Gearing architecture for agility
CodeMonsters Sofia 2018-11 // Christian Heger

Christian Heger

@zyklotrop
linkedin.com/in/christianheger/
christian.heger@zuehlke.com

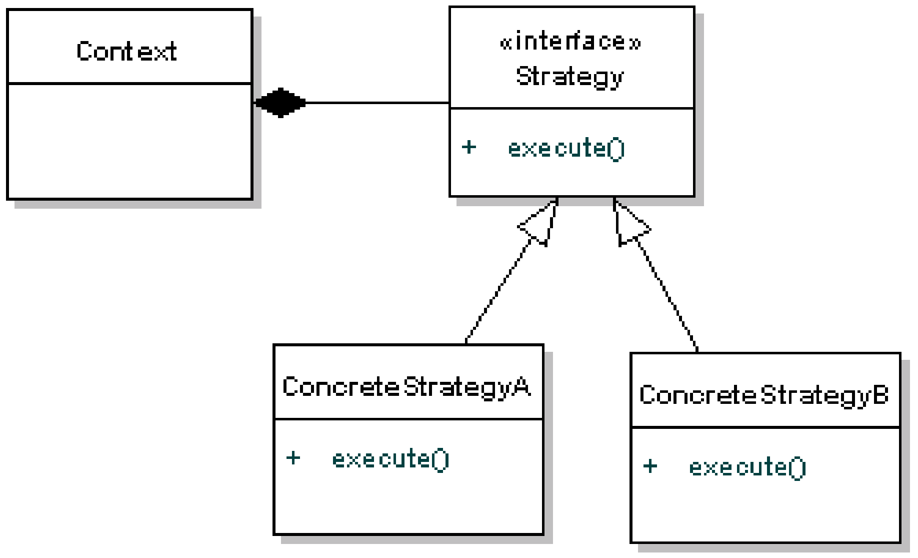
Software Architecture

- vs –

Agility

Strategies for solving the technical aspects of software engineering

so that we can embrace Agile.

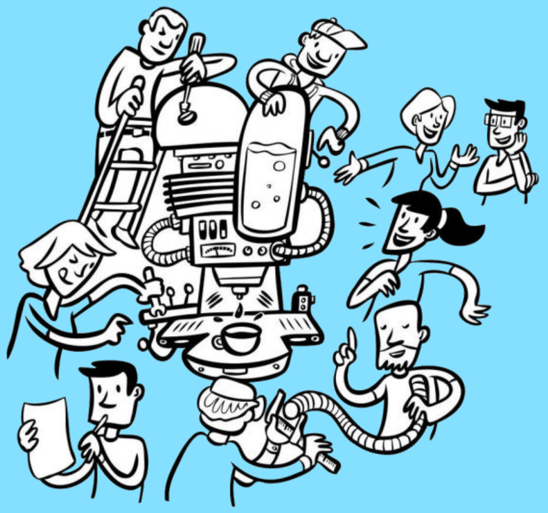


We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

That is, while there is value in the items on
the right, we value the items on the left more.

