

DDD Journey

Apprenticeships for the aspiring software craftsman
<http://www.beingtheworst.com/>

Outline

- What is DDD?
- Creating a Domain Model
 - Extending the Model
- Commands
 - Messages
- Persistence

DDD in a nutshell

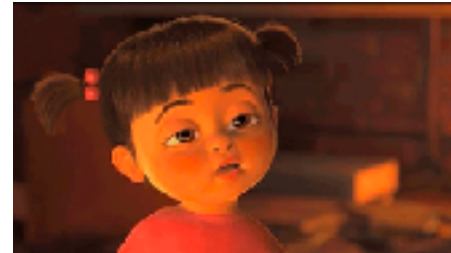
- Nothing new!
- Context & Language
 - Use the words of the customer - no tech talk
- Language & Code
 - The language should be visible in the code
- Make the implicit explicit

- Wald: Jäger: Schale: die Klauen des wiederkäuenden Haarwildes
- Theater: In Schale werfen -> Anzug
- Am Couchtisch mit Obst: Mandarine/Orange
- Beim Kochen: Schale mit Zutaten

What happens on click?

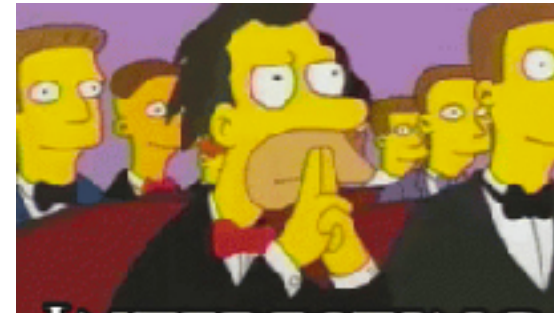
1. Validation

- A. Is the data entered?
- B. Is the data format correct?



2. Business-Logic

- C. Is the operation valid at the current state?
- D. What are the side effects of the operation?



Assignment



- Implement the Button-Click
 - Ignore Validation (Assume only valid data arrives on the server)
 - Derive a "Priority" (depending on the date)
- Ignore infrastructure code
 - Implement/Called in a Test
 - Compile is enough for now

```
public interface IRepository<T>
{
    void Save(T entity);
}
```

Priorität!!

Observations?

- Names?
- How is the code structured?
- Where is the logic/"action" implemented?
- Information hiding?

Getting things done



- Allow users to finish a task

Commands

- Tell a component to do something
 - "Persisted Function Call"

```
service.CreateTodo(  
    "Sommerreifen aufziehen",  
    new DateTime(2018, 4, 15)  
);
```

```
var command = new ComposeTodoCommand  
{  
    Task = "Sommerreifen aufziehen",  
    Until = new DateTime(2018, 4, 15)  
};  
...  
service.When(command);
```

- Are usually “valid” if they arrive on the server
- Have no return values
 - Commands don't fail

Assignment - implement command

Benefit of Commands

- Uniform interface to handle them
 - `void handle(T message)`
- Easy to build cross cutting functions
 - Logging, Persistence, Queuing, Chain-of-Responsibility, ...
- Treats data as data
 - even if the data is only of temporary nature

Herzeigen von möglichkeiten

What about Persistence?

- Goal: make Domain Objects persistence transparent
 - They should focus on “Business-Logic” only
- Options:
 - “Classical” Database (Table based)
 - Document Store
 - Event Sourcing