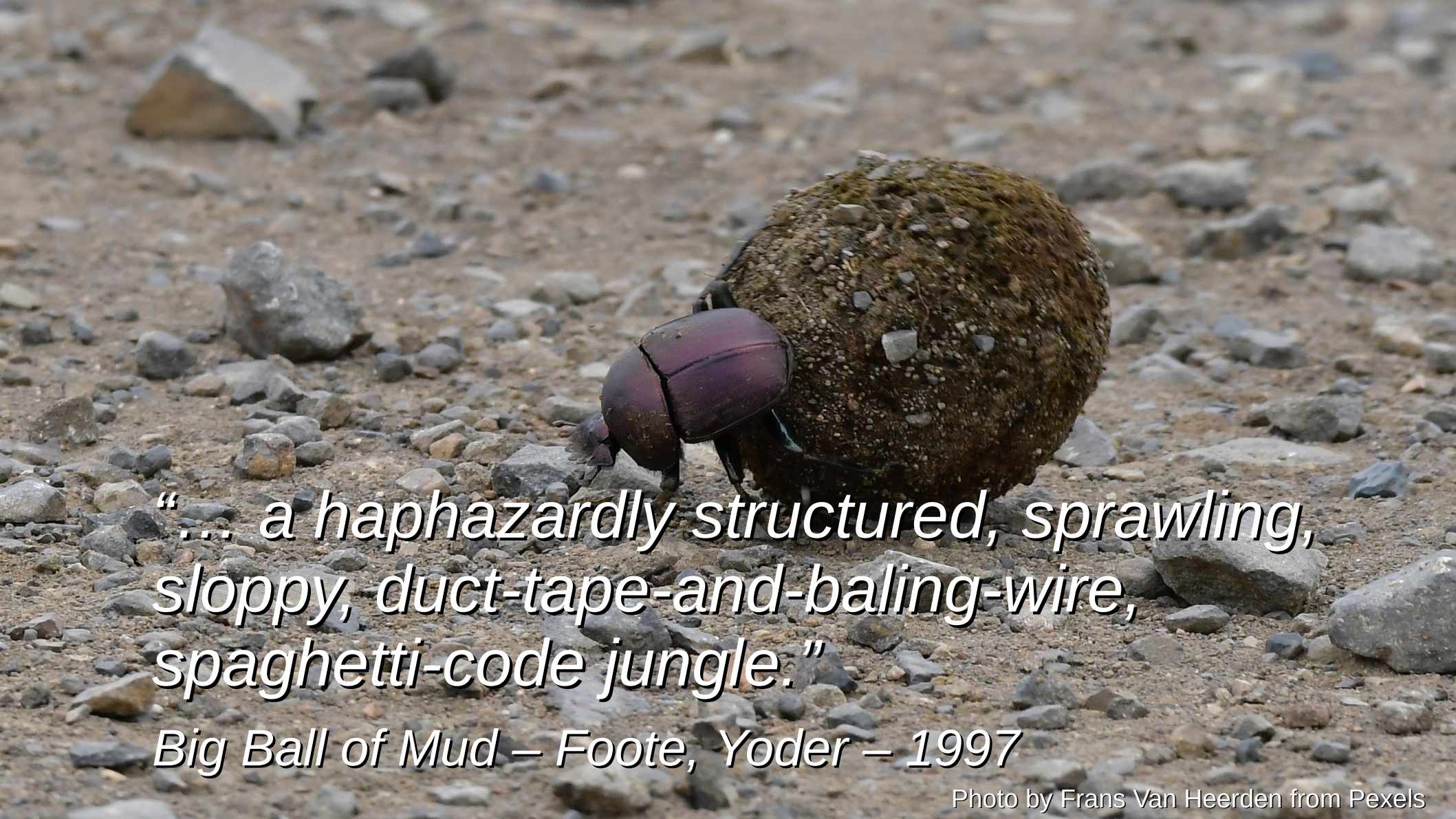
«Any fool can write code that a computer can understand. Good programmers write code that humans can understand.»

