



Domänen Driven Design

Service Engineering



Inhalt

- Motivation und Ziele
- Aufgaben zur Übung
- Hinweise zum Domain Driven Design (kurz DDD)
- DDD und Web-APIs
- Bezüge zu existenten Domänenmodellen – (hier TMF)

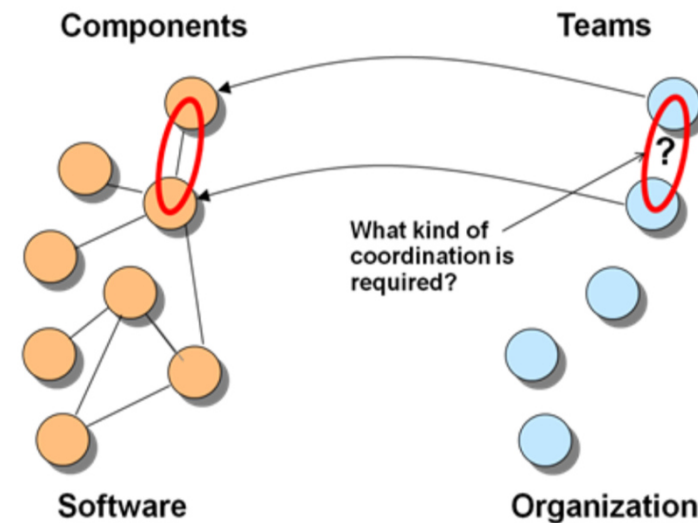


Motivation und Ziele

Conway's Law,

“Any organization that designs a system will inevitably produce a design whose structure is a copy of the organization’s communication structure”

[Conway 1968]



Quelle: Conway, M. E. “How Do Committees Invent?” Datamation 14, 4 (April 1968): 28–31.,
Genutzte Sekundärquelle: Bass, L. et al.: A Workshop on Architecture Competence, S. 15 | CMU/SEI-2009-TN-005

Hintergründe zum DDD

"Für die meisten Softwareprojekte sollte der Fokus auf der jeweiligen Domäne und Domänenlogik liegen. Und das Design komplexer Domänen sollte auf Modellen aufbauen."

Diese Kernaussage von DDD kann man für die meisten Domänen noch etwas schärfer formulieren: *"Wenn man ein Problem nicht ohne Software lösen kann, kann man es mit Software (meist) auch nicht."*

Quelle: Martincevic, NDDD: Context is King – Kein Context, keine Microservices,
<https://www.informatik-aktuell.de/entwicklung/methoden/ddd-context-is-king-kein-context-keine-microservices.html>



Ziele der Übung

- Idee des Domänen Driven Design (kurz DDD) erschließen.
- Möglichkeiten des DDD im Bezug auf Web-APIs (REST-APIs).
- Bewertung des TMF-Frameworks hinsichtlich der DDD-Anforderungen.
- Bedarf einer fachlich orientierten Herleitung von Web-APIs erkennen.

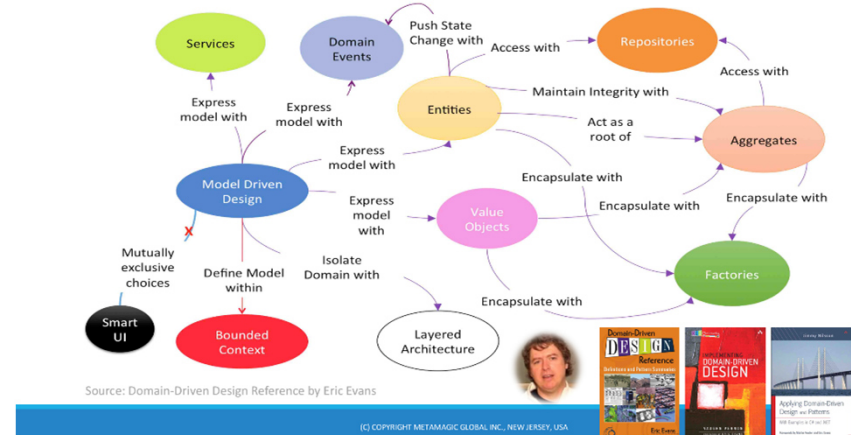


Aufgaben der Übung

Aufgaben 1

- Machen Sie sich mit der grundlegenden Idee des DDD vertraut.
- Was ist unter dem Begriff des „bounded context“ zu verstehen?
- Was wird unter dem Begriff der Fachsprache (ubiquitous language) im DDD verstanden?
- Warum sollte die Modularität von Web-APIs aus fachlichen Anforderungen erwachsen?

