

hackathon@thalia in Münster 2021



The poster is a vertical rectangular graphic with a dark grey background and a blue footer. At the top, a green banner contains a lightbulb icon inside a head silhouette, followed by the text 'ZUSAMMEN KREATIV' in large white letters, and 'gegen den Corona Blues und für kreative Kundenlösungen' in smaller white letters below it. Below this, four white icons are arranged vertically: a group of hands, a box, a trophy, and two clinking glasses. Each icon is followed by a line of white text. The bottom section features a green virus icon, the text 'THALIA HACKATHON 25.06.2021' in large white letters, and 'ONLINE CORONA EDITION' in smaller white letters at the bottom left.

ZUSAMMEN KREATIV
gegen den Corona Blues und für kreative Kundenlösungen

Zusammen wollen wir kreative und auch unkonventionelle Kundenlösungen für spannende Fragestellungen finden

Alle Teilnehmer des Hackthons und des Sommerfests erhalten ein Survival Paket

Die Ergebnisse stellen wir allen Interessierten vor
Die besten Ideen werden mit einem Preis honoriert

Nach dem Hackathon geht es zusammen beim digitalen Sommerfest gegen den Corona Blues weiter

**THALIA
HACKATHON
25.06.2021**

ONLINE CORONA EDITION

Zusammen gegen den Corona Blues und für kreative Kundenlösungen:

Nach einem Jahr Pause haben wir 2021 wieder einen Hackathon veranstaltet, diesmal rein digital. Auch nach vielen Monaten üben und gestalten von neuen Online-Formaten sind diese noch immer spannende Experimente für uns.

Wir haben dieses sehr generische Motto gewählt, um möglichst viele Kolleg:innen anzusprechen und den Raum für innovative Perspektiven weit zu öffnen. Das hat funktioniert:

5 kreative Ideen mit Mehrwert für unsere (internen) Kund:innen konnten durch ihren Pitch überzeugen und wurden an diesem Tag von spontan zusammengefundenen interdisziplinären Teams weiter gedacht.

Nicht nur ITler:innen, auch Expert:innen aus den Fachbereichen waren eingeladen sich zu beteiligen, was durch die unterschiedliche Ausrichtung der Ideen angenehm leicht gemacht wurde. So gab es sowohl eher fachlich als auch eher technisch motivierte Themen zur Auswahl.

5 kreative Ideen

1_Produktentwicklung auf Basis von Kundenfeedback

Worum ging es?

Nicht direkt die offensichtlichste Lösung übernehmen, sondern kreativ sein. Nicht direkt mit der Tür bzw. Lösung ins Haus fallen, sondern sich eingehend mit dem Problem auseinandersetzen. Das Problem verstehen, um die für den Kunden beste Lösung erarbeiten - das war das Ziel dieses Projektes.

Ein spontan zusammengefundenes Team aus verschiedenen Fachbereichen wurde dazu mit einer abgespeckten Variante der Methode [Spark Canvas](#) zunächst auf vorgefiltertes Kundenfeedback losgelassen.



Die App ist super, obwohl ich mir besonders bezüglich der Merkliste mehr Optionen wünschen würde. Ich merke mir viele Bücher mit unterschiedlichen Zwecken. Ich würde mir so etwas wie Ordner/ Pinnwände wünschen, die man benennen kann und in die man die gemerkten Bücher einordnen kann. z.B. eine Liste für Geburtstagsgeschenk für einen Freund, eine für klassische Literatur, eine für einen bestimmten Autor, etc. So wäre es einfacher, Ordnung und Überblick zu behalten.

Abbildung: Eine berücksichtigte Rezension

Aus diesen wurde der Use Case „sich einen Überblick über interessante Inhalte bei Thalia verschaffen“ identifiziert und damit verbundene Challenges des Kunden erarbeitet. In einem zweiten Teil haben wir uns zunächst komplett vom Produkt entfernt, um Inspiration an anderer Stelle zu suchen, z.B. bei der App „[KptnCook](#)“. Welche Probleme löst die App besonders gut? Können wir etwas von ihr lernen? Am Ende wurde als die für uns interessante Speciality die Begrenzung der Auswahl, bei KptnCook auf drei Rezepte pro Tag, gewählt.

Ergebnisse

Use Case, Challenges, Inspiration und Speciality wurden am Ende zusammengeführt zu dem Spark „Eine Startseite mit nur drei Büchern“:

Hey Max,
welche Genre
interessieren Dich?

- Krimi
- Romane
- Fantasy

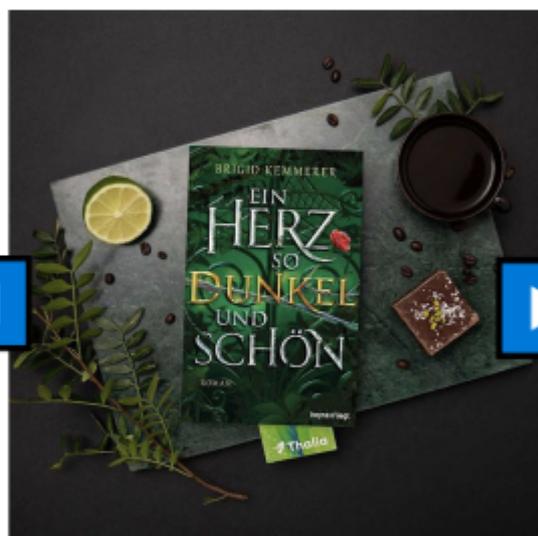
Überspringen

Danke!
Und noch eine
Frage:

- liest du?
- hörst du?