Individuals and interactions over processes and tools

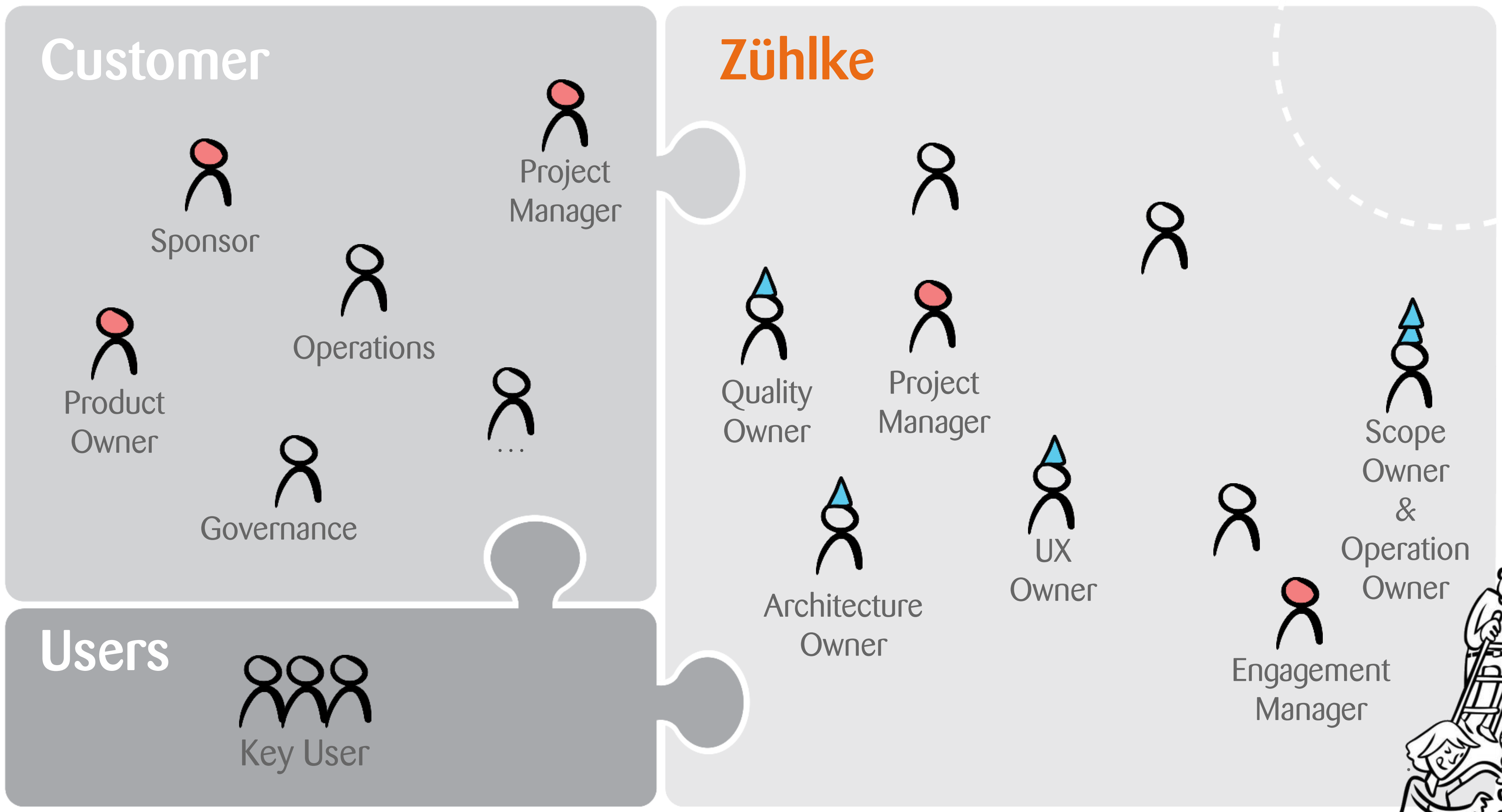


Jakob Dettner, Reiner Zenz - CC BY 2.0

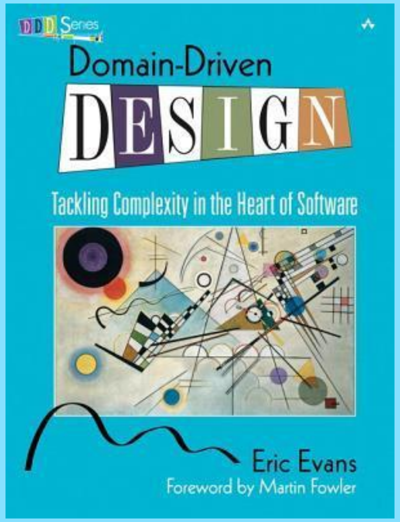
Jakob Dettner, Reiner Zenz - CC BY 2.0

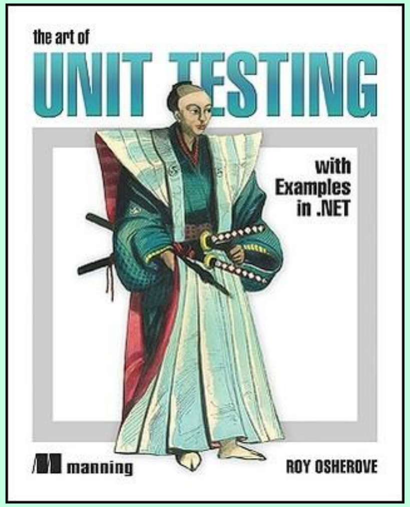
Architectural Decision! We're all going to die!

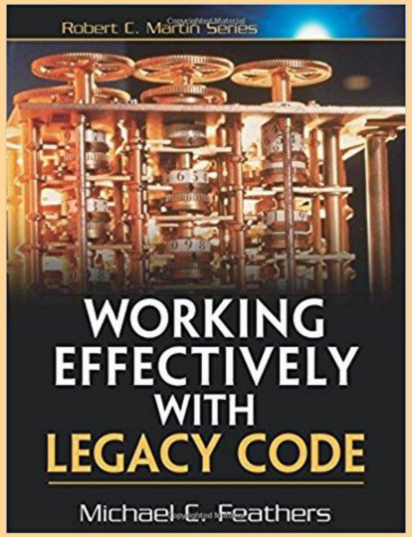
Meteor: Public domain – NASA
Dino: CC0 / no attribution

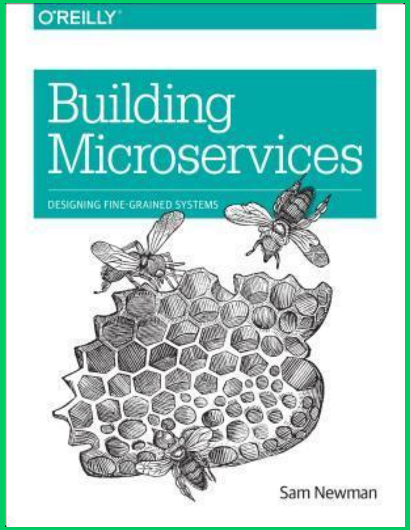

Customer
Sponsor
Project Manager
Operations
Product Owner
. . .
Governance
Users
Key User
Zühlke
Quality Owner
Project Manager
Architecture Owner
UX Owner
Scope Owner & Operation Owner
Engagement Manager

