

Domain-driven design

Tactical patterns

by @tojans

AGILE

software

1 BASIC

concept

Make the

implicit

EXPLICIT

Language
Boundaries
Code

Language

Boundaries

Code

Language

Boundaries

Code



A language

defines – and is defined by

A CONTEXT

"Language comes first.
It's not that language
grows out of consciousness,
if you haven't got language,
you can't be conscious."

- Alan Moore -

Language

Boundaries

Code

Language

Boundaries

Code

Language

Boundaries

Code



A scenic view of a mountain range with a winding road and a large bridge in the foreground. The mountains are covered in lush green vegetation, and the sky is filled with soft, white clouds. The overall atmosphere is peaceful and majestic.

Proper Boundaries

A scenic view of a winding road through a lush green valley with mountains in the background. The road is a light brown color and curves through the green landscape. The mountains in the background are blue and hazy. The sky is overcast with grey clouds.

Organisational
Dimensions

Language
Contracts
Dependencies

Race conditions
Availability
Lifecycle

Modules

Entities

Services

Factories

Value objects

Aggregates

Repositories

CQS / CQRS

Event sourcing

Language

Boundaries

Code

Language

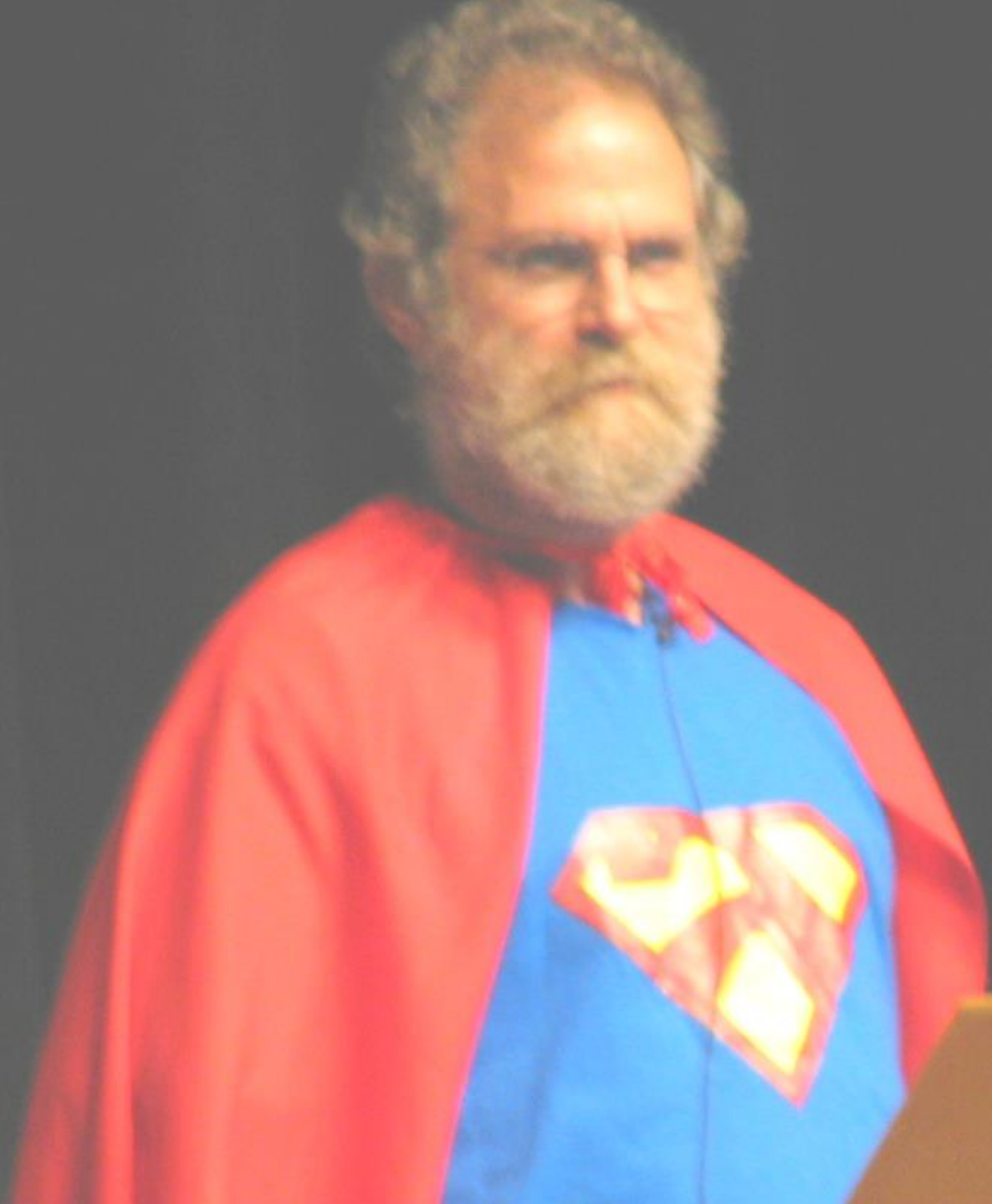
Boundaries

Code

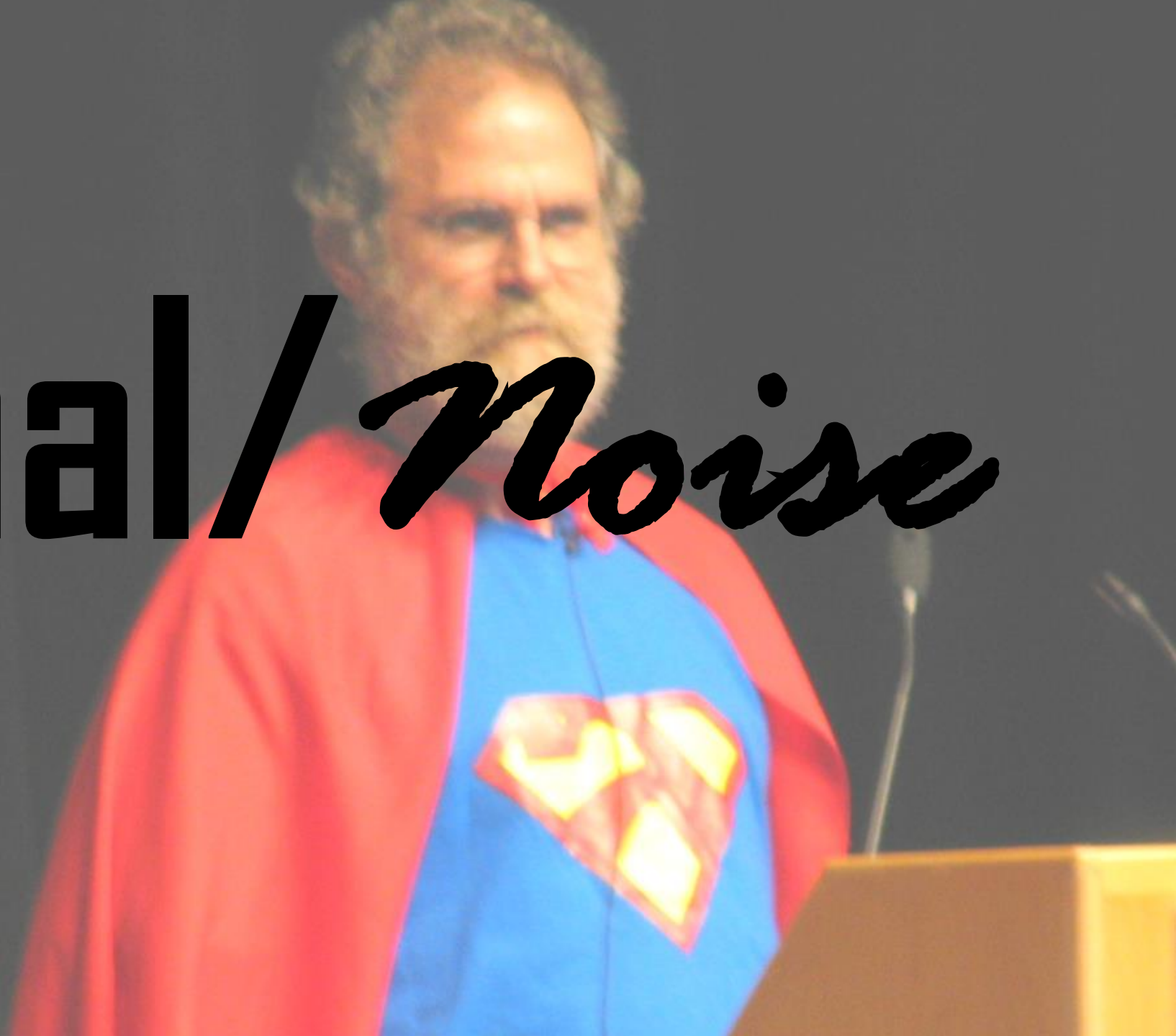
Language

Boundaries

Code



Signal/*Noise*




```
module InventoryItems(Command(..), Event(..), handle) where

import Data.Maybe(isJust)

type Id = String
type Name = String
type Amount = Int

data Command = CreateInventoryItem      Id
              | RenameInventoryItem     Id Name
              | RemoveItemsFromInventory Id Amount
              | AddItemsToInventory      Id Amount
              | DeactivateInventoryItem  Id
              deriving (Show)

data Event = InventoryItemCreated      Id
            | InventoryItemRenamed     Id Name
            | ItemsRemovedFromInventory Id Amount
            | ItemsCheckedInToInventory Id Amount
            | InventoryItemDeactivated Id
            deriving (Show, Read)

data Item = Item { itemId      :: Id
                  , itemActivated :: Bool
                  }
```

```
handle' maybeItem (CreateInventoryItem id)
  | itemExists    = error "item already created"
  | otherwise     = [InventoryItemCreated id]
  where itemExists = isJust maybeItem

handle' Nothing _command = error "please create the item first"

handle' (Just item) (DeactivateInventoryItem id)
  | deactivated    = error "already deactivated"
  | otherwise     = [InventoryItemDeactivated id]
  where deactivated = not (itemActivated item)

handle' (Just Item {itemActivated = False}) _command = error "item is deactivated"

handle' (Just item) (RenameInventoryItem id name)
  | invalidName    = error "newName"
  | otherwise     = [InventoryItemRenamed id name]
  where invalidName = name == []

handle' (Just item) (RemoveItemsFromInventory id amount)
  | invalidAmount = error "can't remove negative amount from inventory"
  | otherwise     = [ItemsRemovedFromInventory id amount]
  where invalidAmount = amount <= 0

handle' (Just item) (AddItemsToInventory id amount)
  | invalidAmount = error "must have an amount > 0 to checkin to inventory"
  | otherwise     = [ItemsCheckedInToInventory id amount]
  where invalidAmount = amount <= 0
```




```
apply events = foldl apply' Nothing events
```

```
apply' Nothing (InventoryItemCreated id)           = Just (Item id True)
```

```
apply' (Just item) (InventoryItemDeactivated id)    = Just (item {itemActivated = False})
```

```
apply' (Just item) _                               = Just item
```