Überspringen

Hey Max,
hier deine
Empfehlungen für
deine Woche



bereits gelesen

interessant

Die Geschichte vom Herz
Lorem larem ipsum dad
njdinkwendjwebn

Hey Max,
hier deine
Empfehlungen für
deine Woche



bereits gelesen

interessant

Die Geschichte vom Herz
Lorem larem ipsum dad
njidnkwendjwebn



ins Bücherregal ▼

Notizen



Hey Max,
hier deine
Empfehlungen für
deine Woche



bereits gelesen

interessant

Die Geschichte vom Herz
Lorem larem ipsum dad
njidnkwendjwebn

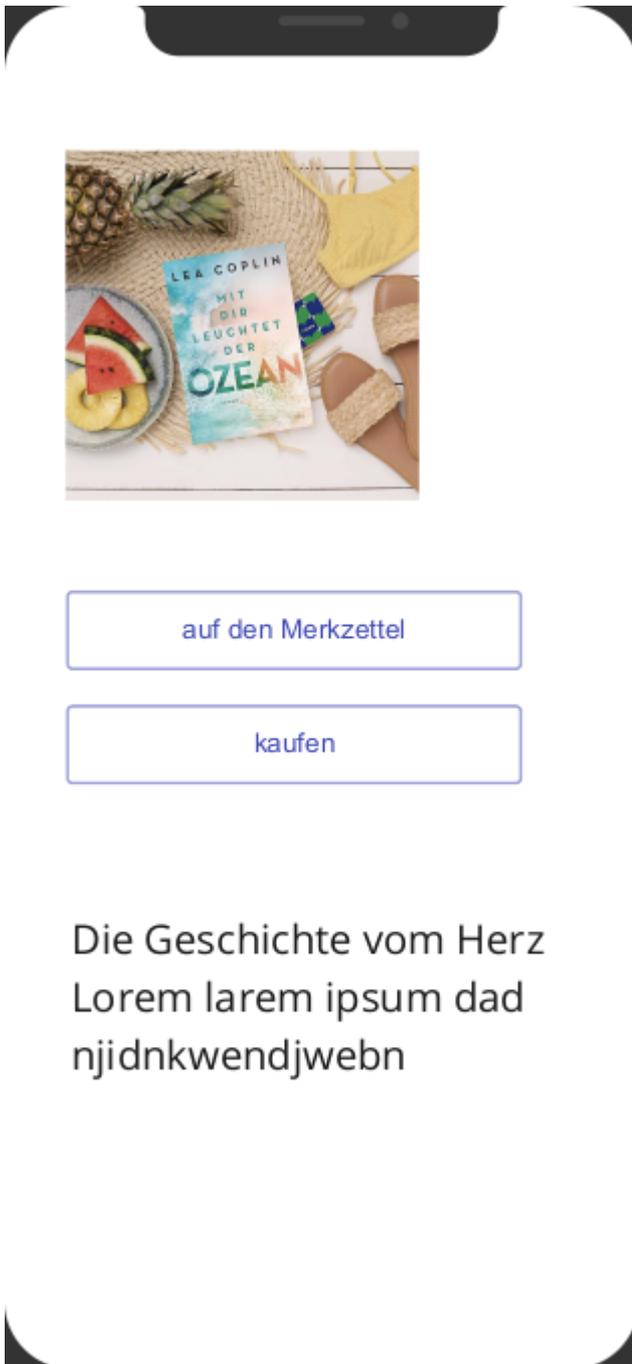


Abbildung: Prototypische Skizze des Sparks

Sich in einem Hackathon mit anderen Themen beschäftigen, oder, in diesem Fall, sich auf eine andere Art mit Themen beschäftigen hat allen viel Spaß gemacht. Es war toll zu sehen, was eine solche kreative Methode aus uns herausholen konnte. Wir sind davon überzeugt, dass dieses Feature unsere Kund:innen gut abholen kann, den Use Case und die identifizierten Challenges berücksichtigt. Er:sie bekommt regelmäßig „Futter“, das unserer Meinung nach gut passt und hat die Möglichkeit, dieses entsprechend einzusortieren.

Mal schauen, ob dieser Spark ein Feature-Feuer entzünden kann...

2_Backend for Frontend mit Apollo GraphQL

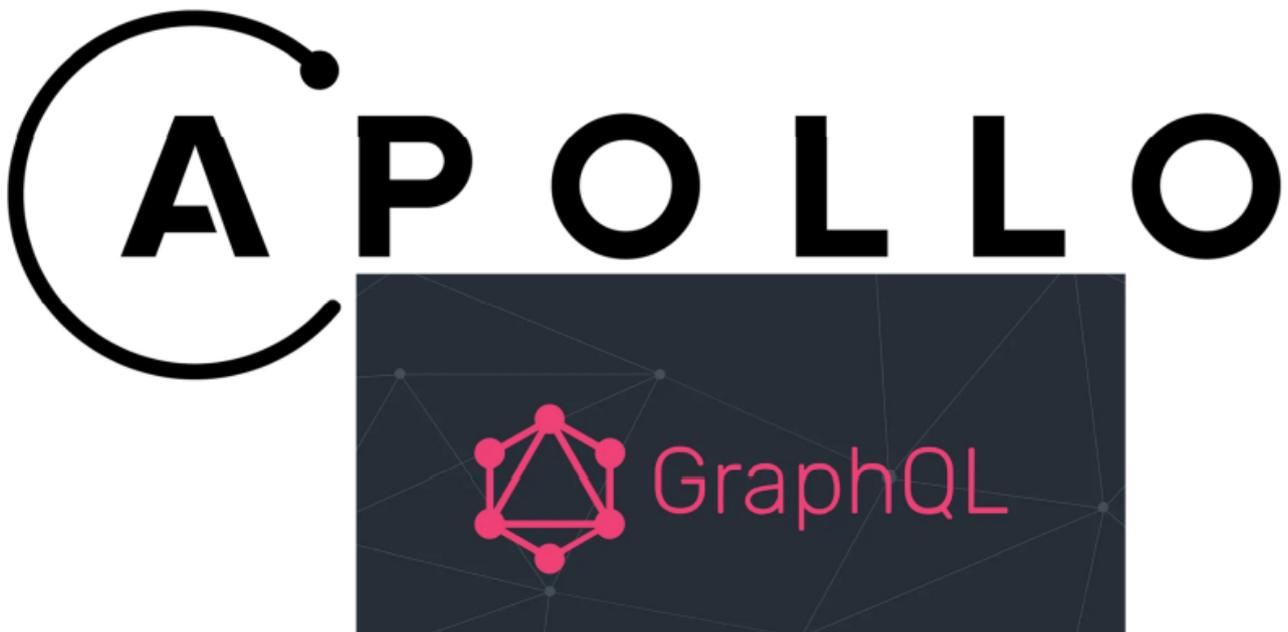
Worum ging es?

Ein [Thema](#), welches uns in den letzten Monaten sehr stark beschäftigt hat. Treiber hierbei ist unser Thalia-App-Team. In der App werden sowohl native Elemente entwickelt als auch Komponenten und Klick-Strecken von anderen Entwicklungsteams eingebunden.