Responding to change over following a plan


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

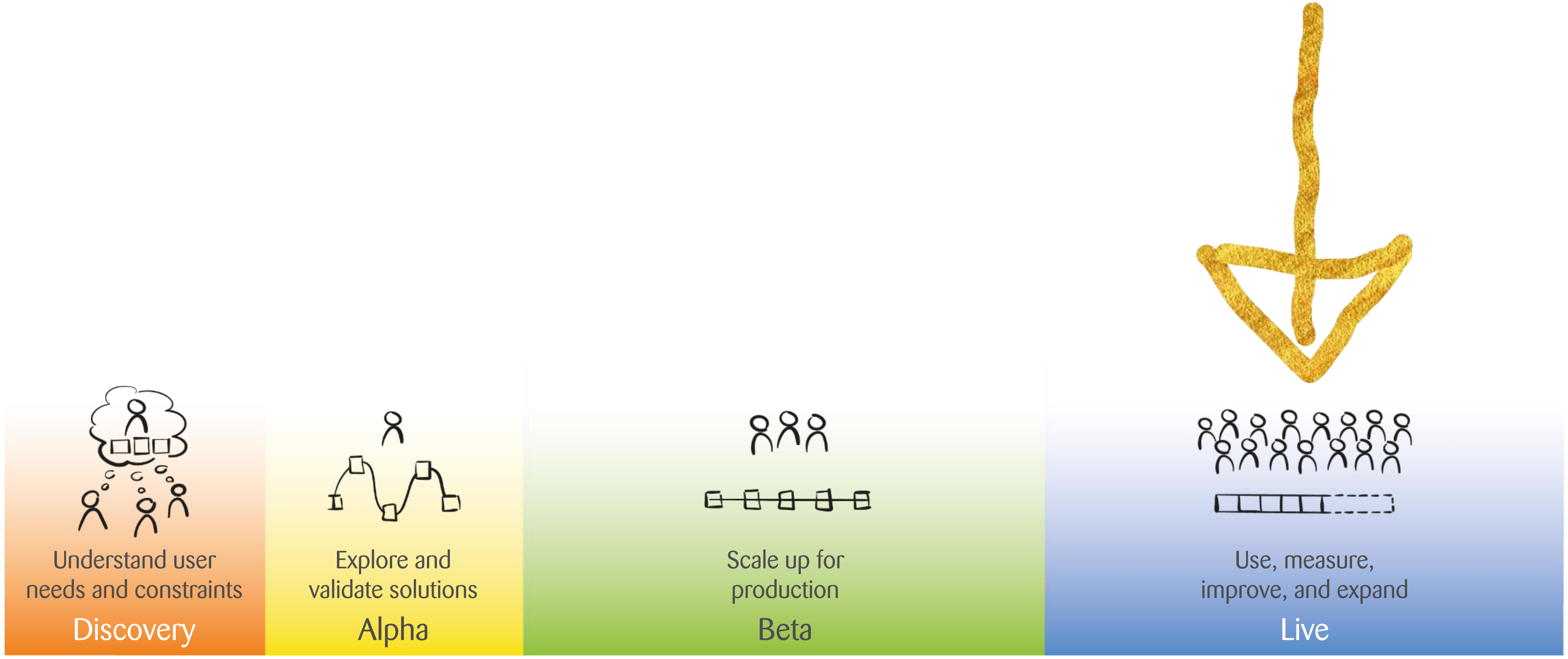






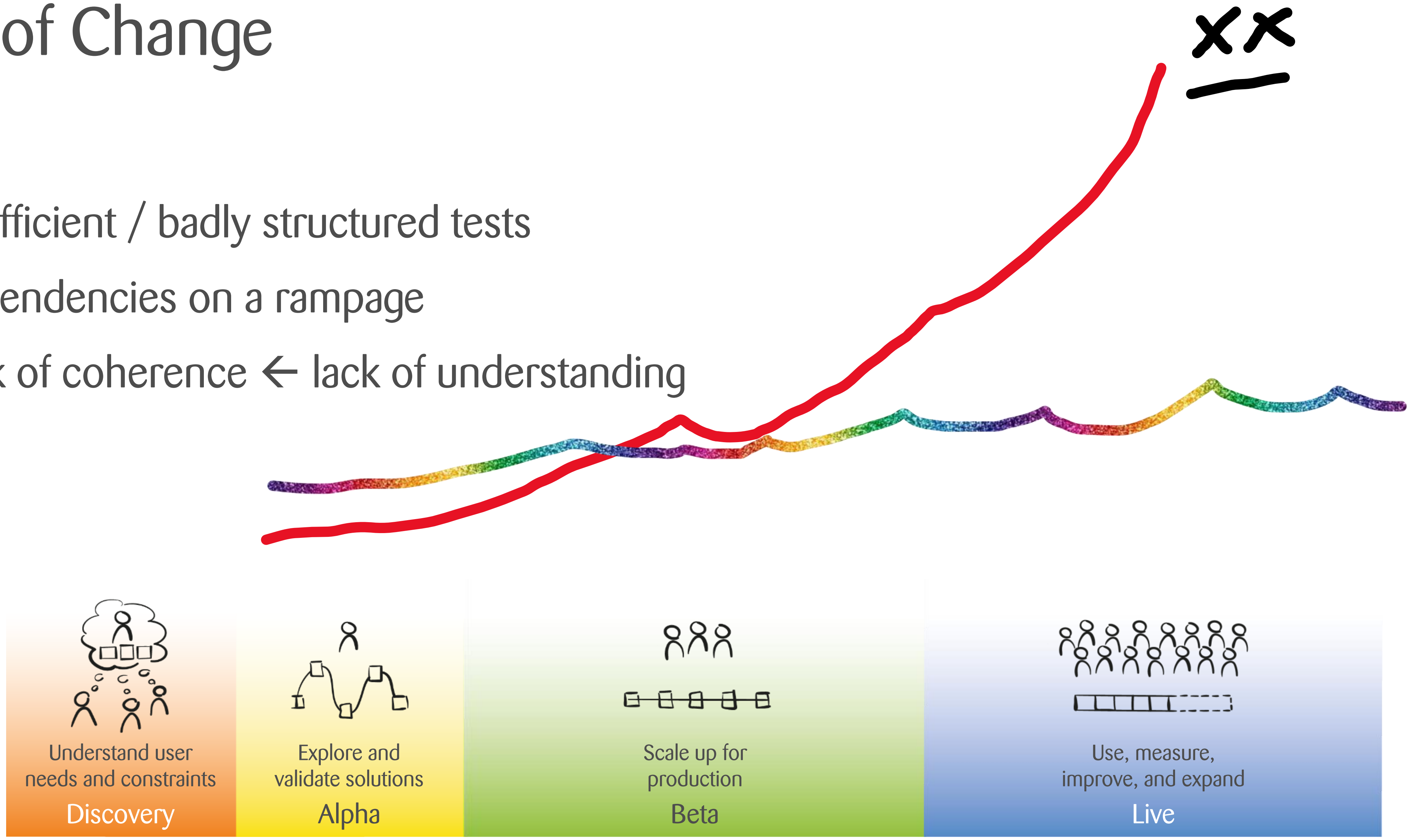
Cost of Change

- Being useful
- Exposed to real users → learning things
- Earning money


Understand user needs and constraints
Discovery
Explore and validate solutions
Alpha
Scale up for production
Beta
Use, measure, improve, and expand
Live

Cost of Change

- Insufficient / badly structured tests
- Dependencies on a rampage
- Lack of coherence ← lack of understanding