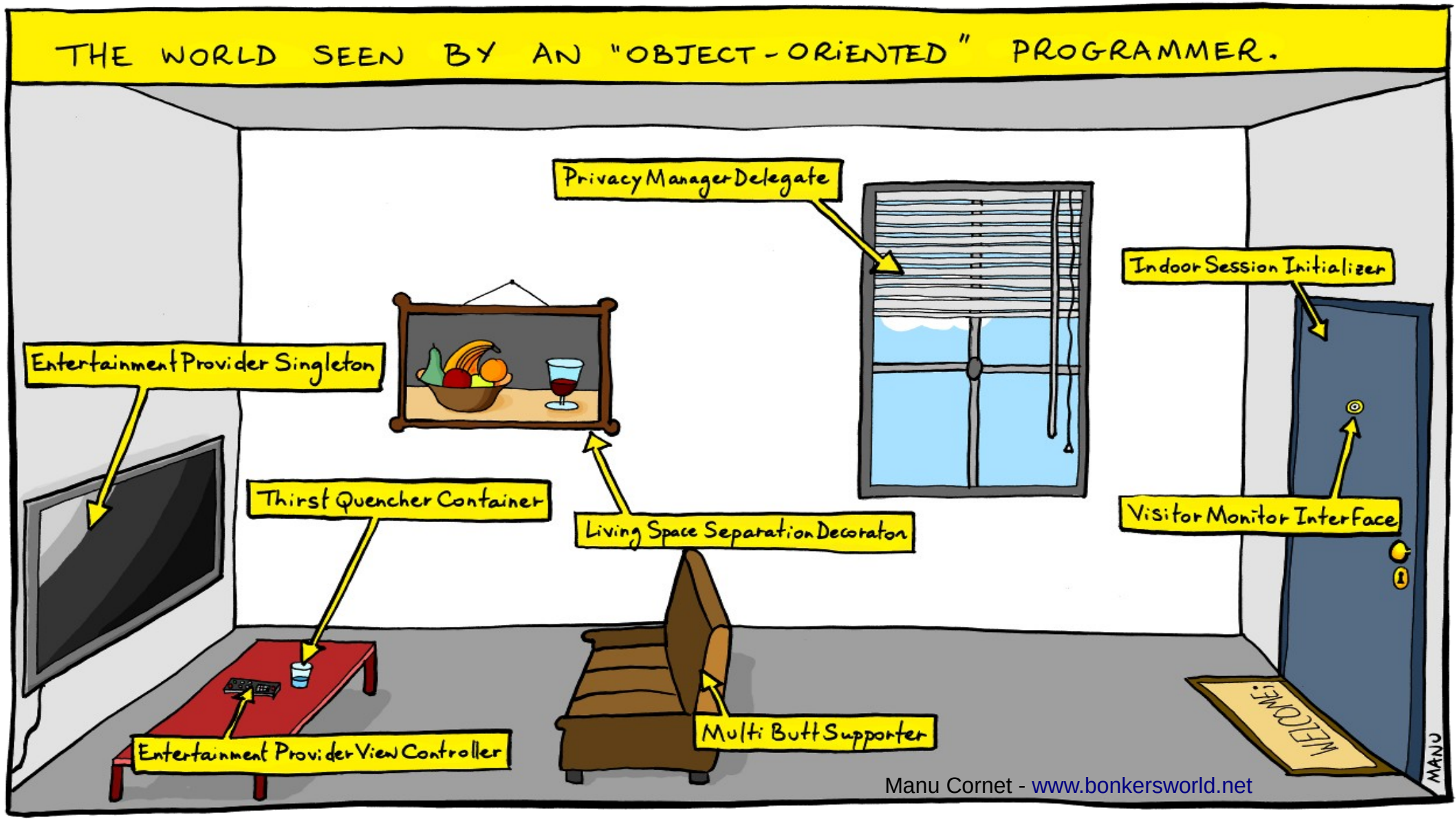
- Martin Fowler

Agenda

- DDD Europe 2018
- Meetup groups
- What is DDD?
- Patterns
- When to use DDD?
- Interested in more?

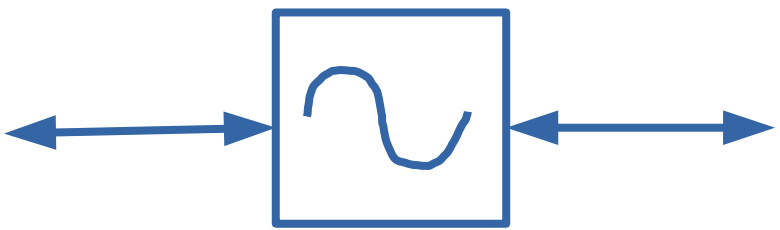
- Domain-Driven Design Norway
- Oslo Software Architecture (OSWA)
- Oslo PostgreSQL User Group
- DEVOPS NORWAY
- Oslo Embedded Software Meetup (just woke up)
- Oslo C++ Users Group (sleeping)

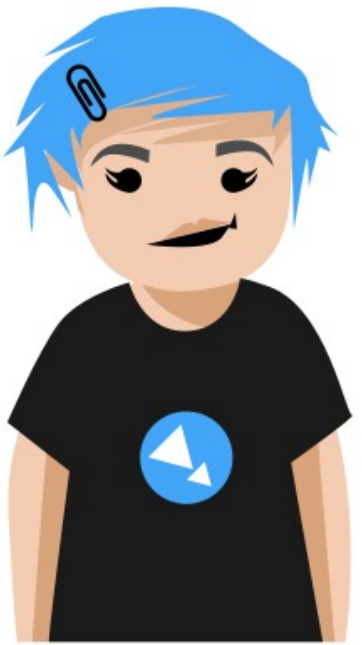

“… a haphazardly structured, sprawling, sloppy, duct-tape-and-baling-wire, spaghetti-code jungle.”
Big Ball of Mud – Foote, Yoder – 1997
Photo by Frans Van Heerden from Pexels


THE WORLD SEEN BY AN "OBJECT-ORIENTED" PROGRAMMER.
PrivacyManagerDelegate
IndoorSessionInitializer
EntertainmentProviderSingleton
ThirstQuencherContainer
LivingSpaceSeparationDecoraton
VisitorMonitorInterface
EntertainmentProviderViewController
MultiButtSupporter
WELCOME!
MANU
Manu Cornet - www.bonkersworld.net

Domain complexity
(essential)

Technical complexity
(accidental)



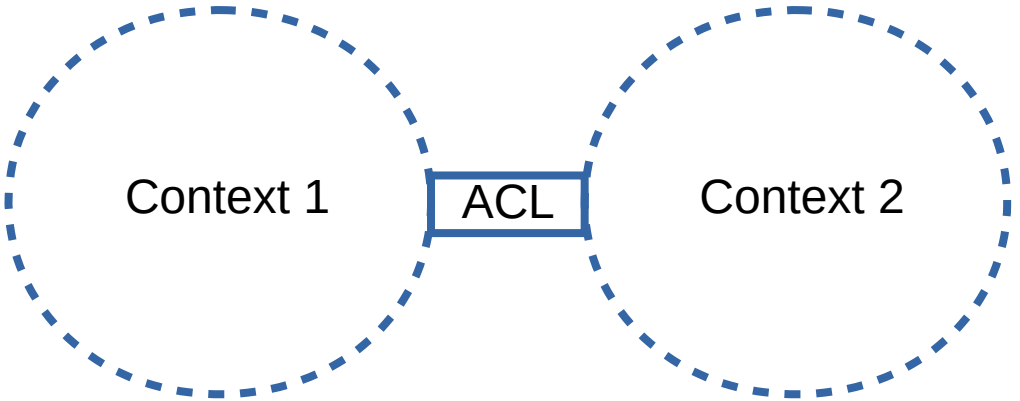


Tackling complexity in the heart of software

Illustrations by rootworks on openclipart

One ubiquitous language per bounded context

- Stream
- Input
- Output
- Source
- User
- Filter
- Service
- Representation
- Channel
- Coder
- Language
- Subtitle
- ID
- Segment



Domain model

- Eric Evans' keynote on DDD Europe 2018
- Joda-Time (Java Date and Time library)
- January 31, 2018 + one month? February 28, 2018
- January 31, 2018 + two months? March 31, 2018
- January 31, 2018 + one month + one month? March 28, 2018
- Calculate Thanksgiving Day (Fourth Thursday in November)

Patterns?