Das Problem am Beispiel beschrieben:

- Team A möchte in einem API-Aufruf alle (für sie) notwendigen Daten von der API von Team B bekommen, um diese in ihrem Frontend darzustellen
- Team B möchte die API nicht erweitern, um Daten „durchzuschleusen“, die nicht zur Domäne des eigenen Teams gehören.

Backend for Frontend mit



Kann diese Problemstellung mit Hilfe von [Apollo](#) und [GraphQL](#) vereinfacht werden?

Im Rahmen des Hackathon sollte versucht werden, ein Setup aufzubauen und

anhand einer einfachen Beispielapplikation die Daten von verschiedenen Thalia-Services, per Backend for Frontend aggregiert, zu konsumieren.

Ergebnisse

Unsere Thalia-App gibt es sowohl in der [Android](#)-Version, wie auch für [iPhone und iPad](#)-Geräte. An folgendem Use Case sollte nun die Verprobung stattfinden:



Ihre persönlichen Empfehlungen



Start



Sortiment



Scanner



Merkzettel



Warenkorb



Das App-Team möchte über eine Schnittstelle nur Artikel inkl. Stammdateninformationen erhalten, für die es eine Empfehlung gibt.

Aktuell werden diese Daten bei uns von 2 unterschiedlichen Teams bereitgestellt

und - nicht überraschend - sind dementsprechend hierfür 2 Microservices verantwortlich:

Der Artikelservice stellt alle Artikelinformationen (EAN, Bilder, Texte, Preise etc.) zur Verfügung.

Der Empfehlungsservice kennt alle Artikel, für die es eine Empfehlung (Personalisierung, PaarKauf-Information, Tipps etc.) gibt.

Das Teams aus Java-Expert:innen, Android- und iOS-Spezialist:innen und Frontend-Entwickler:innen hat eine Umgebung aufgebaut, in der ein Proof of Technology auf beiden App-Versionen lauffähig war.

Per Apollo GraphQL konnten die Apps dann auf kombinierte Daten aus Empfehlungsservice und Artikelservice zugreifen. Der Empfehlungsservice hat in diesem Szenario dann nur ArticleIds (Primärschlüsselinformation) von empfohlenen Artikeln und eben nicht alle Artikeldaten ausgespielt. Der Artikelservice hat dann den Rest der Daten bereitstellt.

Das Team war am Ende des Tages sehr zufrieden: Neben den gesammelten Erfahrungen hat sich die Einschätzung gefestigt, dass hiermit das beschriebene Problem gelöst werden konnte. Eine Fortsetzung im Daily Business ist mehr als wahrscheinlich. Ein prima Ergebnis!

3_Judge a book by its cover

Worum ging es?

Meine Thalia [Lieblingsfiliale](#) in Münster → Rolltreppe in die erste Etage nehmen → dann leicht rechts halten → links hinter dem Info-Point: Mein Lieblingsbüchertisch! Kuratiert von den Kolleg:innen in der Buchhandlung - Neues, Altes, Preisgekröntes, Ungewöhnliches, Mutiges.

Wie schaffen wir es, die Exzellenz der Auswahl in den [Webshop](#), die [App](#), vielleicht sogar den [Tolino](#)-eReader zu bringen? Wir möchten den Kund:innen ein vergleichbares „Stöber-Erlebnis“ bieten, an einem Sonntag oder in einer

Lockdownphase. Oder anders formuliert: Wie können unsere Kund:innen bequem und barrierefrei diese Erfahrung an allen Berührungspunkten zu und bei Thalia nutzen?

Ergebnisse

Agenda

Vision / Ziel

Dein digitaler Büchertisch zum Erleben

10:30 - 11:15 Ideen sammeln

11:15 - 12:00 Zieldefinition

12:00 - 12:45 MITTAGSPAUSE

12:45 - 14:00 Work work work

14:00 - 15:00 Build a prototype

15:00 / 15:15 "Fertig"

15:30 Ergebnispräsentation

Ein bunt gemischtes Team aus den Bereichen Business Development, UX, Softwareentwicklung, Coaching und Produktmanagement hat sich dieser Fragestellung angenommen. Möglichst schnell sollte die Ideenphase in eine Entwicklungsphase übergeben.

Mit Hilfe eines digitalen [Whiteboards](#) (Brainstorming, thematisches Clustern und Voting) und einer straffen Agenda hat die Gruppe, aufgeteilt in 2 Teams, die folgenden 2 Fragestellungen bearbeitet:

- *Wie werden eigentlich die Tische zusammengestellt und wie kommen wir an die genaue Zusammenstellung der Artikel? Pro Tisch und pro Filiale?*
- *Wie kann so ein digitaler Tisch oder Regal aussehen?*

Für beide Fragestellungen haben wir Lösungen im eigenen Umfeld bei Thalia und Best Practices auf dem Markt angeschaut. Virtuelle Buchhandlungen, Raumbilder mit klickbaren Elementen & Augmented Reality: Bei der Recherche sind uns viele innovative Lösungen begegnet, die wir uns mit mehr Zeit sicher genauer angeschaut hätten. Als Inspiration für einen ersten MVP aber schon sehr hilfreich.

[Themenwelten](#), Empfehlungen unserer [Lieblingsbuchhändler:innen](#), kuratierte [Sortimentslisten](#) - schon heute hätten wir etliche Möglichkeiten die Artikelzusammenstellung über Schnittstellen bereitzustellen. Verknüpft mit der eindeutigen Filialkennzeichnung, dann in einem weiteren Schritt über die Tischkennzeichnung haben wir für uns festgestellt, das kann funktionieren. Eine gute Nachricht. Mehr konnten und wollten wir an der Stelle nicht erreichen.

Für den Büchertisch selbst hatten wir schnell die notwendigen Artikelattribute definiert. Es sind genau die, die Kund:innen in der Situation sehen: Titel, Autor:in, Klappentext, Preis, Cover-Bild und, um nun den Schwenk auf eCommerce zu schaffen: Bestandsverfügbarkeit in der Filiale (8 Bücher auf dem Stapel), ein Link zur Artikeldetailseite und natürlich die Möglichkeit zu kaufen bzw. zu merken.

Das Ergebnis mit viel UX- und Frontend-Magic kann sich sehen lassen.