The Pyramid

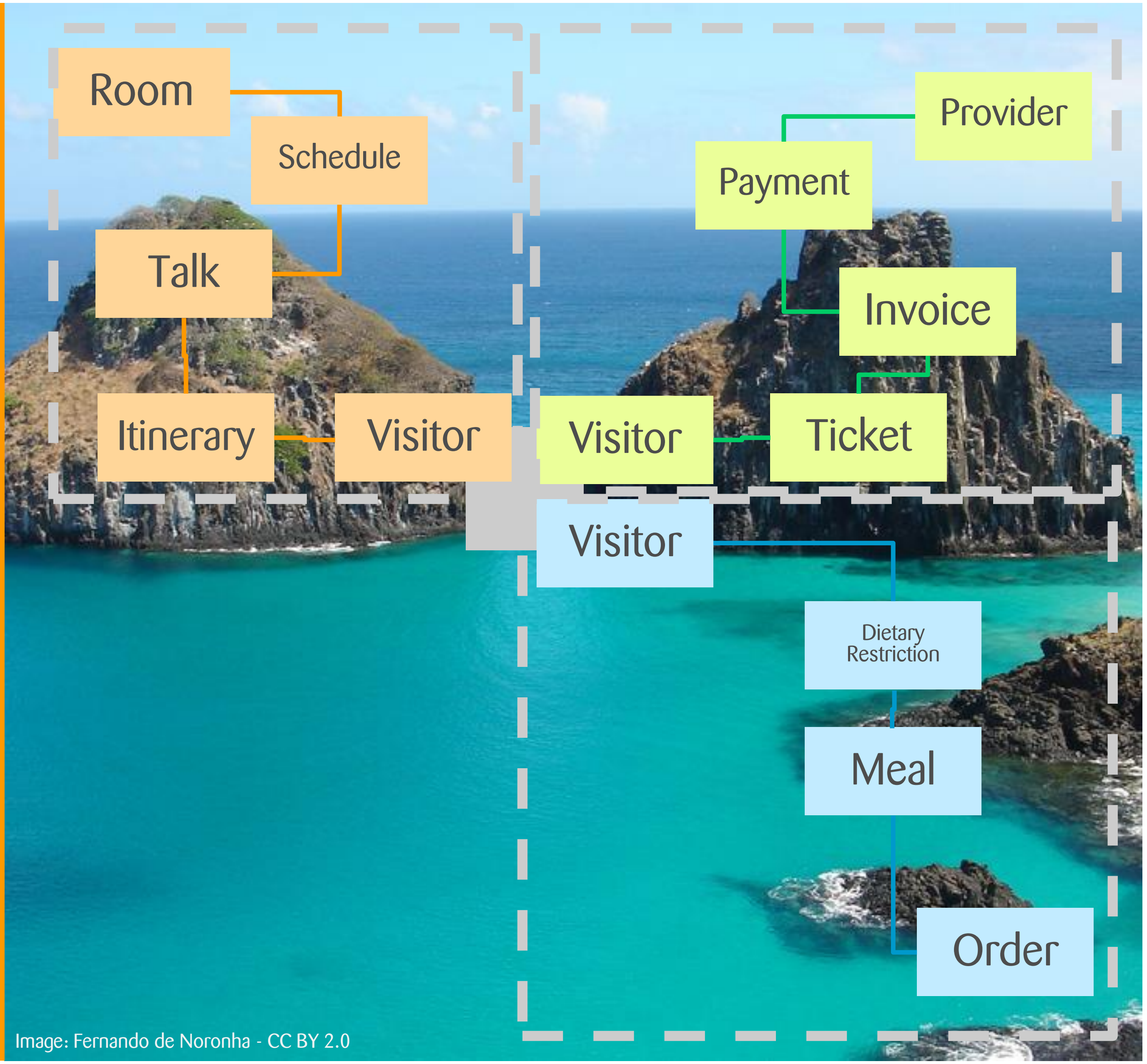
Wikipedia – User Chales J. Sharp / CC BY-SA 3.0

The Cupcake

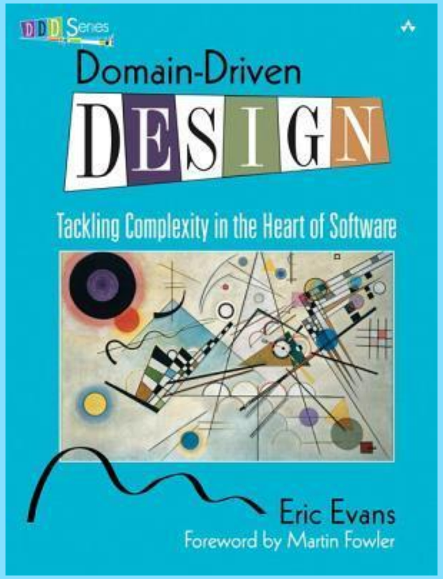
Se
the art of
UNIT TESTING
with Examples in .NET
manning
ROY OSHEROVE

The Doomsday Cloud

Bounded Contexts



Customer collaboration over contract negotiation


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

PACT FOUNDATION

Knowledge Crunching

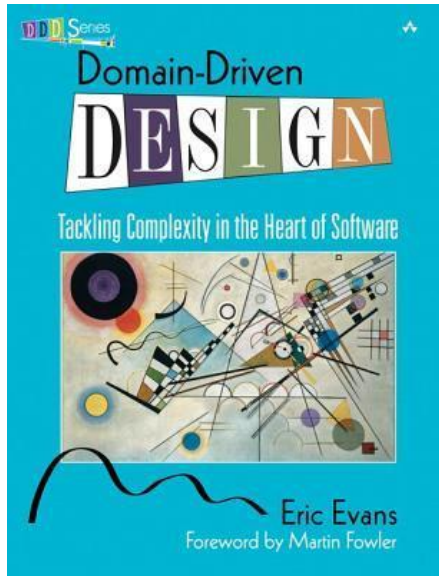
Continuous Learning

Knowledge-Rich Design

Deep Models

Ubiquitous Language

Explanatory Models


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

MAKE
Model Line
Vehicle Type
Inspection
Additional Work
Garage
Motor Oil
Booking
Offer Factory
Work Your Service
An additional work belongs to a VehicleType
An inspection belongs to a VehicleType
Own Image

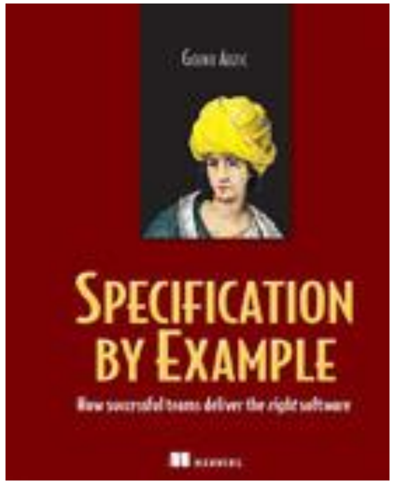
Specification by Example // Gherkin DSL

Natural language executable specification

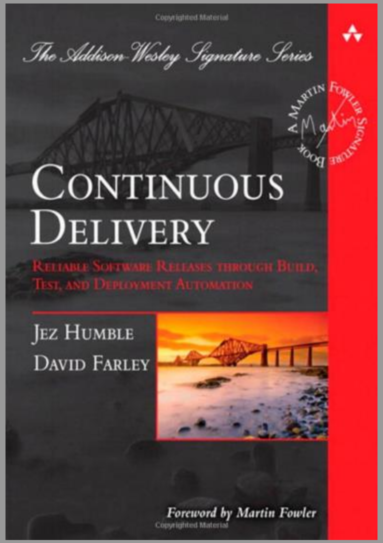
```
Feature:
    As the Professional
    I can see audiogram changes when I change channel levels

Scenario: Increase level on channel #1
    Given Hearing instrument “elia S” is mounted on left ear
    And Hearing instrument is reset
    And Baseline-Profile #5 is applied
    When left channel #1 level is increased by 5dB
    Then left audiogram at 120Hz is between 8dB and 10dB


Scenario: Increase level on channel #2
    …
```