Language

Boundaries

Code

Language

Boundaries

Code

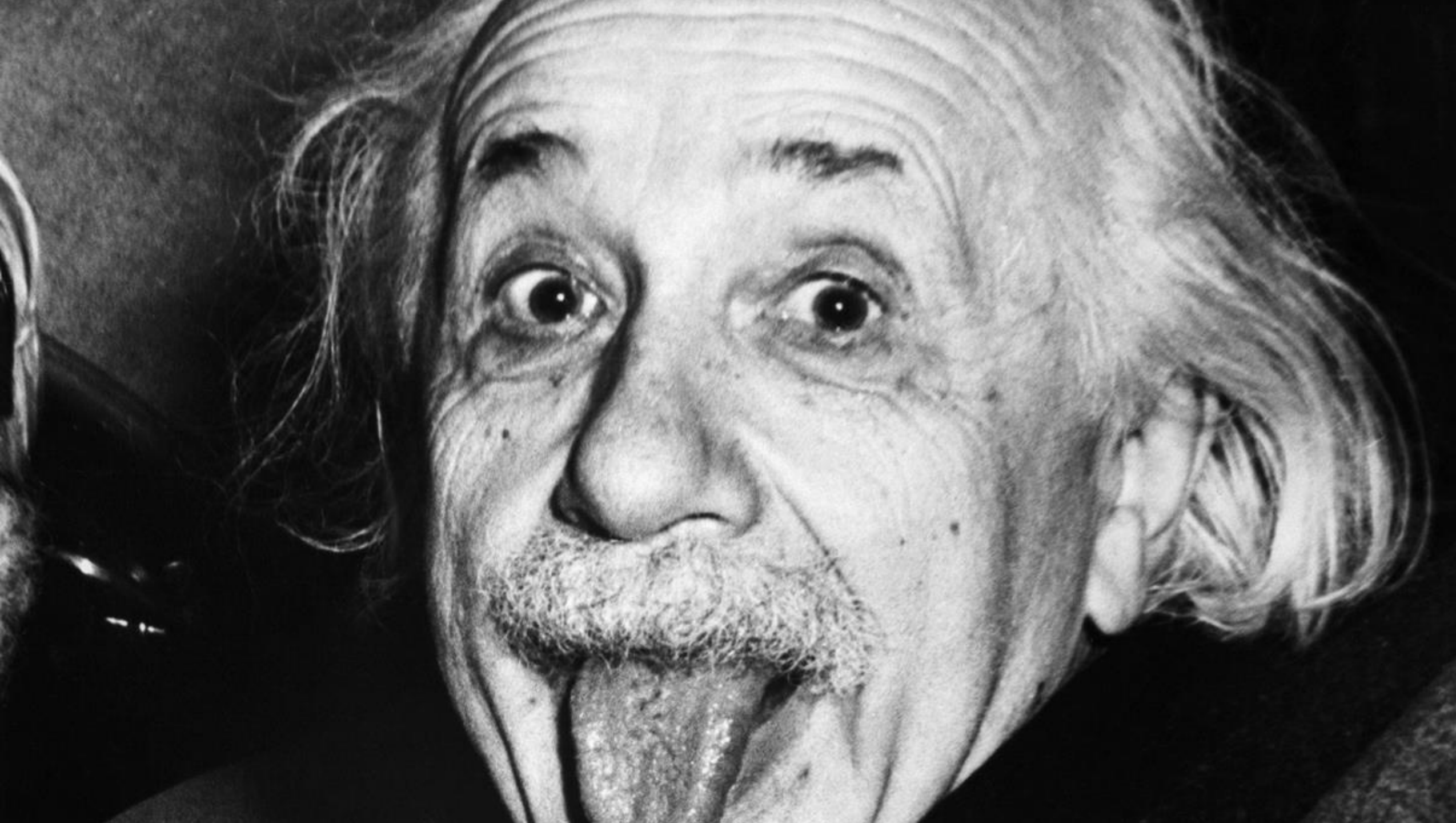
Make the

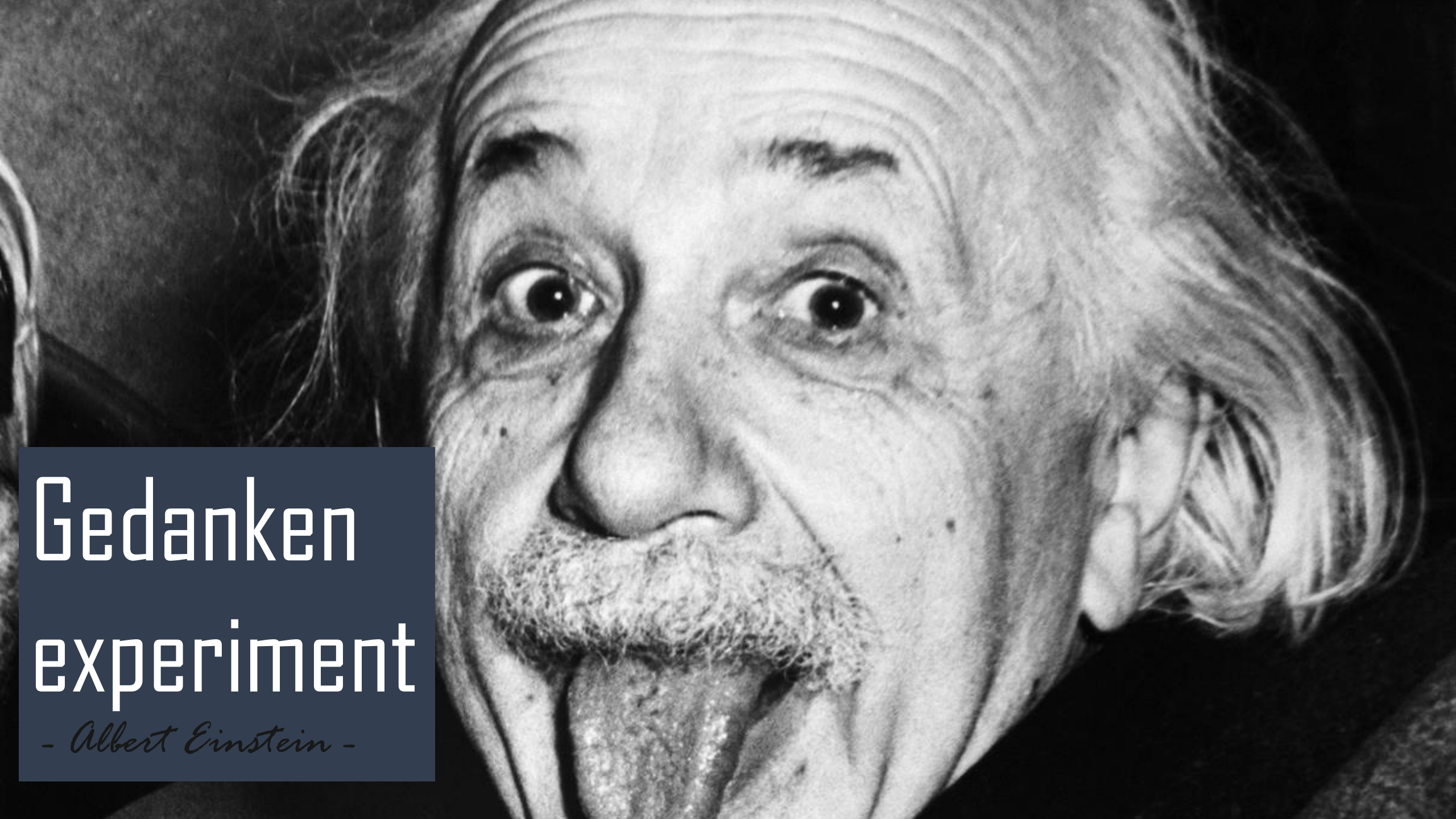
implicit

EXPLICIT

Pop Quiz

guess who





Gedanken experiment

- Albert Einstein -

***The amount of energy
necessary to refute
bullshit is an order of
magnitude bigger
than to produce it***



Software development
is a learning process
Working code
is a side effect.

- Alberto Brandolini -







"When we set out to write software,
we never know enough."

- Eric Evans -

Questions?

@tojans