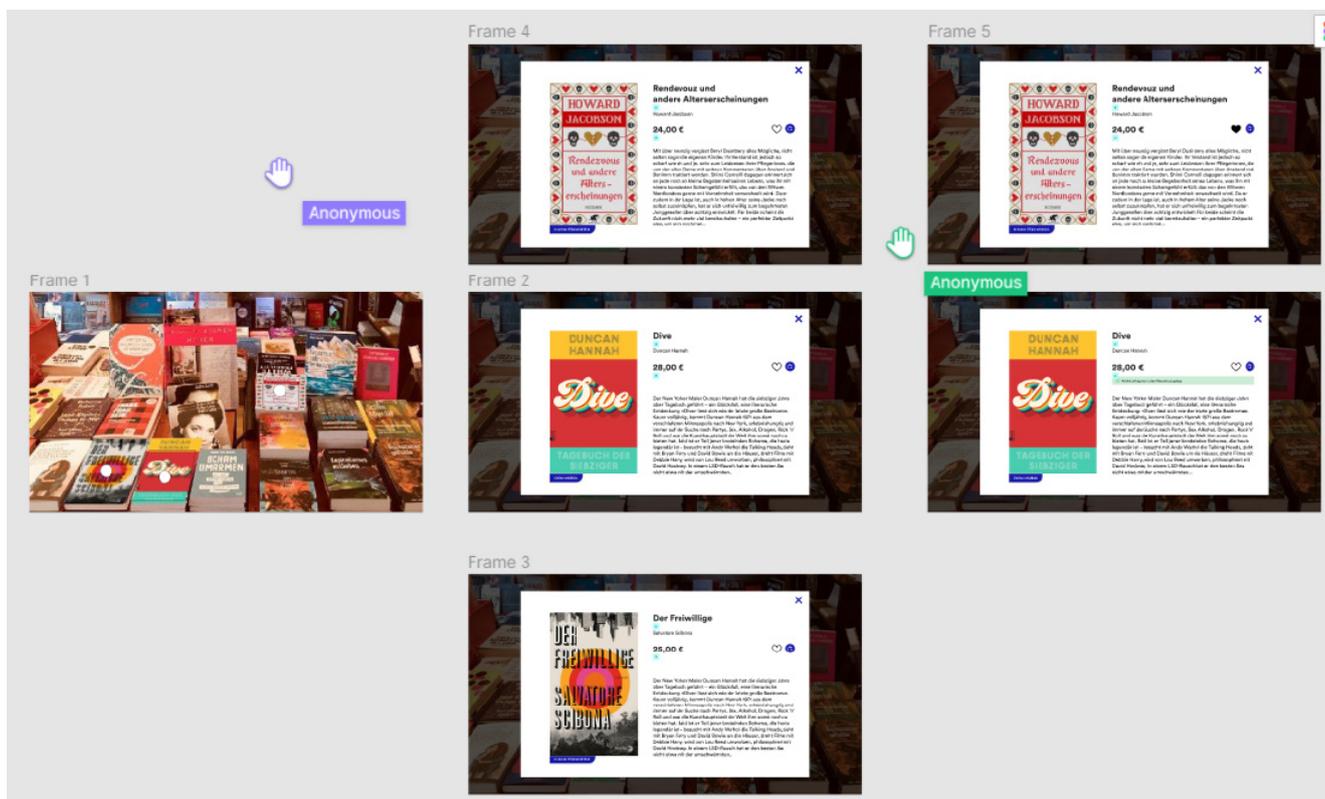


Abbildung: Ergebnispräsentation am Beispiel von 3 Artikeln des Büchertisches.

4_Frequent Pattern Mining FTW!

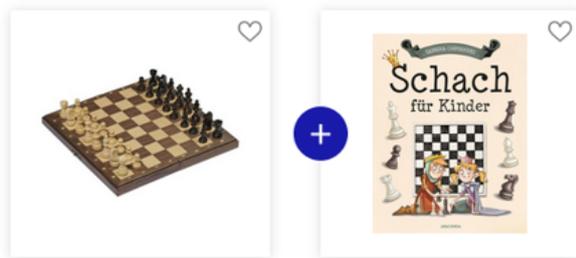
Worum ging es?

Personalisierung mit dem Ziel der perfekten Produkt-Empfehlungen sind für jeden eCommerce-Händler sowohl eine große Herausforderung wie auch eine *signifikante Möglichkeit Kundenzufriedenheit und Umsatz zu steigern.*

Dabei konzentrieren wir uns hier auf den Zusammenhang zwischen gleichzeitig gekauften Produkten. Wie finden wir heraus, dass die Ähnlichkeit zwischen einem Schachspiel und einem Lehrbuch sehr wahrscheinlich größer ist als zwischen einer Schürze und einer DVD?

Oder um es mit den Worten eines Kollegen zu beschreiben: Frequent Pattern Mining ist ein Weg für faule Leute (z.B. Informatiker:innen) diese Zusammenhänge zu finden und nutzbar zu machen, z.B. über Paarkaufempfehlungen.

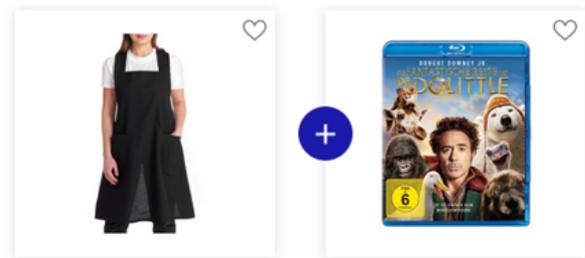
Wird oft zusammen gekauft



Goki 56920 - Magnetisches Schachspiel in...
Spielwaren
29,99 €
★★★★★ (1)

Schach für Kinder
Sabrina Chevannes
Buch (gebundene Ausgabe)
9,95 €
★★★★★ (0)

Wird oft zusammen gekauft



BUTLERS BISTRO Schürze L
100 x B 72cm
Anthrazit
15,99 €
★★★★★ (0)

Die fantastische Reise des Dr. Dolittle
Robert Downey jr.
Film (Blu-ray)
10,49 € ~~12,99 €~~
★★★★★ (7)

Abbildung: Paarkaufbeispiele auf Thalia.de

Ergebnisse

In einer überaus unterhaltsamen Präsentation wurden spannende Erkenntnisse geteilt:

- Datenanalyse, -bereinigung, -filterung im Vorfeld sind unabdingbar, um gute Ergebnisse zu erhalten. Schlicht 8 Millionen Transaktionen (mit

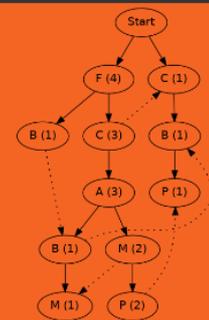
mehr als 2 Artikeln in einem Kauf) zu nehmen und dann auf ein Wunder zu hoffen, funktioniert nicht.