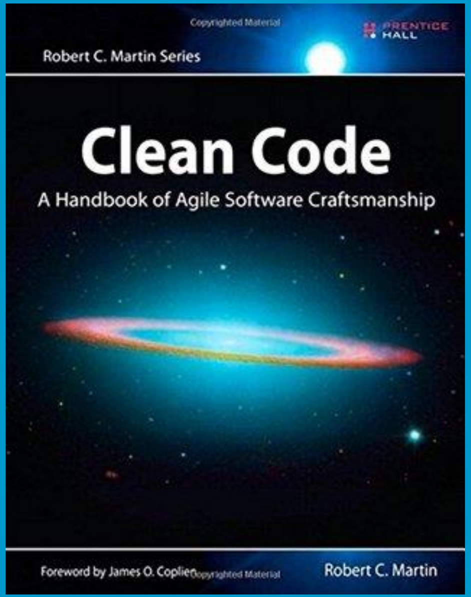


Working software over comprehensive documentation

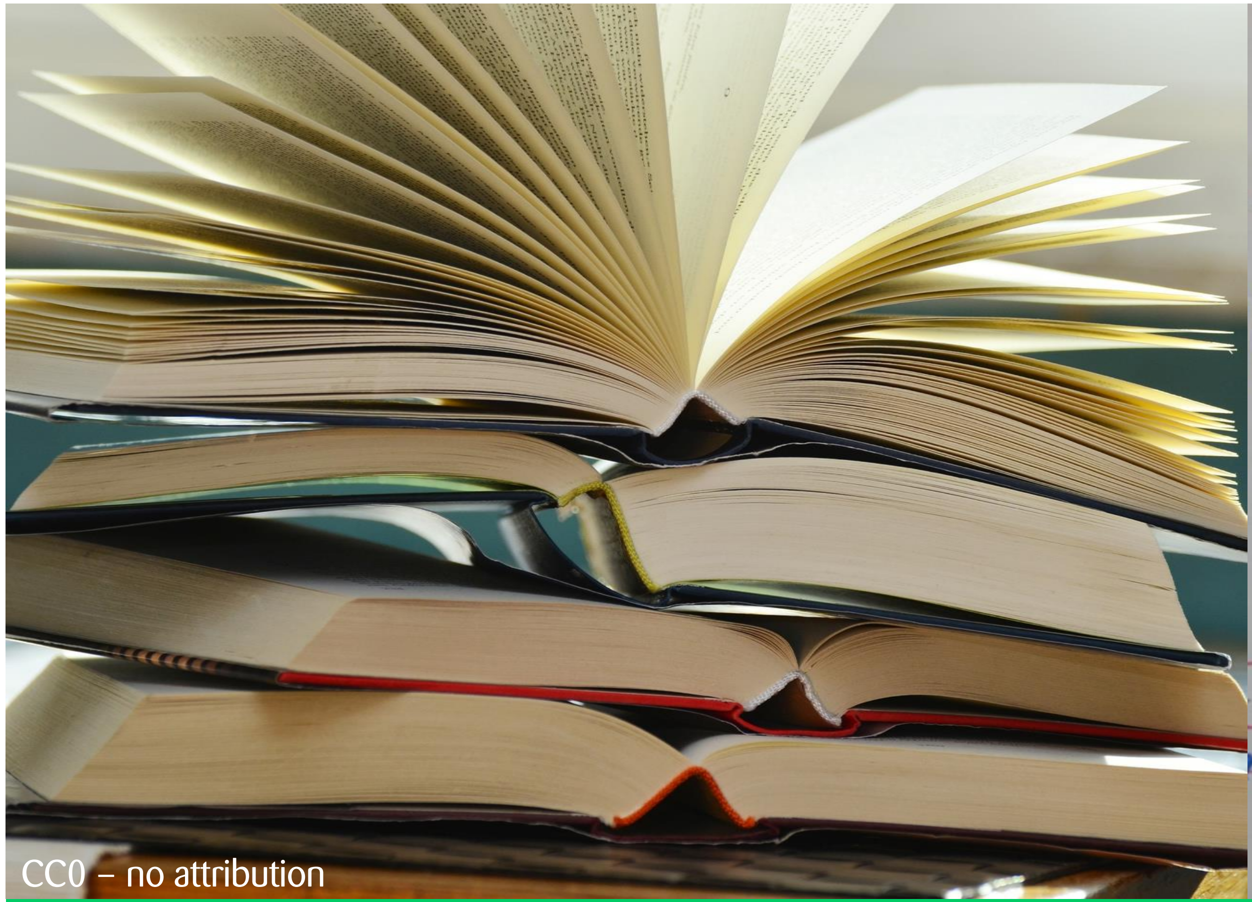


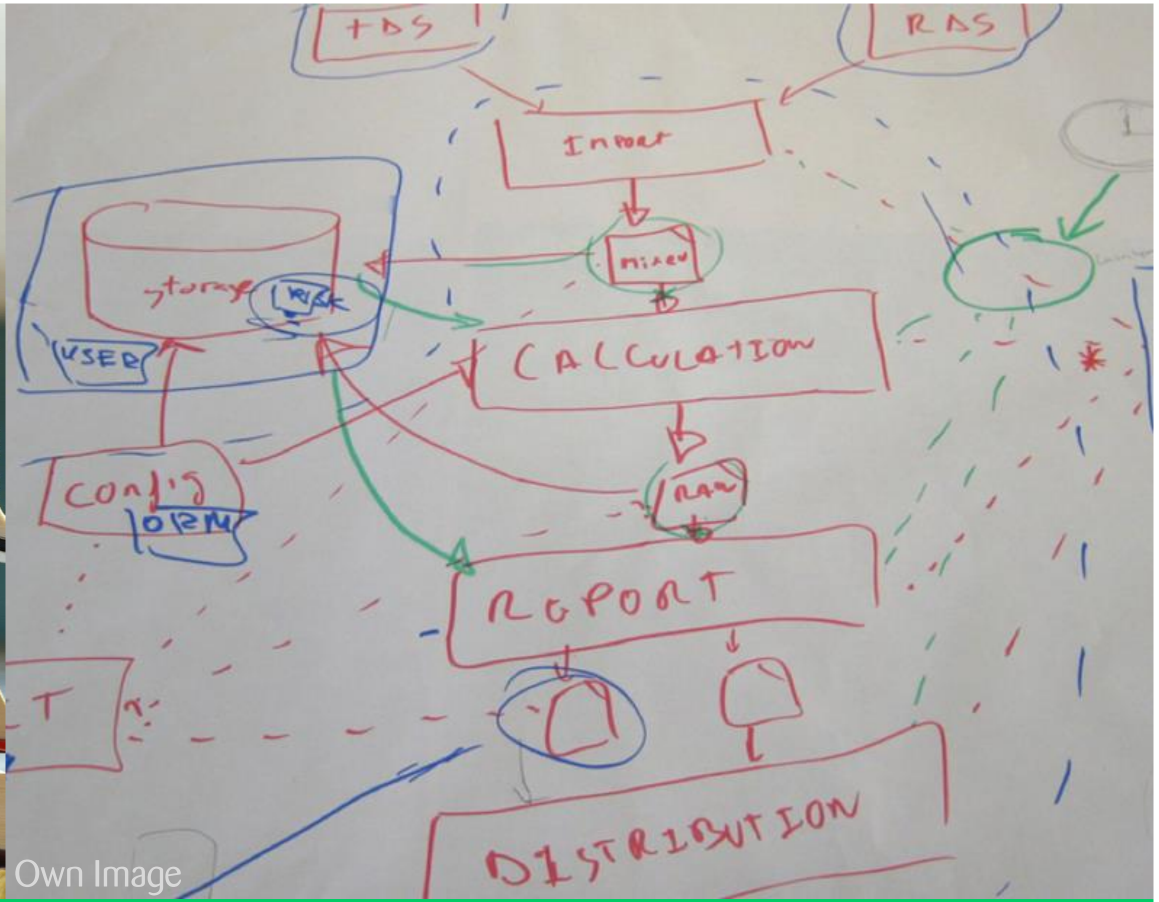
The Addison Wesley Signature Series
A Martin Fowler Signature Book
Continuous Delivery
Reliable Software Releases through Build, Test, and Deployment Automation
Jez Humble
David Farley
Foreword by Martin Fowler

