

EventStorming für Domain-Driven Design



NICOLE RAUCH
software development &
development coaching

Warum Domain-Driven Design?

DDD in a Nutshell

- ▶ Gemeinsames Verständnis schaffen
- ▶ Trennung Business Logik ↔ Technik
- ▶ Strukturierung des Codes

Gemeinsames Verständnis schaffen

- ▶ Hat hohen Stellenwert
- ▶ „Knowledge Crunching“

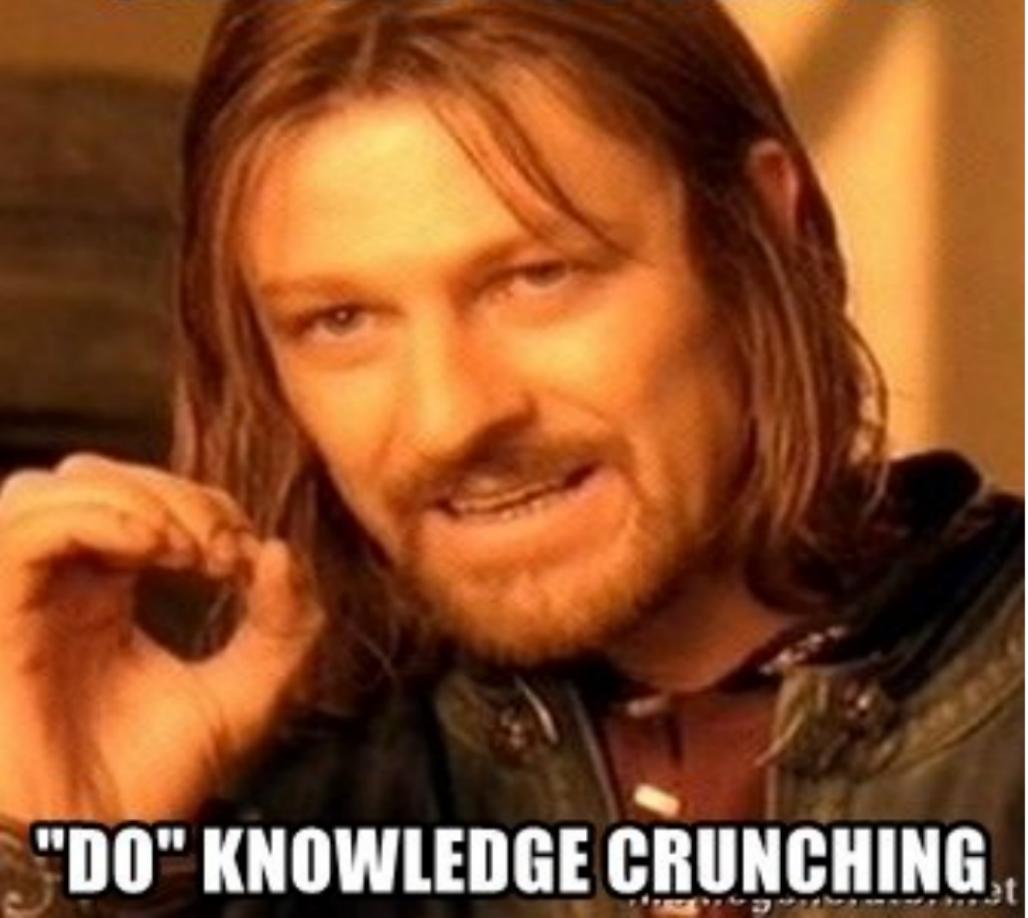
DO KNOWLEDGE CRUNCHING



Spongebob Squarepants vector trace by kssael. ©Nickelodeon

AND ALL WILL BE WELL

ONE DOES NOT SIMPLY



"DO" KNOWLEDGE CRUNCHING,







Gemeinsames Verständnis schaffen - Warum?