- Insbesondere die Filterung der richtigen Artikel erhöht signifikant die Ergebnisqualität. So blieben im Testlauf nur ~ 4.000 Artikel übrig, die in mind. 0,01% aller Transaktionen enthalten waren.
- Somit sind neben zu überprüfenden Artikelstammdaten (Stichwort kostenlose eBooks) auch übergreifende Sortimentsinformationen relevant.

Ein kleiner Teaser...

F, A, C, D, G, M, P, I
A, B, C, F, L, M, O
B, F, H, J, O
B, C, K, S, P
A, F, C, E, L, P, M, N

F, C, A, M, P
F, C, A, B, M
F, B
C, B, P
F, C, A, M, P



Alle Transaktionen sind fiktiv, Ähnlichkeiten mit real existierenden Transaktionen sind rein zufällig

Abbildung: Transaktionen und Ähnlichkeiten

Unterschiedliche technische Ansätze im Bereich FPGrowth wurden ausprobiert und bewertet:

- Python #1 (aus Pandas): Schlüssig war diese Implementierung nicht und funktioniert hat sie für unseren Use Case auch nicht. Daher keine Empfehlung, da „irgendetwas“ gemacht wurde, aber nicht FPGrowth.
- Python #2 (github.com/chonyy): Funktions- und lauffähig, aber nicht optimiert auf unsere Anforderung. Auch hier: Viel gelernt, aber keine 5 Sterne.
- [SparkML](#): Ein vielversprechender Ansatz, der ein wenig kompliziert zu implementieren ist. Jedoch überzeugt hier die Möglichkeit der Parallelisierung.

Wir haben gelernt, dass die Berechnung an sich mit einfachen Mitteln machbar ist, die Datenaufbereitung jedoch der Potentialbringer sein wird.

Eine Live-Demo vervollständigte die gelungene Präsentation und hinterließ

weniger fragende Gesichter, als das doch sehr technische Thema im Vorfeld vermuten ließ. Well done!

5_Spring-Cloud-Config: Konfigurationsmanagement für Microservices

Worum ging es?

Konfigurationsmanagement ist häufig eine fehleranfällige und lästige Aufgabe in der Software-Entwicklung. Insbesondere in einer Microservice-Architektur, in der gesamtheitliche Funktionen von mehr als einem Produktteam implementiert, getestet und deployed werden.

Dieses betrifft dann nicht nur mehrere Service-Instanzen, sondern eben auch unterschiedliche Services. Für Integrationstest ist es wichtig, dass auch auf den Stages vor der produktiven Umgebung diese konsistent und effizient gepflegt werden. Auf die Frage, ob dies auch ohne das Durchführen einer Deployment-Pipeline oder Datenbankänderung erfolgen kann, wollte dieses Team eine Antwort geben und ist bereits mit einer konkreten Idee gestartet: [Spring-Cloud-Config](#)

Im Rahmen des Hackathons sollten Erfahrungen mit dieser Implementierung gesammelt werden und eine Machbarkeitsstudie anhand einer konkreten Beispielanwendung erfolgen.

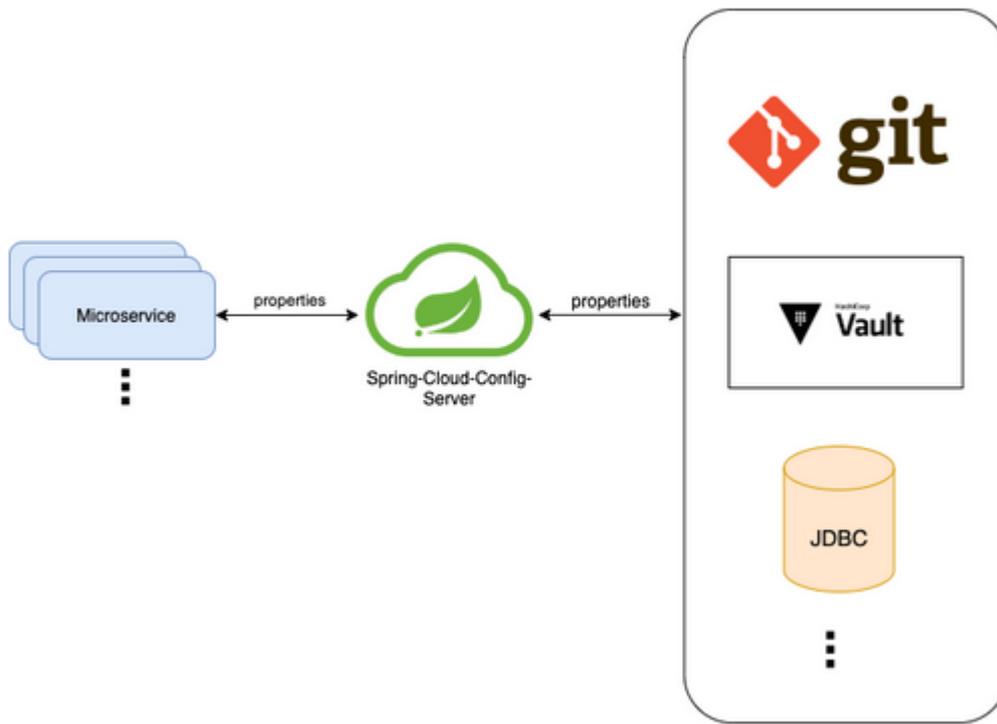


Abbildung: *Infrastruktur Idee, s. auch*
<https://cloud.spring.io/spring-cloud-config/reference/html/>
Ergebnisse:



Das Team hat sich eine fachliche Konfiguration als Beispiel genommen: Eine generelle Versandkostenfreiheit soll Webshop-spezifisch eingestellt werden.

Als Anmerkung: Wir betreuen auf einer einheitlichen IT-Plattform unterschiedliche Shops, neben unserer Marke Thalia.de auch weitere, wie z.B. bol.de, Thalia.at oder orellfuesli.ch.

Die Versandkosten-Information werden auf der Prüfen&Bestellen-Seite im Rahmen des Checkouts interpretiert. Technisch übernimmt hier der bestelluebersichtsservice die Orchestrierung und ist somit der Hauptakteur des Experiments. Als Messaging-System ist RabbitMQ im Einsatz, die Datenbank ist MySql und das Ganze läuft in einem Docker-Container.