Structurizr
Visualise, document and explore your software architecture with the C4 model



Robert C. Martin Series
Prentice Hall
Clean Code
A Handbook of Agile Software Craftsmanship
Foreword by James O. Coplien
Robert C. Martin


CC0 – no attribution



Own Image

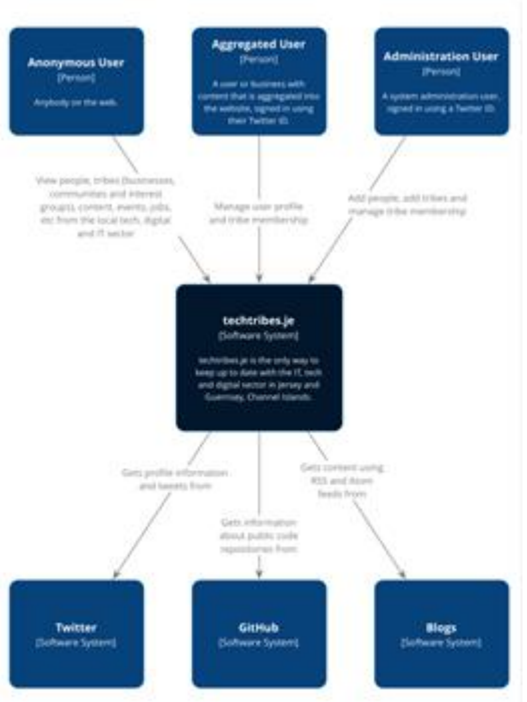
Unreadable Architecture

pharmacy Mareshki
Аптека Марешки
Britakar
Бритакар
"Nikola Gabrovski"
Zuhlke Engineering EOOD
Tara Soft
Watch Store
Ремонт, реставрация и продажба на...
Twins Center
Pod Voda
Под вода
ul. "Baku"

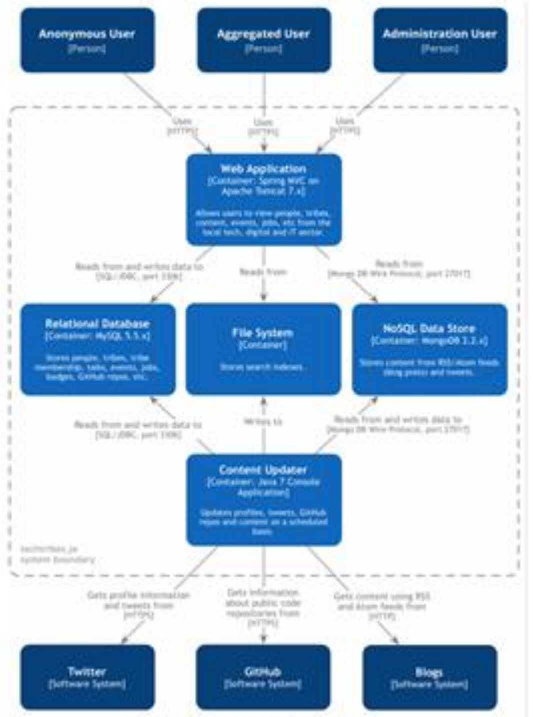
Architecture diagrams should be maps

Simon Browns C4 Architecture Model

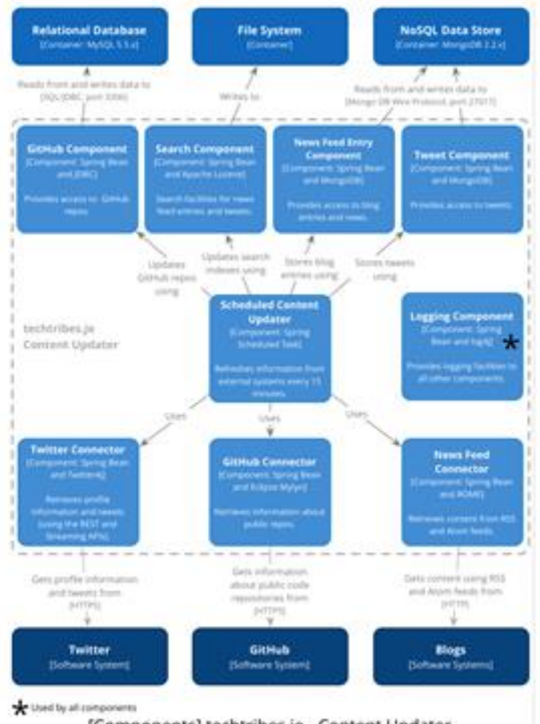
"Diagrams are maps that help you **navigating**"


