Towards Autonomous Aligned Teams with Domain-Driven Design
Instructie dia: YouTube links, open deze in de diavoorstelling en klik op de gewenste video link.

1. Wanneer nodig, verwijder de bestaande afbeelding d.m.v. de 'Delete'-knop. Klik op het pictogram om een nieuwe afbeelding in te voegen. (zie onderstaand voorbeeld).

2. Selecteer de gewenste afbeelding en klik op 'Invoegen'.

3. Klik met de rechter muisknop op de afbeelding en kies 'Naar achtergrond'.
Enterprise Application “Spaghetti”
Any organization that designs a system (defined more broadly here than just information systems) will inevitably produce a design whose structure is a copy of the organization's communication structure.

— Mel Conway
Buildings system

Buildings
7 people

Wires System

Wires
30 people

Rods system

Roads
16 people
Siloed functional teams... ... lead to siloed application architectures. Because Conway's Law

https://www.martinfowler.com/articles/microservices.html
[In our study at Thoughtworks we found] work takes an order of magnitude longer when it leaves a team.

— James Lewis (@boicy)

Credit: Nick tune
Cross-functional teams...

...organised around capabilities
Because Conway’s Law

https://www.martinfowler.com/articles/microservices.html
Mortgage company Lowlandisers
Big ball of mud
A monolithic application puts all its functionality into a single process...

... and scales by replicating the monolith on multiple servers

A microservices architecture puts each element of functionality into a separate service...

... and scales by distributing these services across servers, replicating as needed.

https://www.martinfowler.com/articles/microservices.html
Domain-Driven Design
Essence of Domain-Driven Design

➔ Using *models* for creating software
➔ Focus on part of the software handling *complex business requirements*
➔ Focus on a language where we really *crisply concisely describe* the situation in the domain
➔ *Shared language* created through *conversations* between business people (specialists) and software people which becomes *the ubiquitous language*
➔ Instead of *one canonical language*, create *multiple bounded languages*
Context is everything
“We finally have more insight into the software we need to build”
“We already knew everything that was discussed”
REAL ORGANIZATION CHART

Book: Corporate tribe, Braum & Kramer
IT as a factory
I'm... I'm not gonna make it. Tell my wife... I love her...

I promise you, *URK!*

You have to tell Jim's wife... that he loves her...

What do you mean Jim rubs birds?
“It is not the domain experts knowledge that goes to production, it is the assumption of the developers.”

-Alberto Brandolini
Problem Space

Our world as we perceive it
(Sub)Domains
Business Architecture
Independent of Software
Language is fluid
Stable system, strategic changes
The business architecture and the wiki.
It is not documentation that we want,
It is a shared state of mind.
The big picture.....
Inviting the right people together in a room
Go emic....
Finding the pain....
Don’t talk about methods....
Talking complexity around a table
The 6 Trumps
Six Learning Principles that trumps Traditional Teaching

- Movement trumps sitting.
- Talking trumps listening.
- Images trumps words.
- Writing trumps reading.
- Shorter trumps longer.
- Different trumps same.

Brain Science - the 6 trumps by Sharon Bowman - https://www.youtube.com/watch?v=DAiXOgFu8Wc
VISUAL MEETINGS

How to use graphics, sticky notes, & idea mapping to transform group productivity

Think

Context, Action, Vision

Imagine, Values, Results, Actions, Engage, Listen, Talk, See

Enact

Here is what we are creating!

David Sibbet
Collaborative modelling tools
The room setup, right before the workshop

https://www.eventstorming.com/
Lack of Oxygen will kill your workshop
All you need to know about a Domain Event to get you started

- Orange sticky note
- Verb at past tense
- Relevant for domain experts
Mortgage request received
Mortgage application passed
Tender signed
Declaration rejected
Business relevant
Managing uncertainty without influencing.
Group dynamics....
Be careful structuring chaos to fast, you might lose valuable insights.
We already have the customer information, but we make them fill it in. This is not customer-friendly.

We are ping-ponging tasks with another team.
Stories will tell itself, as long as you listen

So slow down....
Events; How instead of why
language is fluid and subject to people’s whims
Holly Brockwell @holly · Jul 20
Wait, WHAT? Americans, can you confirm?
Show this thread

Americans use “quite” to amplify their enthusiasm for the adjective, in the way they would use “really”, “very” or “totally”. By contrast, if a Brit volunteers that a visitor is “quite attractive”, they’re only saying “fairly” or “sort of attractive”. This ambiguous intensifier bestows a “damning with faint praise” effect on any word with which it is teamed.

I NEVER KNEW THIS
I NEVER KNEW THAT WAS WHAT AMERICANS MEANT WHEN THEY SAID “QUITE”
WHY DIDN’T ANYBODY TELL ME
SUDDENLY THAT ONE SONG THAT GOES “HELLO I MISS YOU QUITE TERRIBLY” MAKES LIKE A MILLION TIMES MORE SENSE

are you serious british people
Check out...
You had to be there.....
Everything has been said before; but since nobody listens, we must always start again.
Emerging bounded contexts after a Big Picture EventStorming.

https://www.eventstorming.com/
Swimlanes are usually a reliable clue for possible different bounded contexts.
Cross-functional teams... ...organised around capabilities
Because Conway's Law

https://www.martinfowler.com/articles/microservices.html
Single source of truth....
A **model** is a *simplified representation* of a thing or phenomenon that intentionally emphasizes *certain aspects* while ignoring others. **Abstraction** with a *specific use* in mind.

- Rebecca Wirfs-Brock
**Problem Space**
Our world as we perceive it
(Sub)Domains
Business Architecture
Independent of Software
Language is fluid
Stable system, strategic changes

**Solution Space**
Our world as we designed it
Bounded Context
Software Architecture
Linked to the business language is consistent
Constant changes, depending on insights
“Architectural design is system design. System design is contextual design — it is inherently about boundaries (what’s in, and what’s out, what spans, what moves between), and about tradeoffs.”

—Ruth Malan
1. **Numerieke bullet**

Meer weten? Ga naar dia 'Instructie dia: You Tube links', open deze in de diavoorstelling en klik op de gewenste video link.

---

**HOE WERKEN TEKSTNIVEAUS?**

1. Start Tekstniveau omhoog
2. Tekstniveau omlaag
3. Bullet
4. Leestekst
5. Subtitel

Gebruik, onder de tab 'Start', de lijstniveau-knoppen, zoals hierboven aangegeven, om een tekst niveau te kiezen. Kies uit:

---

**Bounded Context**

- Unit of Consistent Language
- A model for a specific purpose
Software Architect
Socio-technical Architect

-Nick tune
If we have a system of improvement that is directed at improving the parts taken separately. You can be absolutely sure that the improvement of the whole will not be improved.
To *achieve autonomous alignment*, we need to *stop thinking* in terms of business and IT and start acting as *an integrated team*.
To achieve autonomous alignment, we need to form a culture in interactions, in conscious dialogue and in decisions, by having rituals and marketplaces.
To achieve autonomous alignment, we need to have an open modern culture with a leadership to match.
Hire an anthropologist
Want to know more?

**Wednesday**, August 7 • 14:00 - 15:15

EventStorming: Continuous discovery between multiple disciplines (Kenny Baas)

**Thursday**, August 8 • 14:00 - 15:15

Challenge Everything! Using EventStorming to challenge your business and architecture too! (Alberto Brandolini)

Experience Report
https://www.agilealliance.org/resources/experience-reports/towards-autonomous-aligned-teams-with-domain-driven-design/