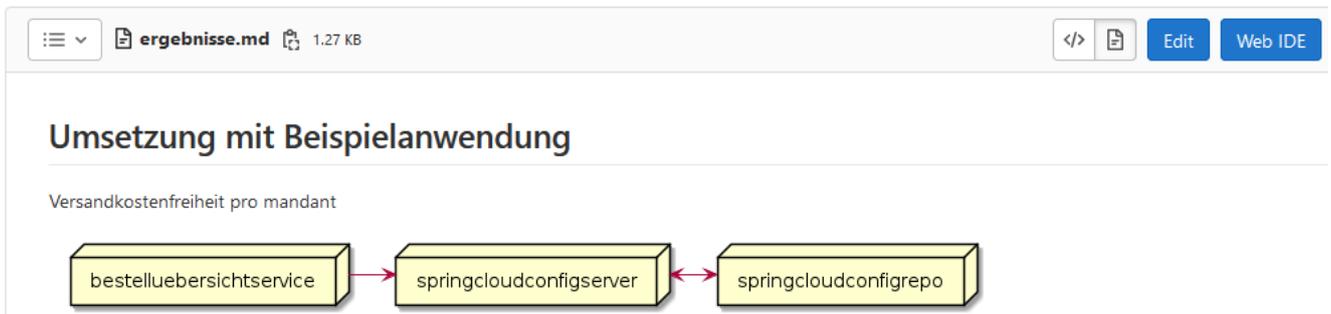


Abbildung: Konkrete Idee - Versandkostenfreiheit pro Webshop

In GitLab wurde ein entsprechendes Projekt aufgesetzt und die einzelnen Schritte dokumentiert.

Mit Hilfe von Spring-Cloud-Config-Server konnten die im Use Case benötigten Konfigurationsänderungen zur Laufzeit und ohne Deployments durchgeführt werden. Das in der Vergangenheit bereits mit einem Spring Boot-Service gearbeitet wurde, war von Vorteil und hielt die Aufwände in Grenzen.

Im Experiment wurde auch deutlich, dass insbesondere der Anspruch, zentrale Änderungen service- und teamübergreifend zur Verfügung zu stellen, hier an seine Grenzen kommt. Nicht zwingend technisch, sondern organisatorisch. Wir halten unsere Microservice-Architektur bewusst flexibel, um schnell und effizient Veränderungen in Produktteams durchzuführen. Je mehr Shared-Services wir nutzen, desto höher sind die Kosten so einer Veränderung.

Auch wenn die Zeit etwas knapp wurde: Bewiesen wurde, wie einfach sich ein dynamischer Konfigurationsserver aufsetzen und einbinden lässt. Ob sich dies nun auch für weitere Anwendungsfälle rechnet und wir diese Lösung übergreifend einsetzen? Das Team hat mit diesem Proof of Concept eine gute Entscheidungsvorlage präsentiert.

Zusammenfassung

HACKATHON 2021

59
Antworten

01:24
Durchschnittliche Zeit für das Ausfüllen

Geschlossen
Status

...

Ergebnisse anzeigen

In Excel öffnen

1. Welches Team hat aus deiner Sicht den Hackathon 2021 gewonnen?

[Weitere Details](#)

- (Buch-)Empfehlungen einsorti... 10
- Virtueller Büchertisch – Team "... 24
- Backend for Frontend für die ... 10
- Empfehlungen auf Basis von g... 7
- "versandkostenfrei" konfigurie... 8



Das Experiment hat sich gelohnt. Ein digitaler Hackathon funktioniert. Auf die Frage, ob und wie sich die Thalia-Kolleg:innen einen Hackathon 2022 wünschen war die Antwort jedoch eindeutig: Machen, und zwar analog vor Ort. Wir geben unser Bestes[].

Ach ja, war da nicht noch etwas mit einem Preis?! Ca. 60 Interessierte haben am Ende der Präsentationen ein Voting abgegeben. Unser neuer Wandpokal hat eine erste Signatur:

Die Teilnehmenden des siegreichen Teams Bookcave (# 2_Judge a book by its cover) konnten sich über diesen und ein kleines Überraschungspaket freuen.

Requirements Engineering -

Zurück zu den Wurzeln

In Retrospektiven und Lessons Learned wurde wiederholt Verbesserungspotential rund um das Thema Anforderungsmanagement (Requirements Engineering) benannt. Das Geschäftsmodell von Thalia bedingt eine enge Verzahnung von Prozessen und Touchpoints, um die Reise unserer Kund*innen zu einem inspirierenden und spielerisch einfachen Erlebnis zu machen. Und hält damit etliche Herausforderungen in der Produktentwicklung bereit:

- OmniChannel-Services - von der Filiale zum Webshop bis zum eReader
- Teamübergreifende Zusammenarbeit - von der Planung über den Schnittstellen-Kontrakt bis zum Regressionstest
- Integration vieler Stakeholder und deren Bedürfnisse - vom Marketing über den Kunden-Service bis zur Buchhaltung
- Priorisierung und Featureentscheidung - von der Idee über die Erfolgsmessung bis zur Weiterentwicklung

Aus diesem Wissen und Bedarf heraus haben wir vor einigen Monaten eine neue Community of Practise gegründet, mit dem Ziel, das Requirements Engineering (RE) bei Thalia weiter zu professionalisieren und dieses Wissen in die Organisation zu tragen.



Miro-Board - Was sind unsere wichtigsten Themen?

In einer der ersten Ideen, die wir in eine konkrete Maßnahme überführten, haben wir eine interne Schulung zu [IREB® CPRE Foundation Level](#) über einen externen Dienstleister organisiert. Wir wollten noch einmal besser verstehen, welche Werkzeuge uns zur Verfügung stehen und wie wir diese effektiver nutzen können. Hierbei haben wir uns bewusst für das Foundation-Level entschieden, um eine gute Basis zu legen. Schwerpunkte dieses Levels sind das Kennenlernen von Terminologie, Methodik und Notation im Requirements Engineering.



Miro-Board – die erste Maßnahme

Für einige Kolleg*innen war es eine Auffrischung bereits bekanntem, aber manchmal auch verschollenem Wissen. Andere haben Neuland betreten.

Wie wichtig es ist, Wissen und Impulse aus von unterschiedlichen Rollen, Erfahrungen oder Kompetenzen zu bekommen, kennen wir aus der Arbeit in unseren crossfunktionalen Produktteams. Daher haben wir bei der Zusammensetzung der 10 Schulungsteilnehmenden auf diese Diversität geachtet. Konnten aber auch bei einigen Kolleg*innen den Wunsch nach individueller Weiterbildung inkl. einer Zertifizierung ermöglichen.