Anonymous User
[Person]
Anybody on the web.
Aggregated User
[Person]
Administration User
[Person]
View people, tribes (businesses, communities and interest groups), content, events, jobs, etc from the local tech, digital and IT sector
Manage user profile and tribe membership
Add people, add tribes and manage tribe membership
techtribes.je
[Software System]
Gets profile information and tweets from
Gets information about public code repositories from
Gets content using RSS and Atom feeds from
Twitter
[Software System]
GitHub
[Software System]
Blogs
[Software System]

[System Context] techtribes.je


Anonymous User
[Person]
Aggregated User
[Person]
Administration User
[Person]
Uses [HTTPS]
Web Application
[Container: Spring MVC on Apache Tomcat 7.x]
Reads from and writes data to [SQL/JDBC, port 3306]
Reads from
Reads from [Mongo DB Wire Protocol, port 27017]
Relational Database
[Container: MySQL 5.5.x]
File System
[Container]
Stores search indexes.
NoSQL Data Store
[Container: MongoDB 2.2.x]
Writes to
Content Updater
[Container: Java 7 Console Application]
techtribes.je system boundary
Gets profile information and tweets from [HTTPS]
Gets information about public code repositories from [HTTPS]
Gets content using RSS and Atom feeds from [HTTP]
Twitter
[Software System]
GitHub
[Software System]
Blogs
[Software System]

[Containers] techtribes.je


Relational Database
[Container: MySQL 5.5.x]
File System
[Container]
NoSQL Data Store
[Container: MongoDB 2.2.x]
Reads from and writes data to
Writes to
GitHub Component
Search Component
News Feed Entry Component
Tweet Component
Updates GitHub repos using
Updates search indexes using
Stores blog entries using
Stores tweets using
techtribes.je Content Updater
Scheduled Content Updater
Logging Component
*
Uses
Twitter Connector
GitHub Connector
News Feed Connector
Gets profile information and tweets from [HTTPS]
Gets information about public code repositories from [HTTPS]
Gets content using RSS and Atom feeds from [HTTP]
Twitter
[Software System]
GitHub
[Software System]
Blogs
[Software Systems]
* Used by all components

[Components] techtribes.je - Content Updater

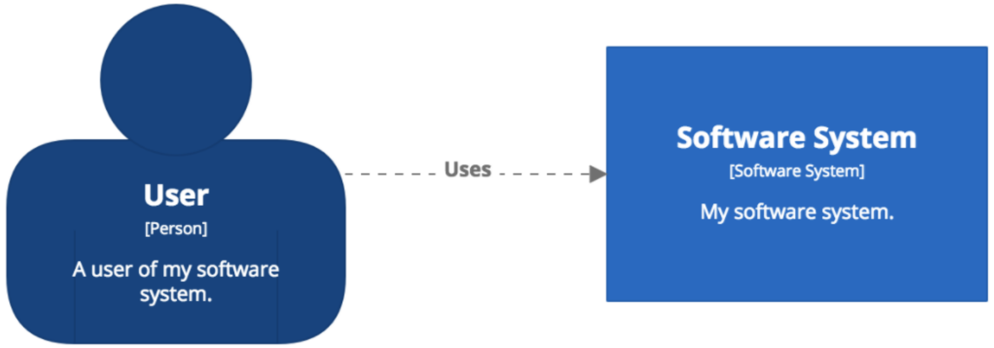
Structurizr

Create software architecture models based upon the C4 model using code

```
Workspace workspace = new Workspace("Getting Started", "This is a model of my software system.");
Model model = workspace.getModel();

Person user = model.addPerson("User", "A user of my software system.");
SoftwareSystem softwareSystem = model.addSoftwareSystem("Software System", "My software system.");
user.uses(softwareSystem, "Uses");

ViewSet views = workspace.getViews();
SystemContextView contextView = views.createSystemContextView(softwareSystem, "SystemContext", "An example of a
contextView.addAllSoftwareSystems();
contextView.addAllPeople();
```



Keep track of where you have been going, and why –

so you don't have to blindly trust or change prior decisions

In the context of **<use case/user story u>**,

facing **<concern c>** we decided for **<option o>** and neglected **<other options>**,

to achieve **<system qualities/desired consequences>**,

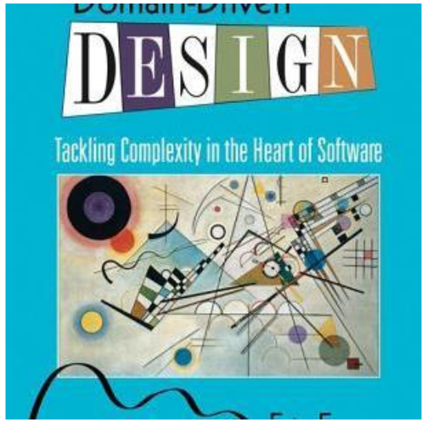
accepting **<downside/undesired consequences>**,

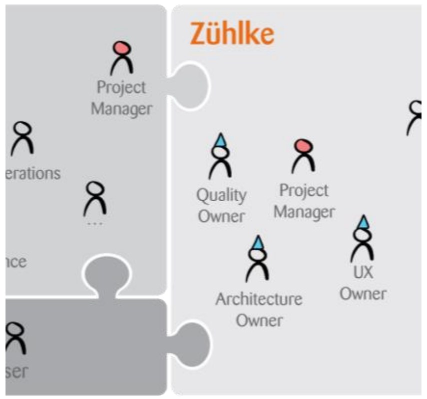
because **<additional rationale>**.