Domain Driven Design

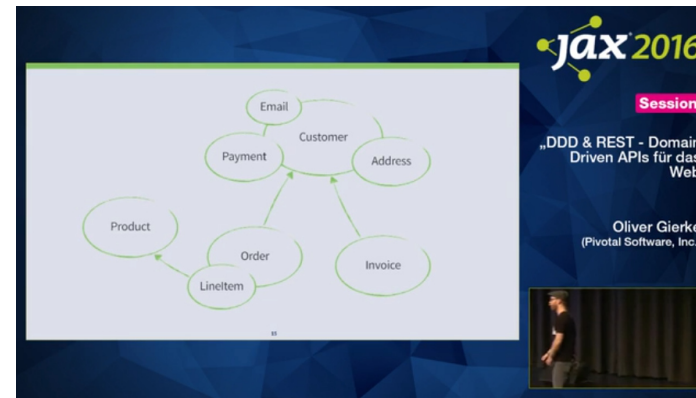


Quelle: <https://github.com/ketan-gote/ddd-example>

Originalquelle: Eric Evans, Domänen Driven Design

Aufgaben 2

- Schauen Sie sich das Video (ca. 65 min.) „DDD & REST – Domain Driven APIs für das Web“ von von Oliver Gierke auf der W-JAX 2016 an.
- Geben Sie die aus ihrer Sicht wichtigsten Kernaussagen (mind. 10) dieses Vortrags stichpunktartig wieder!
- Aus welchen Gründen hat sich der DDD-Ansatz in der Praxis nur bedingt durchsetzen können? Gehen Sie auf Vor- und Nachteile ein!



Oliver Gierke: DDD & REST – Domain Driven APIs für das Web, <https://jaxenter.de/ddd-rest-45713>, 30. August 2016

Was ist Domain-Driven Design (DDD)?

Domain-Driven Design (DDD) gibt es zwar bereits seit 2004, doch hat sich das Vorgehen in all der Zeit nicht übermäßig weit verbreiten können. Seit einigen Jahren erlebt der Begriff jedoch einen zweiten Frühling. Daher ist es an der Zeit zu fragen, was Domain-Driven Design eigentlich ist.

Lesezeit: 11 Min. In Pocket speichern 49

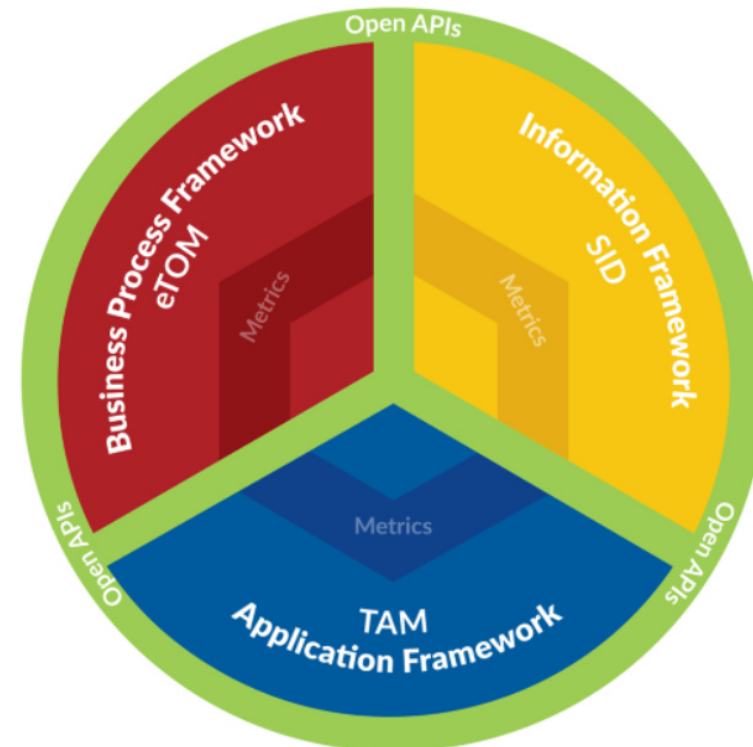
14.12.2020 10:13 Uhr | Developer

Von Golo Roden

<https://www.heise.de/blog/Was-ist-Domain-Driven-Design-DDD-4983527.html>, 14. Dezember 2020

Aufgaben 3

- Welche Bezüge und Problembereiche des DDD sehen Sie im Zusammenhang mit der Idee des TMF-Framework (eTOM, SID und TAM)?
- Identifizieren Sie potentielle Bezüge zu den Open-API Spezifikationen des TMF bzw. in welcher Weise sollte das DDD bei diesen Services berücksichtigt sein?



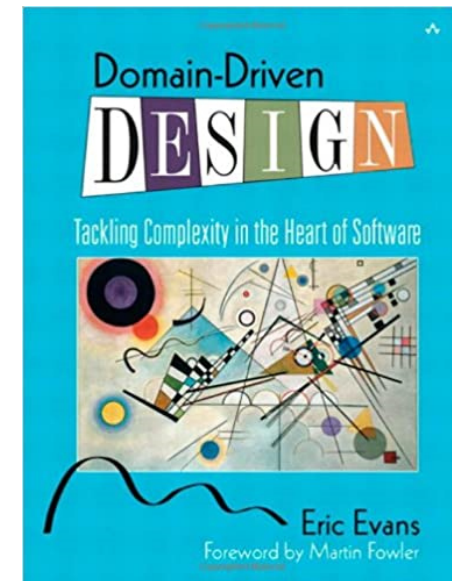
Quelle der rechten Abbildung: Framework 19.5 - <https://www.tmforum.org/framework-homepage>, interaktive Modellerläuterung: http://casewise.tmforum.org/evolve/statics/framework/index.html?_ga=2.6214451.1210546848.1585908342-126195025.1585908342



Hinweise zum DDD

Aspekte im DDD

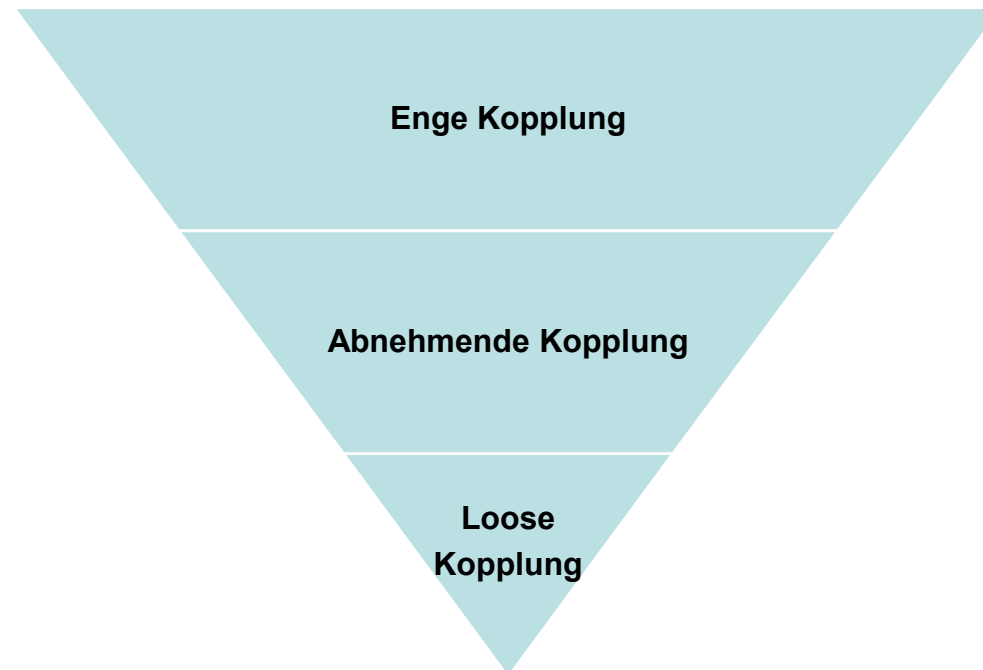
- Vermeidung monolithischer Ansätze.
- Aufteilung bzw. Schneidung komplexer Systeme.
- Bilden autonomer bzw. entkoppelter Einheiten.
- Fokus auf Modularität und Agilität.
- Fachliche Domäne als Treiber von z.B. Microservices.
- Analyse des Problem- (Fachlich) und Lösungsraums (Technisch).
- Modellbasierte Analyse orientiert an Architekturmustern.



Quelle der Abbildung: Evans, E.: Domainen-Driven-Design: Tackling Complexity in the heart of Software, Addison-Wesley, 2004,
Blick ins Buch: <https://www.amazon.de/Domain-Driven-Design-Tackling-Complexity-Software/dp/0321125215>

Schnittstellen und Context Mapping

- Shared Kernel
- Customer/Supplier
- Published Language
- Open-Host-Service
- Anitcorruption Layer
- Conformist
- Seperate Ways

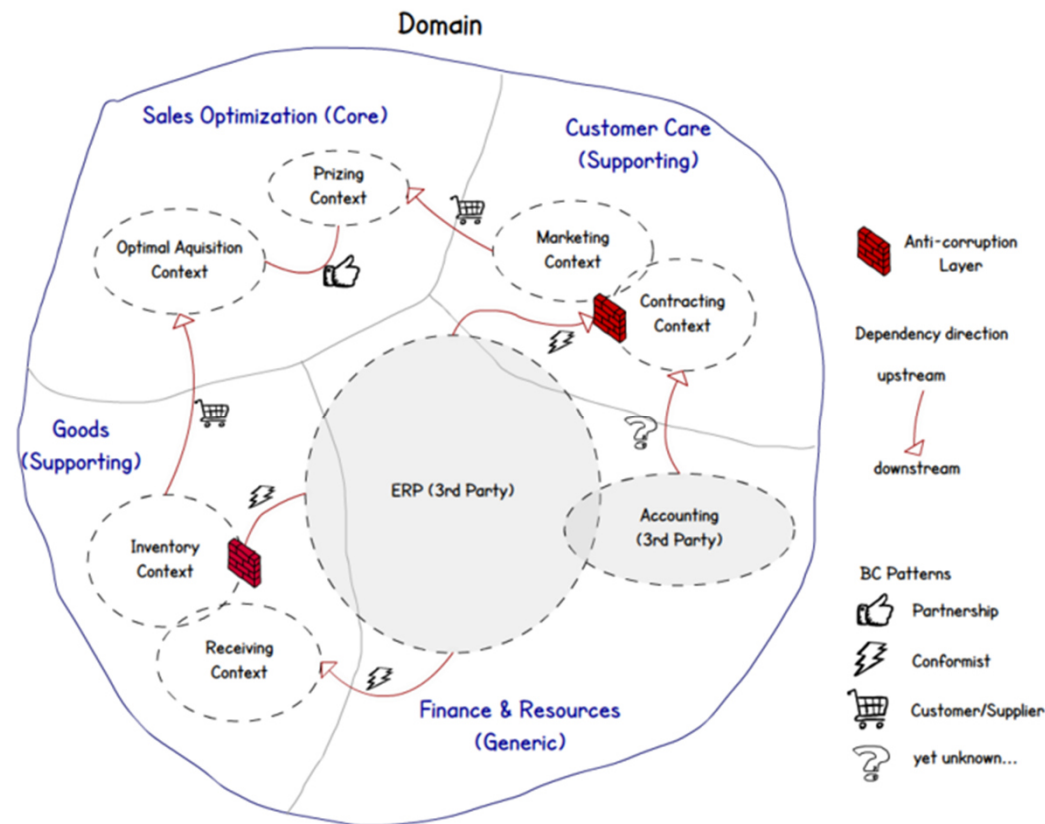


Quelle: Lilienthal, C.: Langlebige Software-Architekturen, S. 103-104, dpunkt.verlag, Heidelberg 2017

Originalquelle: Evans, E.: Domainen-Driven-Design: Tackling Complexity in the heart of Software, Addison-Wesley, 2004

Beispiel einer Context Map

(Quelle: Nino Martincevic)



Quelle der Abbildung: Martincevic, N.: Context is King – Kein Context, keine Microservices,
<https://www.informatik-aktuell.de/entwicklung/methoden/ddd-context-is-king-kein-context-keine-microservices.html>



DDD Beispiel unter GitHub

The screenshot shows the GitHub repository page for 'ketan-gote / ddd-example'. At the top, there are buttons for 'Watch' (1), 'Star' (7), and 'Fork' (7). Below these are tabs for 'Code', 'Issues' (1), 'Pull requests' (0), 'Actions', 'Projects' (0), 'Wiki', 'Security', and 'Insights'. The repository description states: 'Domain Driven Design. Examples focuses on key concept of ddd like Entities, Aggregate root, Repository, Value Objects & ACL.' Below the description are tags: 'domain-driven-design', 'entities', 'aggregate-root', 'repository', 'value-object', 'acl', 'ddd-example', and 'ddd-patterns'. A summary bar shows '3 commits', '1 branch', '0 packages', '0 releases', and '1 contributor'. Below this are buttons for 'Branch: master', 'New pull request', 'Create new file', 'Upload files', 'Find file', and 'Clone or download'. The commit history table shows the latest commit by ketan-gote updating the README.md on 26 Apr 2018. Below the commit history is a file list table.

File	Commit Message	Time
.mvn/wrapper	DDD Examples	2 years ago
order_ddd_example	DDD Examples	2 years ago
.gitignore	DDD Examples	2 years ago
README.md	Update README.md	2 years ago

Quelle der Abbildung: GitHub – Beispiel eines DDD abgebildet mit Java,
<https://github.com/ketan-gote/ddd-example>



Organisation und Quellen

Aufbau des Protokolls

Verwenden Sie bitte das folgende Muster zur Protokollerstellung:

- Allgemeine Angaben
 - Versuch, beteiligte Studenten, Datum
 - Rahmenbedingungen (Hardware, Software, ...)
 - Methodisches Vorgehen zur Bearbeitung
- Aufgaben des Laborversuchs
 - Entsprechend der Struktur der Aufgabenstellungen
 - Textliche Ausführungen zu den Lösungen
 - Verwendung von Grafiken und Screenshots
- Zusammenfassung (Bewertung der erreichten Ergebnisse)
- Exakte Quellenangaben (z.B. Literatur, Internet, ...)

Interessante Quellen zum nachlesen

- Martincevic, N.: Context is King – Kein Context, keine Microservices, <https://www.informatik-aktuell.de/entwicklung/methoden/ddd-context-is-king-kein-context-keine-microservices.html>, 22. November 2016
- Gierke, O.: DDD & REST – Domain Driven APIs für das Web, <https://jaxenter.de/ddd-rest-45713>, 30. August 2016
- Roden, G.: Domain-driven Design erklärt, <https://m.heise.de/developer/artikel/Domain-driven-Design-erklaert-3130720.html?seite=all>, 11. März 2016
- Roden, G.: Was ist Domain-Driven Design (DDD)?, <https://www.heise.de/blog/Was-ist-Domain-Driven-Design-DDD-4983527.html>, 14. Dezember 2020
- Schwentner, H.; Janine Jochum-Frenster, J.: Der interessante Aspekt von DDD ist für mich die direkte Zusammenarbeit mit den Anwendern, <https://entwickler.de/ddd/ddd-tipps-interview>, Mai 2022
- Vernon, V.: Domain-Driven Design kompakt, dpunkt.verlag, <https://dpunkt.de/produkt/domain-driven-design-kompakt/>, 2017