Welche Motivation hattest Du für die Schulung?

Felix (Software Developer): Grundlegend war meine Motivation, einen Überblick und eine Einordnung der Aufgaben und Werkzeuge des Requirements Engineering zu erhalten. Außerdem war mir der Erfahrungsaustausch mit Kolleg*innen aus anderen Teams besonders wichtig. Diese Fragestellungen waren dabei für mich primär relevant:

- Warum betreibt man Requirements Engineering?
- Wie geht man konkret an die Aufgaben heran?
- Was kann ich davon für mich in den Arbeitsalltag einbauen?
- Wie wird das Thema in den anderen Teams gelebt?

Lukas (QA Manager): Meine Motivation als QA Manager für das Thema Requirements Engineering liegt in der Tatsache, dass meine komplette Arbeit auf Anforderungen basiert. Eine meiner Hauptaufgaben ist es, Umsetzungen gegen

die erhobenen Anforderungen zu prüfen - aber auch die Sicherung der Qualität von Anforderungen vor deren Umsetzung ist ein Thema, das mir am Herzen liegt. Requirements Engineering hilft mir, den gesamten Lebenszyklus von Anforderungen mit einem Blick für Details zu begleiten.

Julian (Software Developer): Meine ursprüngliche Motivation war einen Überblick über die verschiedenen Aufgaben des RE zu bekommen. Außerdem wollte ich gerne die Möglichkeiten kennenlernen, wie RE in der agilen Vorgehensweise eingesetzt werden kann.

Christoph (App Developer): Als Touchpoint hat man bei Thalia fast zu jedem Team Berührungspunkte, sodass eine gute Kommunikation über Anforderungen unerlässlich ist. Dies habe ich nicht nur beim Schreiben der Tickets für andere Teams, sondern auch beim Bearbeiten unserer eigenen gemerkt. Nicht nur, dass das Verständnis für bestimmte Dinge nicht da war, oftmals waren die Grenzen für eine bestimmte Aufgabe nicht klar abgesteckt, sodass durch Nachfragen schon einiges an Zeit verloren ging. Aus diesem Grund habe ich mich für Techniken interessiert, welche ein gemeinsames Verständnis fördern und zudem den Aufgabenbereich klar definieren. Wichtig hierbei war es mir, dieses Wissen nicht nur theoretisch zu erlernen, sondern auch anhand praktischer Beispiele gezeigt zu bekommen.

Auch in agilen Vorgehensweisen kann davon ausgegangen werden, dass man durch die Anfangsiteration und die kontinuierlichen Aufwände für die Requirements-Themen über die gesamte Projektlaufzeit auf einen Aufwand von ca. 15%-20% des Gesamtprojektaufwands kommt. Das ist in etwa gleich oder etwas geringer als bei nicht agilem Vorgehen. Der wesentliche Unterschied ist, dass die Aufwände anders verteilt sind und zielgerichtet genau dort anfallen, wo sie den größten Nutzen bringen.

Johannes Bergsmann, Requirements Engineering in der agilen Softwareentwicklung, dpunkt.Verlag, 2. Aufl. 2018

Wie korrespondiert nun ein vermeintlich klassischer Ansatz im Anforderungsmanagement mit den agilen Vorgehensmodellen innerhalb unserer

Produktentwicklung? Ziemlich gut. Was wir vorher schon ahnten, hat sich während der Schulung und den darauffolgenden Wochen manifestiert: Eine systematische, disziplinierte und strukturierte Herangehensweise unterstützt das iterative und inkrementelle Vorgehen. Die Community of Practise wird die Entwicklung mit großem Interesse in den nächsten Monaten verfolgen.

Hatte die Schulung schon Auswirkungen auf Deine tägliche Arbeit?

Jan (Product Owner): Allein wenn man die Prinzipien etwas deutlicher im Hinterkopf hat, hilft das schon in der täglichen Arbeit und der Überwindung des inneren Schweinehundes:

- Wertorientierung (Anforderungen sind Mittel zum Zweck, kein Selbstzweck)
- Stakeholder (Wünsche und Bedürfnisse der Stakeholder befriedigen)
- Gemeinsames Verständnis (Erfolgreiche Systementwicklung ist ohne eine gemeinsame Basis nicht möglich)
- Kontext (Systeme können nicht isoliert verstanden werden)
- Problem -Anforderung -Lösung (Ein unausweichlich ineinandergreifendes Tripel)
- Validierung (Nicht-validierte Anforderungen sind nutzlos)
- Evolution (Sich ändernde Anforderungen sind kein Unfall, sondern der Normalfall)
- Innovation (Mehr vom Gleichen ist nicht genug)
- Systematische und disziplinierte Arbeit (Wir können im RE nicht darauf verzichten)

Wenn ich Anforderungen bearbeite, achte ich stärker darauf und habe auch ein Gespür dafür, wo ich das gerade nicht tue, aber besser täte. Es fällt leichter Lösungen abzulehnen, die gar kein Problem lösen, oder wo dies nicht klar herausgestellt / weit genug gedacht ist. Es macht die Entscheidung leichter, etwas zu tun, oder nicht zu tun.

Deniz (Software Developer): Grundsätzlich denke ich, wir machen schon viel von dem Erlernten, könnten aber hier und da vielleicht noch präziser werden, da vieles nicht immer konsequent angewandt wird (z.B. Stories nur mit Akzeptanzkriterien). Es gibt auch Dinge, die wir gefühlt nicht machen: Zum Beispiel die Versionierung von Anforderungen zur Nachverfolgbarkeit oder die Validierung von Anforderungen. Einen großen Impact könnten wir denke ich auch erreichen, wenn wir einige Punkte an die Fachbereiche und die Geschäftsleitung spiegeln. Ich denke da z.B. an den angehängten Screenshot. Hier sieht man das wir häufig komplett komplementäre Ziele verfolgen. Wir wollen marktorientiert sein, sind aber häufig kundenorientiert.

[No 4] - Auf der Suche nach dem perfekten Termin

Zu viel, zu wenig oder gerade richtig? Die Frage nach Grad, Inhalt, Intensität und Effizienz bei wiederkehrenden Terminen umtreiben Teams und Organisationen nicht erst seit Corona.