- ▶ Gedanken sichtbar und “begreifbar” machen
- ▶ Modell schafft Klarheit
- ▶ Ubiquitous Language grenzt Begriffe ab



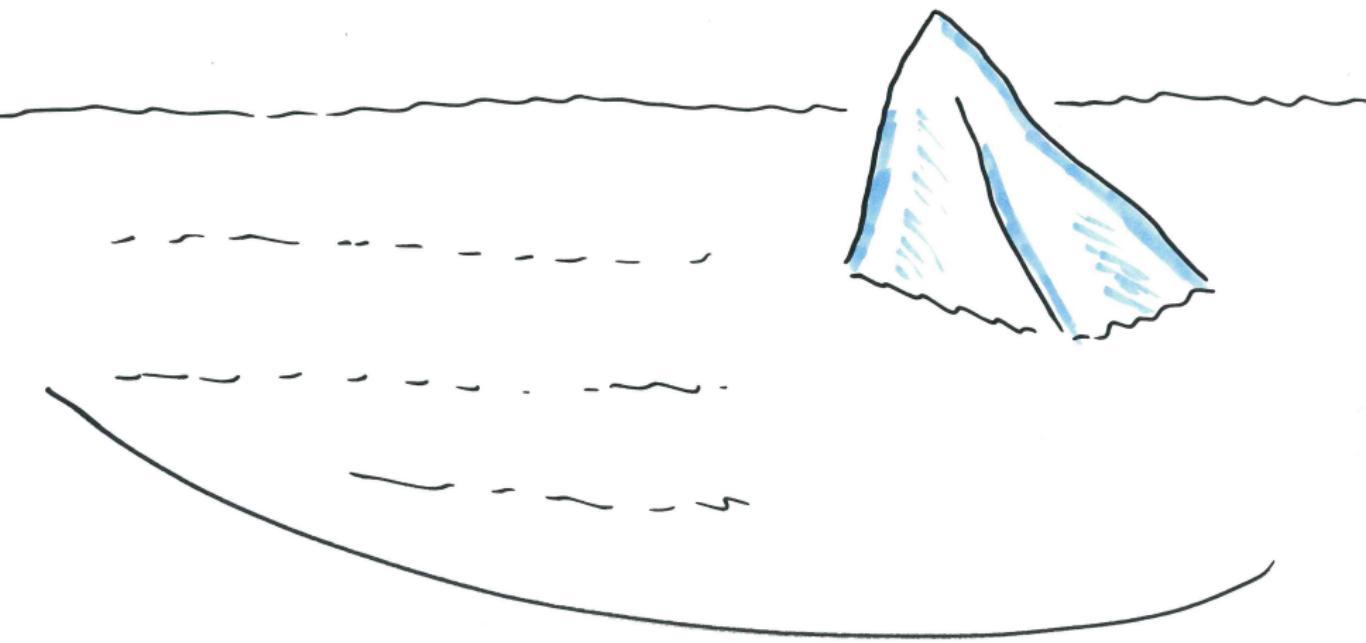
Strand ...
Meer ...
Sonne ...











WARNING

Hier bei uns passiert gerade dasselbe!

WORKSHOP

Unsere Domäne: eCommerce

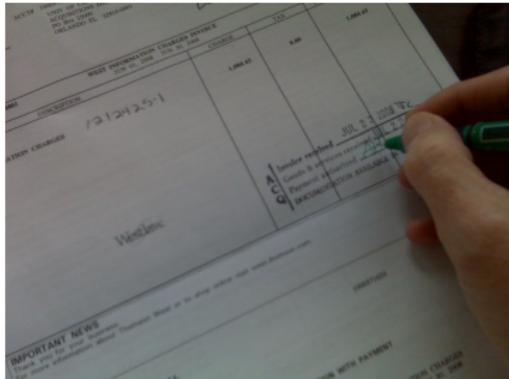
Unsere Domäne: eCommerce



Unsere Domäne: eCommerce



Unsere Domäne: eCommerce



EventStorming I

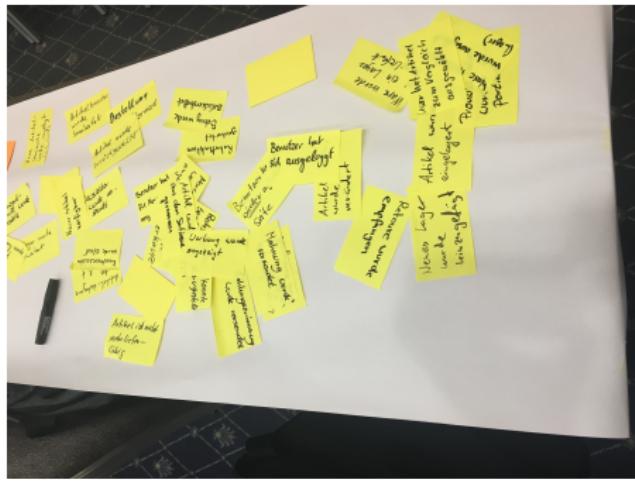
Events erfassen

- ▶ Ereignis in der Vergangenheit
- ▶ Relevant für den Fachbereich
- ▶ Beobachtbar im System
- ▶ Beschrieben durch **Verb in der Vergangenheit**