- Tactics
  - Value objects
  - Entities
  - Aggregates
  - Services
  - Factories
  - Repositories
  - CQRS
- Strategy
  - Context maps
  - Event storming
  - Bounded context
  - Ubiquitous language
  - Domain model

Value objects

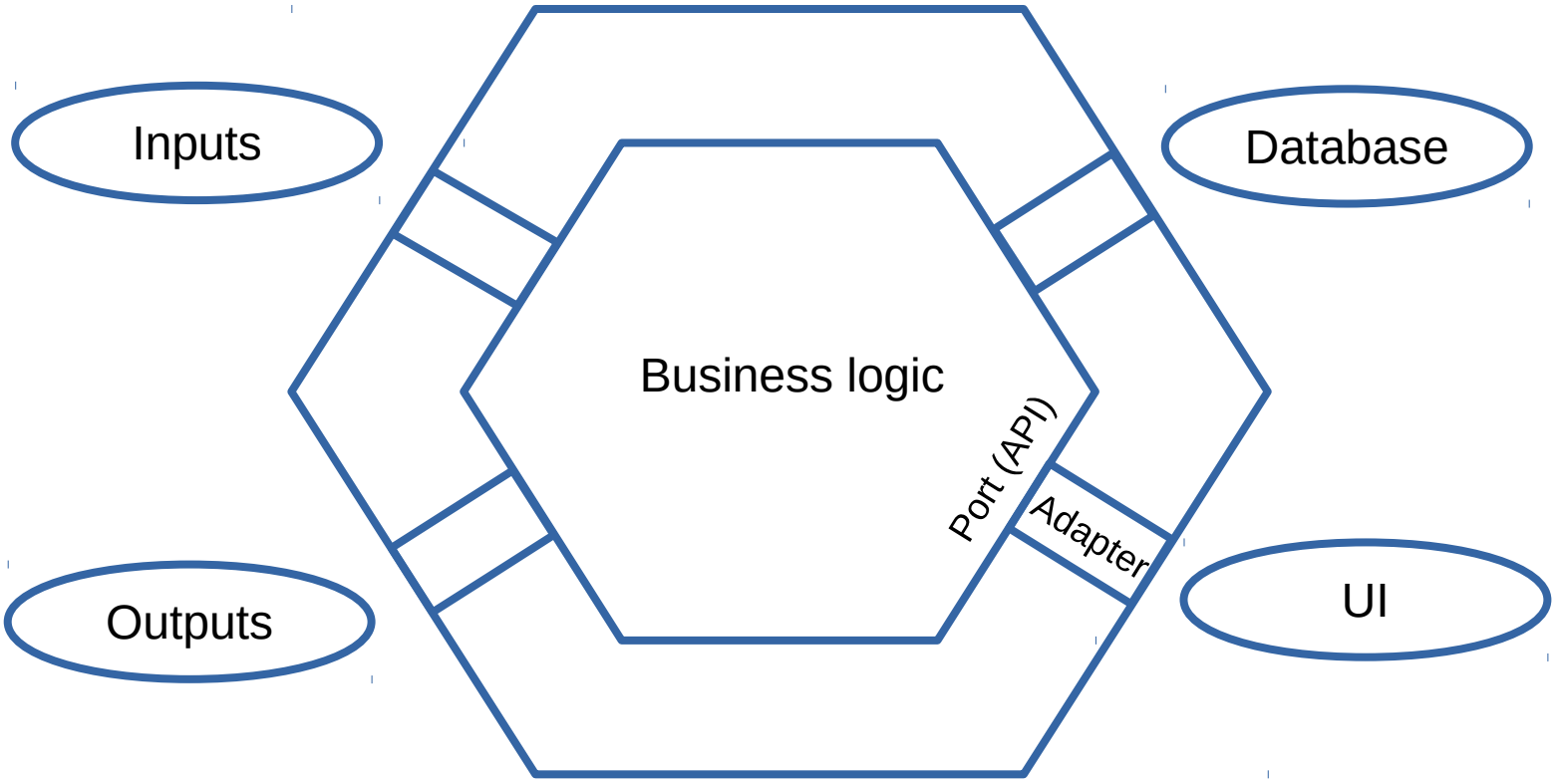
```
const int day = getDay();