In den ersten Tagen haben wir im Führungskreis unsere bisherigen Terminserien beibehalten, jedoch schnell gemerkt, dass es vermehrt zu Informationsdefiziten und Missverständnissen gekommen ist. Einfach gesagt: Wir waren nicht mehr gut genug abgestimmt.

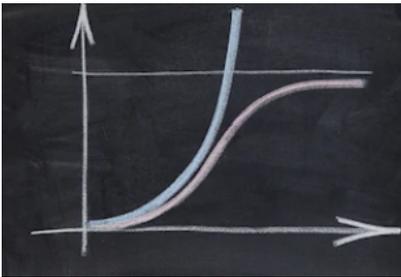
In Organisationen mit crossfunktional aufgestellten Teams ist diese Abstimmung der unterschiedlichen Führungskräfte (Servant Leader) zwingend notwendig, um u.a. allen Menschen in diesen Strukturen ein einheitliches Bild auf Rahmenbedingungen, Regeln und Unternehmensrichtlinien zu geben.

Bisher hatten wir im Bereich IT eCommerce folgende Terminserien (ein Auszug), die nur in Ausnahmefällen Remote durchgeführt wurden:

- **ProduktorgaDaily**: **täglicher** Austausch einiger (!) IT- und Produktmanagement TeamleiterInnen

- IT-Leitungskreis: **wöchentliche** Abstimmung aller TeamleiterInnen im Bereich IT
- Austausch ScrumMaster / TeamleiterInnen: **2-wöchentlicher** Termin der Agile Coaches mit dem TeamleiterInnen aus der Produktorganisation

Unsere konkrete Herausforderung: Sicherstellen, dass in Zeiten mit schneller Veränderungsfrequenz (teilweise mehrfach täglich) und erhöhter Relevanz für die Menschen in der Produktorganisation die Informationen geteilt und kommuniziert werden.



Hier unsere Lernkurve am Beispiel des täglichen **ProduktorgaDaily-Termins**:

1. Iteration - nach ca. 3 Tagen

Veränderung:

- IT-Leitungskreis nimmt am täglichen ProduktorgaDaily teil

Gut: Besserer Abstimmung insbes. am Anfang der Corona-Krise in Bezug auf Unternehmensregeln

Schlecht: Kommunikation in die Teams und Einbindung der wichtigen Führungsrolle ScrumMaster fehlt. Es kommt erneut zu bereits o.g. Problemen; ProduktorgaDaily wird eher IT-lastig

2. Iteration - nach ca. 1,5 Wochen

Veränderung:

- Beteiligung der ScrumMaster am ProduktorgaDaily
- Zusätzliches tägliches IT-Leitungskreis-Meeting (15 min)

Gut: Konsolidierter als vorher; es werden nur noch die wirklich dringenden und

akuten Themen besprochen; IT-Lastigkeit nimmt ab; Corona-Zustand hat sich eingeschwungen; Veränderungsfrequenz nimmt ab

Schlecht: Themen werden doppelt besprochen; der ‚Spirit‘ vom eigentlichen ProduktorgaDaily ist abhandengekommen durch die Fokussierung auf ‚dringend‘. Kein Austausch über weitere übergreifende Themen mehr möglich; mittel- und langfristige Anliegen und Diskussionen liegen brach.

3. Iteration - nach ca. 3 Wochen

Veränderung:

- Zusätzliches tägliches IT-Leitungskreis-Meeting (15 min)
 - ProduktOrga-Daily wird aufgelöst
 - Neue ProduktOrga-Besprechung im ursprünglichen Kreis wird auf 3x pro Woche je 30 min geändert
 - Austausch ScrumMaster/TeamleiterInnen wird auf den Regeltermin verschoben

Gut: Tägliche IT-Leitungskreis-Meeting findet großen Zuspruch und wird bis auf weiteres etabliert; allgemeine Produktorga-Themen finden wieder Raum und Zeit

Unter spezieller Beobachtung: Noch unklar, ob der regelmäßiger Austausch zwischen ScrumMaster/TeamleiterInnen so ausreicht oder auf wöchentlich geändert werden sollte.

4. Iteration - pending...

Fest steht: Ändern sich die Rahmenbedingungen, müssen Verhaltensweisen, Vorgehen, Werkzeuge mindestens überprüft und häufig angepasst werden. Und wenn es nur ein Termin ist.

Mehr zum Thema

[*Both sides of the story - Thalia in Zeiten von Corona*](#)

[No 1] - Alleine ist man weniger zusammen

Im eCommerce-Bereich von Thalia sind wir mittlerweile im Home Office. Unsere Entwicklungs-Teams, die Agile Coaches, alle Product Owner, die gesamte IT und natürlich auch alle TeamleiterInnen.

Unsere Organisation und die Art, wie wir arbeiten ist bisher auf Präsenz ausgerichtet. Wir haben unsere Räume so gestaltet, dass Menschen, die zusammen arbeiten auch zusammensitzen. Wir treffen Annahmen, dass teamübergreifende Zusammenarbeit besser funktioniert, wenn der persönliche Kontakt da ist. Wenn für die Klärung eines Problems maximal die nächste Etage besucht werden muss. Wenn das Treffen an der Kaffeemaschine (und wir haben wirklich gute Kaffeemaschinen) das Erstellen von JIRA-Tickets ersetzt. Wenn ein beiläufiges ‚kann mal jemand...‘ ein halbstündiges Meeting obsolet macht. Wenn bei der ausführlichen Bewertung des Tatorts (aka Bundesligaspiel, beliebiges Online-Spiel) am Montagmorgen auch Meinungen von ‚Über-den-Flur-Eilenden‘ gehört werden können.



Unser IT-Leitungsteam im Home Office.

Und jetzt haben wir Zoom. Und Slack. Und Telefonkonferenzen. Ach ja, und eMails.

Natürlich haben wir auch vorher schon diese Kommunikationstechnologien genutzt. Aber sie waren, bei persönlicher Kommunikation, eher die Ausnahme als die Regel.

Was wir schon ahnen:

Arbeiten nach agilen Werten und Vorgehensmodellen – das hilft: Transparenz durch digitale Boards, Offenheit bei Feedbacks in auch mal angespannten Situationen, Fokus auf die Umsetzung der richtigen Dinge und vor allem: Auf Veränderungen zu reagieren. Und – yes indeed – wir haben es hier mit einer Planänderung zu tun.

Crossfunktionale Teams, die selbstorganisiert ihre Produkte entwickeln – sind sehr gut vorbereitet. