291@N06/6271934371>