```

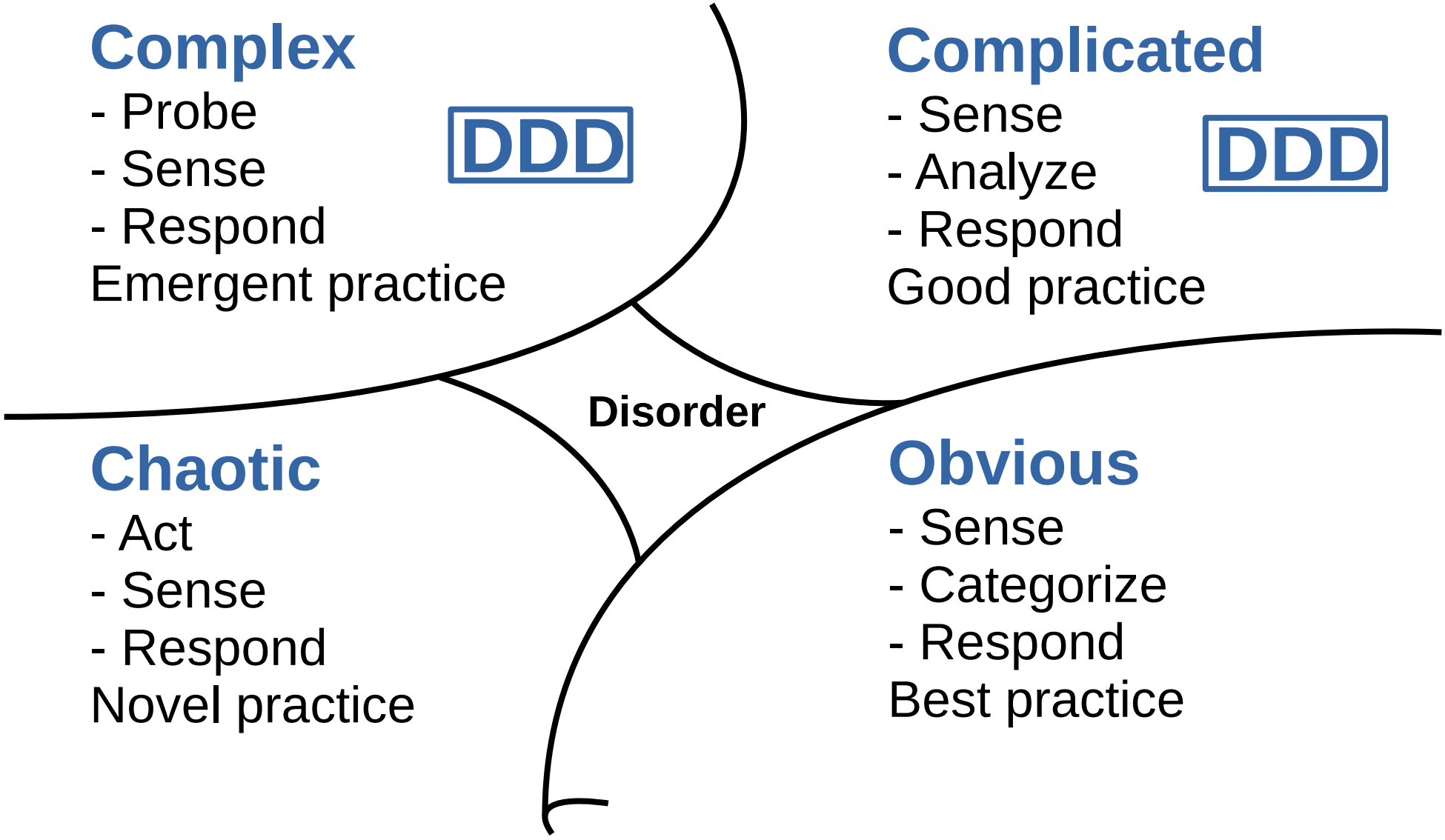
Value objects

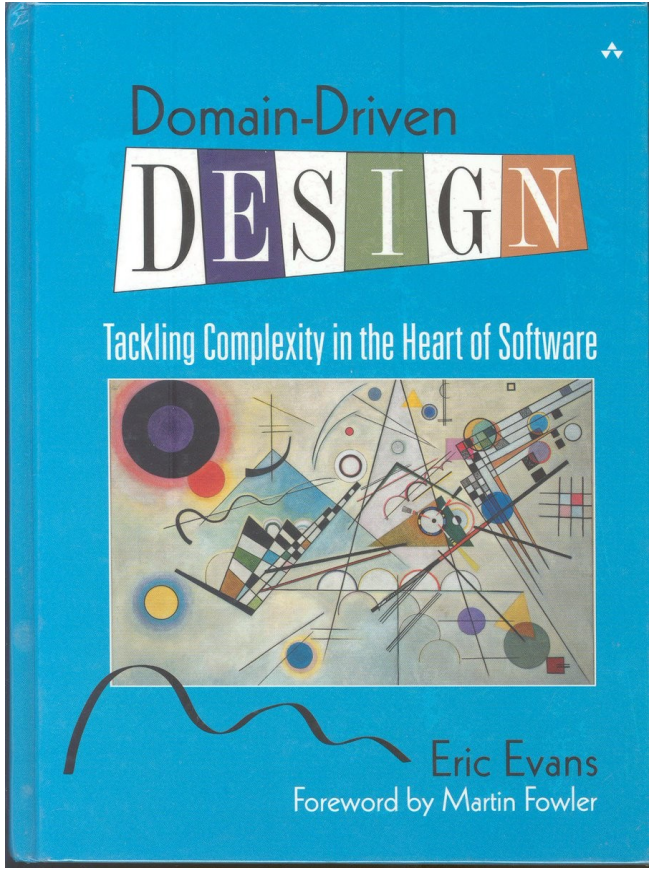
```
using Day = int;
const Day day = getDay();
if (day >= 1 && day <= 31)
{
  // continue processing here ...
}
```

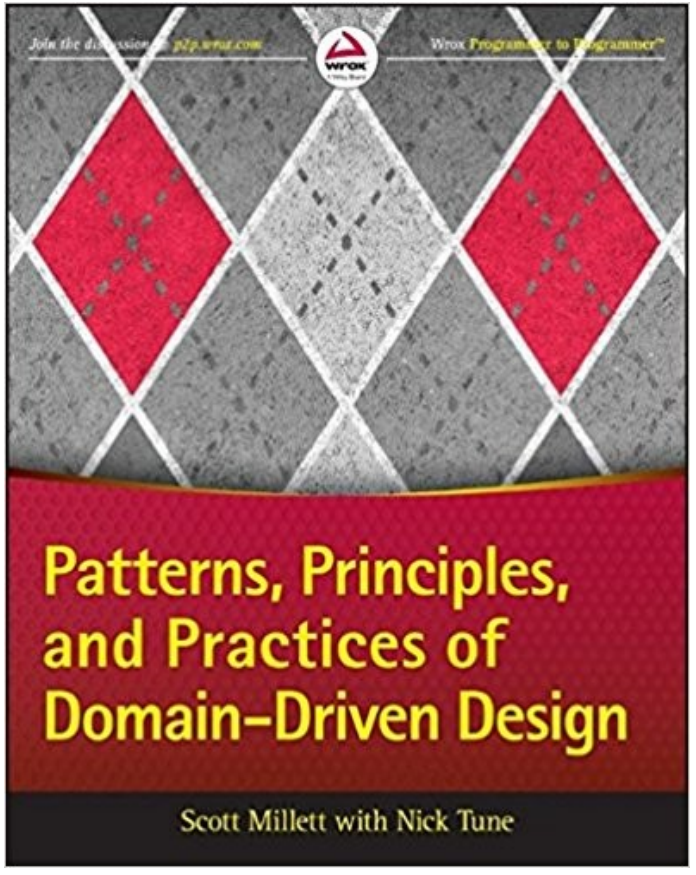
```
class DayOfMonth
{
public:
  DayOfMonth(const int day)
  : m_day(day)
  {
    const bool dayIsValid = (day >=1 && day <=31);
    if (!dayIsValid)
    {
      throw std::invalid_argument("Invalid day: " + std::to_string(day));
    }
  }

private:
  const int m_day;
};

const DayOfMonth dayOfMonth(getDay());
```

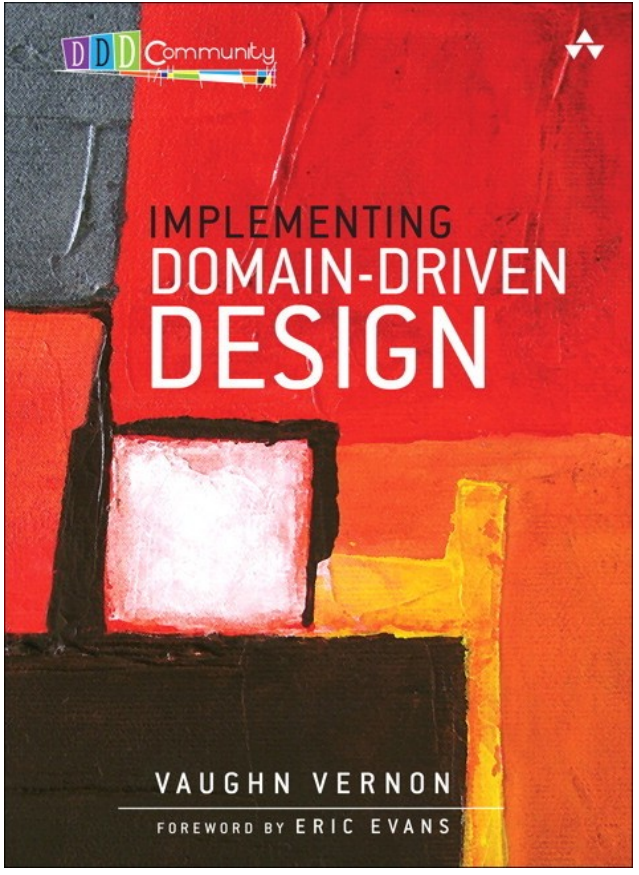
Ports and adapters




Complex
- Probe
- Sense
- Respond
Emergent practice
DDD
Complicated
- Sense
- Analyze
- Respond
Good practice
DDD
Disorder
Chaotic
- Act
- Sense
- Respond
Novel practice
Obvious
- Sense
- Categorize
- Respond
Best practice


Domain-Driven
DESIGN
Tackling Complexity in the Heart of Software
Eric Evans
Foreword by Martin Fowler


Wrox Programmer to Programmer™
Patterns, Principles, and Practices of Domain–Driven Design
Scott Millett with Nick Tune


DDD Community
IMPLEMENTING
DOMAIN-DRIVEN
DESIGN
VAUGHN VERNON
FOREWORD BY ERIC EVANS

Interested in more?

- Cynefin framework – Prof. Dave Snowden
- Event storming – Alberto Brandolini
- Wardley maps – Simon Wardley
- Bill Poole's Creative Abrasion – Bill Poole
- Event sourcing – Martin Fowler
- Conway’s law – Melvin Conway