Beispiel

Artikel wurde
in Warenkorb
gelegt

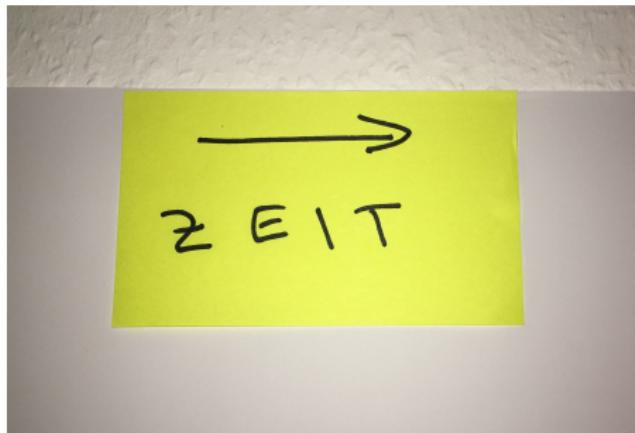
So kann das aussehen



EventStorming II

- ▶ Viele Events unterliegen einer zeitlichen Abfolge
- ▶ Zeit verläuft **von links nach rechts**

Zeit



Beispiel

Artikel
wurde
angezeigt

Artikel wurde
in Warenkorb
gelegt

Warenkorb
wurde
bezahlt

Bestellung
wurde versandt

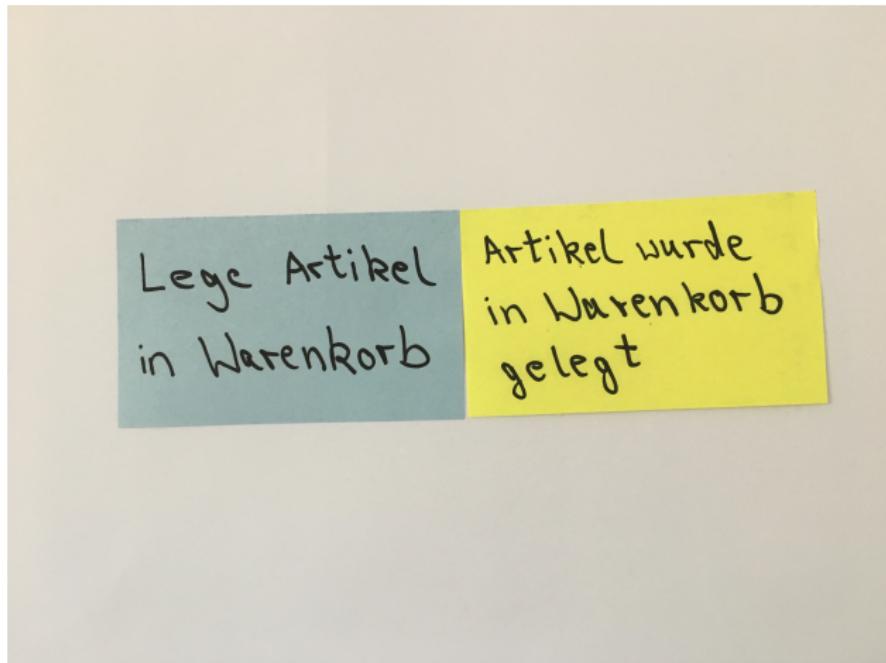
So kann das aussehen



EventStorming III

- ▶ Vor einem Event muss etwas im System passiert sein:
- ▶ **Command**
- ▶ Werden in Befehlsform ausgedrückt