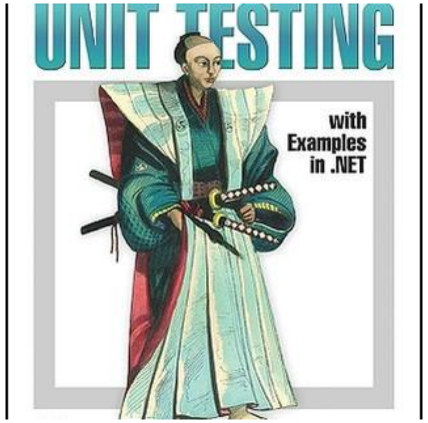
Architectural Decision Record

Books and sources

Books and sources







Specification By Example
Gojko Adzic

Domain Driven Design

Eric Evans

PACT

docs.pact.io

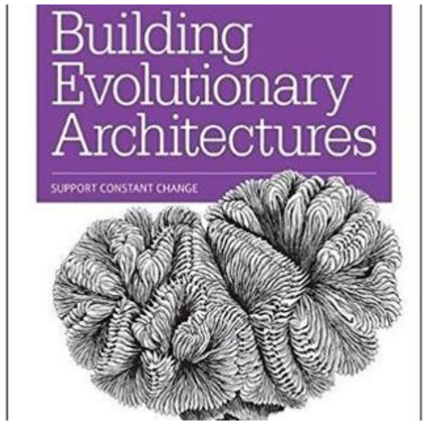
ADM Models

Alex Bögli, Christian Straube, Christian Heger and many more
Zühlke Engineering

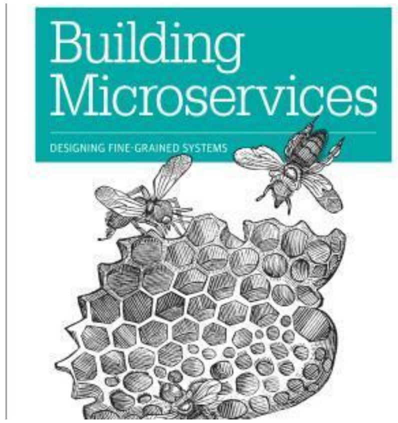
The Art of Unit Testing

Roy Osherove

Books and sources

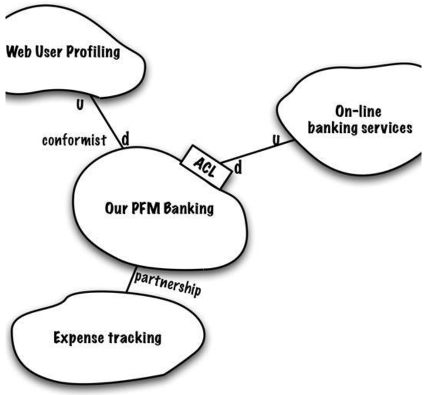


Building Evolutionary Architectures
Neal Ford, Rebecca Parsons, Patrick Kua



Building Microservies

Sam Newman



Context Mapping

Alberto Brandolini

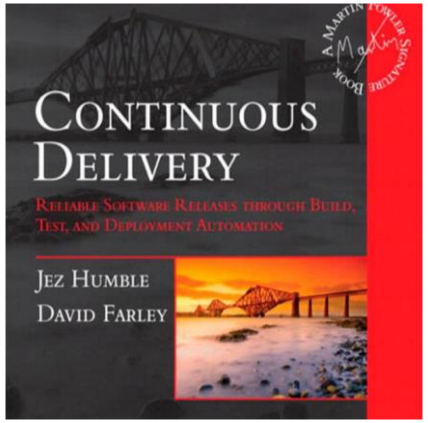
https://www.infoq.com/articles/ddd-contextmapping/

C4 diagrams Structurizr
Simon Brown
structurizr.com

Selenium

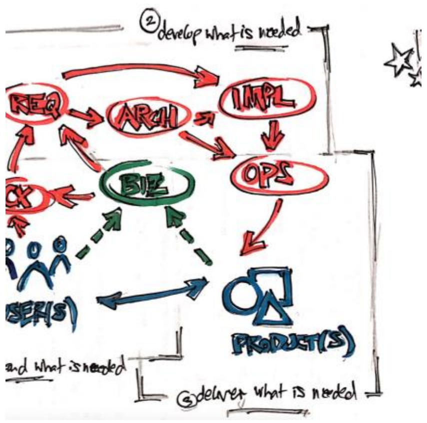
seleniumhq.org

Books and sources


A Martin Fowler Signature Book
Continuous Delivery
Reliable Software Releases through Build, Test, and Deployment Automation
Jez Humble
David Farley

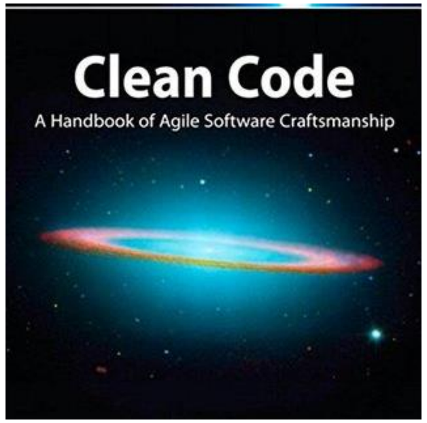
Continuous Delivery

Jez Humble, David Farley



Discipline Flow

Stephan Janisch
Zühlke Engineering

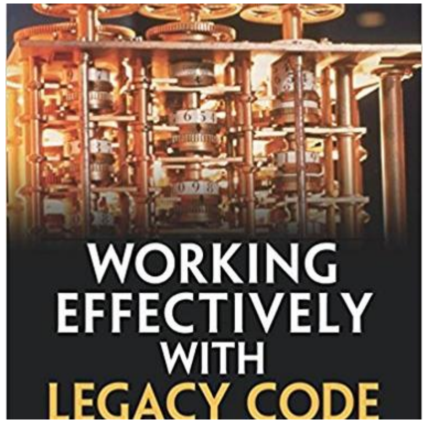


Clean Code

Robert “Uncle Bob” C. Martin

Gherkin DSL

cucumber.io
specflow.org



Working Effectively with Legacy Code
Michael C. Feathers

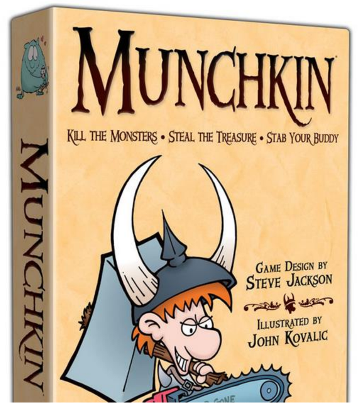
Books and sources

zühlke
empowering ideas

adr.github.io

```
KeyVault = new MonkeyKeyVault(
Hub = new MonkeyHub(this, envi
CrmConnector = new MonkeyCrmCo
EventStore = new MonkeyEventSt
UI = new MonkeyUI(this, EventS
MessageProcessor = new MonkeyM
DPS = new MonkeyDeviceProvisio

TechnicalSupportUser = workspa
```


MUNCHKIN
KILL THE MONSTERS • STEAL THE TREASURE • STAB YOUR BUDDY
GAME DESIGN BY STEVE JACKSON
ILLUSTRATED BY JOHN KOVALIC
MUNCHKIN

Architectural Decision Record tooling

adr.github.io

Architectural Decision Record

Michael Nygard

http://thinkrelevance.com/team/members/michael-nygard

Christian Eder

Structurizr / Infrastructure as Code

https://github.com/ChristianEder/Structurizr.InfrastructureAsCode

Munchkin

Fantasy Card Game

www.worldofmunchkin.com