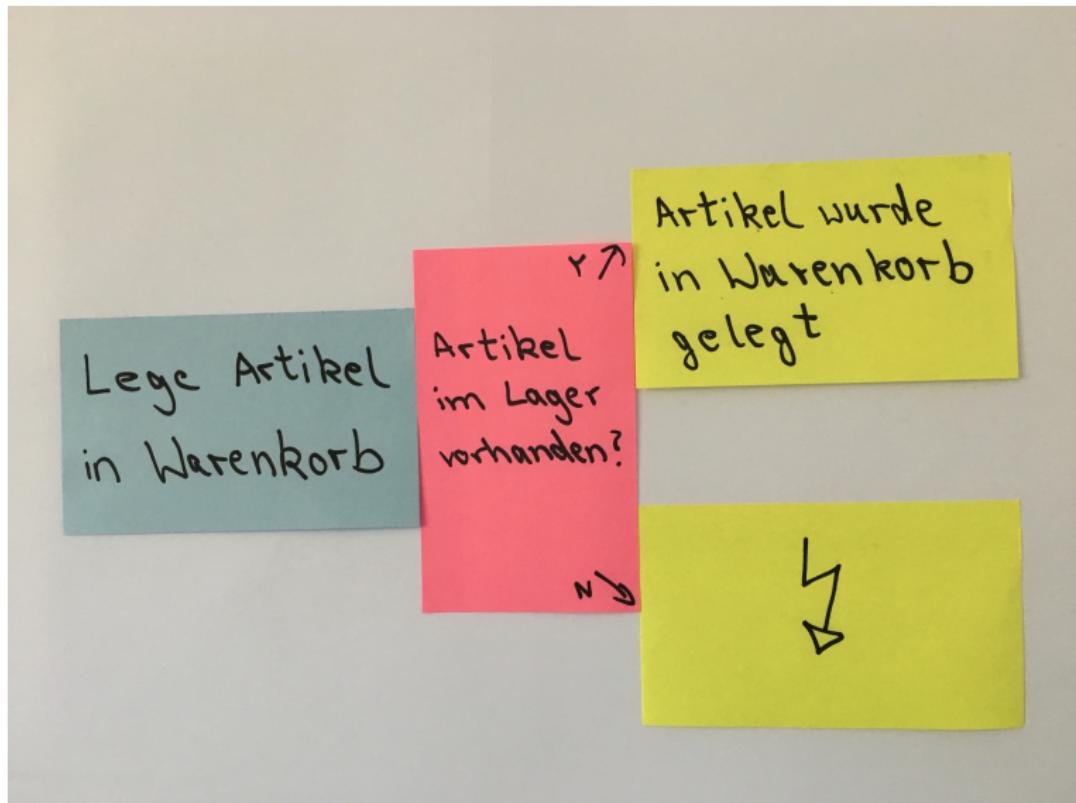
Beispiel



EventStorming III

- ▶ Nicht alle Commands führen direkt zu einem Event
- ▶ **Constraints** müssen berücksichtigt werden
- ▶ Entscheidung als Frage formulieren
- ▶ Events beantworten die Frage

Beispiel

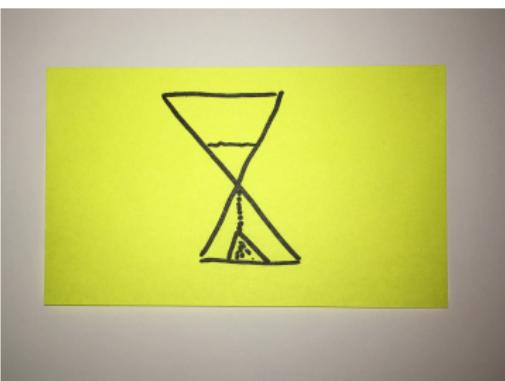


So kann das aussehen

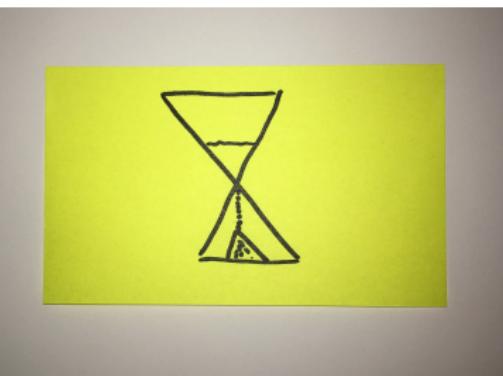


Darstellung von Zeit

Darstellung von Zeit



Darstellung von Zeit



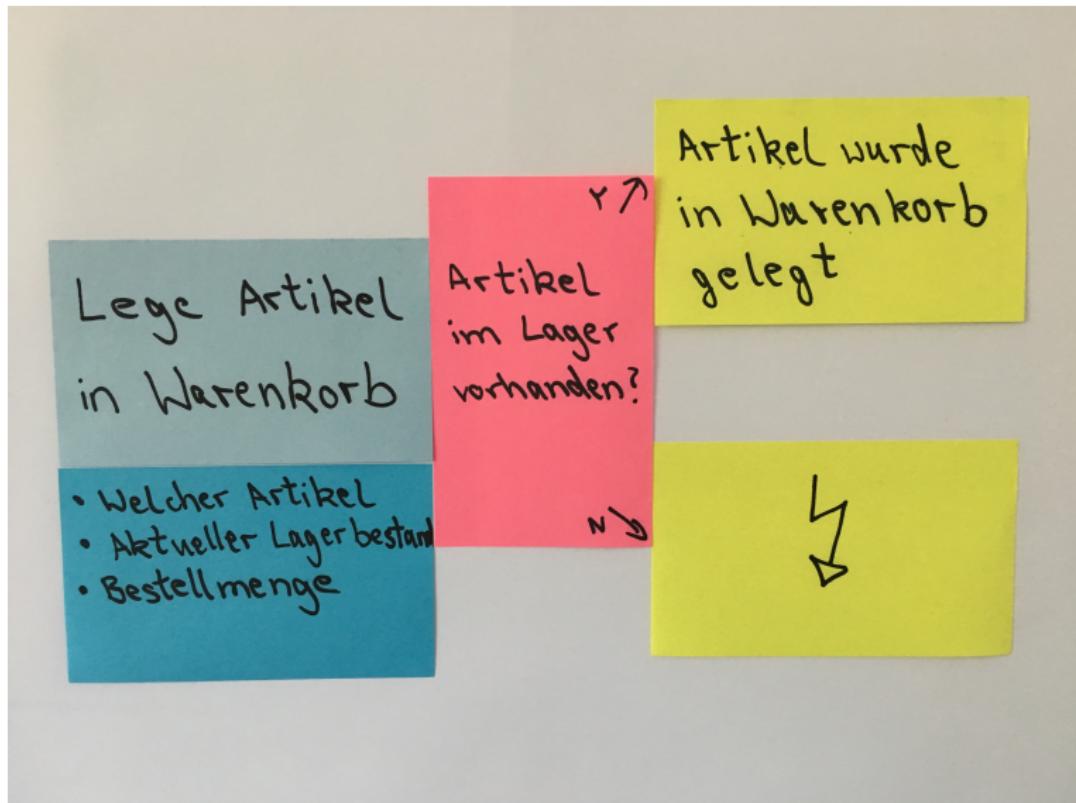
Darstellung von Wiederholung



EventStorming IV

- ▶ Constraints benötigen **Daten** zur Entscheidung
- ▶ Müssen verfügbar sein
 - ▶ Vorherige Erfassung in Event
 - ▶ Aus anderem Systemteil
- ▶ Fehlen sie, muss das Modell ergänzt werden

Beispiel



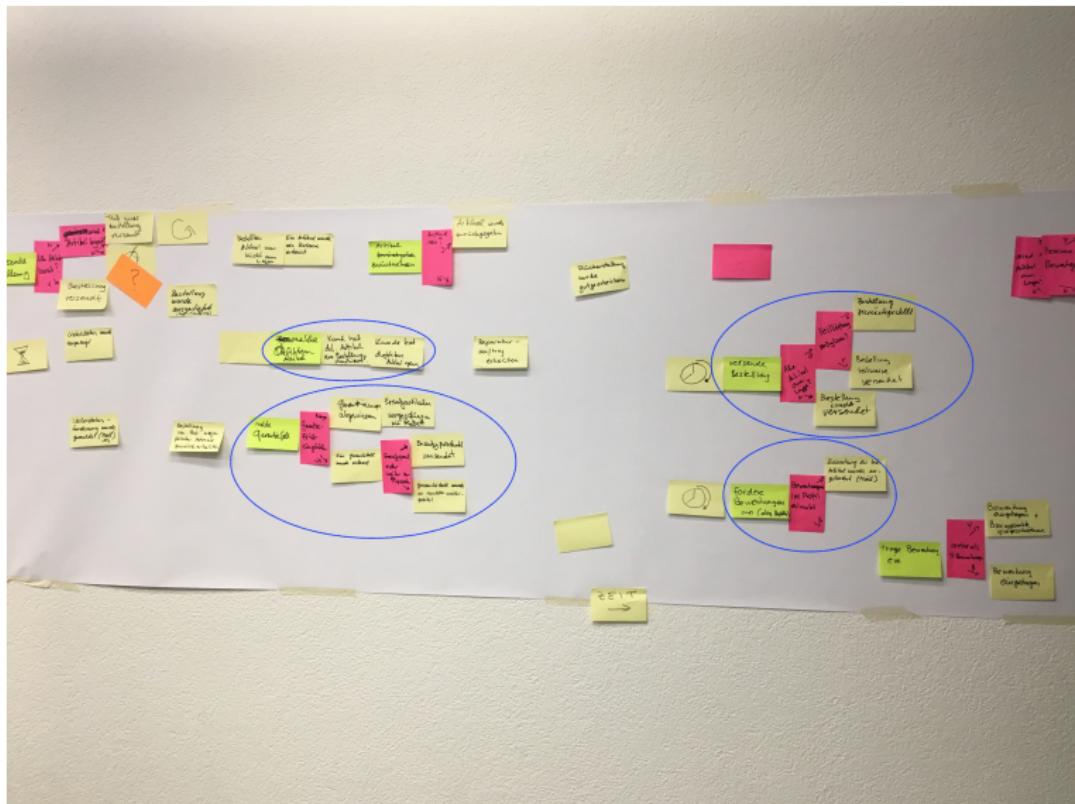
EventStorming V

- ▶ **Bounded Contexts**
 - ▶ hohe Kopplung **innerhalb**
 - ▶ Wenige Abhängigkeiten **dazwischen**
- ▶ Abhängigkeiten ggf. mit Wollfäden symbolisieren

EventStorming VI

- ▶ **Entwicklungs pakete** (User Stories) ableiten
- ▶ Command mit Events ⇒ Story
 - ▶ Ggf. aufteilen in Happy Path + weitere Fälle

Beispiel



Wie geht es weiter?

Umsetzung einer Story:

- ▶ Zuerst sehr detailliert modellieren, dann sofort implementieren
- ▶ Modell an die Wand hängen während der Entwicklung
- ▶ Alle Diskussionen an und mit diesem Modell durchführen
- ▶ Bei Änderungen Modell und Code aktualisieren!
- ▶ Event Sourced Implementierung ist einfach realisierbar

Zusammenfassung

Warum Modellieren?

- ▶ Modellieren dient dazu, Gedanken sichtbar und „begreifbar“ zu machen

Warum Modellieren?

- ▶ Modellieren dient dazu, Gedanken sichtbar und „begreifbar“ zu machen
- ▶ Jeder entwickelt eigene Vorstellungen von etwas Gehörtem

Warum Modellieren?

- ▶ Modellieren dient dazu, Gedanken sichtbar und „begreifbar“ zu machen
- ▶ Jeder entwickelt eigene Vorstellungen von etwas Gehörtem
- ▶ Ein Modell versucht Klarheit zu schaffen
- ▶ EventStorming ist ein Weg, sehr schnell zu einem sehr detaillierten Modell zu kommen

Kritische Fragen stellen

- ▶ Details mit dem Fachbereich abklären
- ▶ Das Modell kritisch hinterfragen!
- ▶ Modellierungs-Alternativen diskutieren!
- ▶ „Was wenn dies vor jenem passiert?“
- ▶ Die üblichen Verdächtigen („immer“, „nie“, „kann nicht sein“ . . .) verfolgen und entkräften

Einsichten in die Domäne gewinnen

- ▶ Kollaboration zwischen Fachbereich und Entwicklern ist der entscheidende Faktor für den Erfolg eines Projekts
- ▶ Man braucht die Domänen-Experten des betreffenden Bereiches
- ▶ EventStorming ist auch ohne DDD einsetzbar

Vielen Dank!

E-Mail info@nicole-rauch.de

Twitter @NicoleRauch

Web <http://www.nicole-rauch.de>



NICOLE RAUCH
software development &
development coaching

EventStorming · Domain-Driven Design
Training · Coaching · Facilitation
Software Craftsmanship
React.js and Redux
Functional Programming

<https://github.com/NicoleRauch/EventStormingForDDD>

Credits

Palm Beach: Grand Velas Riviera Maya - Beach Palm Trees Riviera Maya

<https://www.flickr.com/photos/grandvelasrivieramaya/3179390917>

Crowded Beach: Deyvis Tejada - Cala Mondrago - Cala Esmeralda
Mallorca

<https://www.flickr.com/photos/143430050@N02/33639336432>

Japan Beach: Aleksander Dragnes - Miyajima Floating Torii

<https://www.flickr.com/photos/adragnes/644727155>

Ice Beach: Kitty Terwolbeck - Colorful waters (Spitsbergen 2012)

<https://www.flickr.com/photos/kittysfotos/7902630982>

Onlineshopping: Elaine Smith - Online Shopping

<https://www.flickr.com/photos/155416046@N05/35933760125>

Warehouse: Gwan Kho - Mediq Sverige Kungsbacka warehouse

<https://www.flickr.com/photos/gwankho/6205837092>

Ordering: cybrgrl - Invoices

<https://www.flickr.com/photos/cybrgrl/2693815369>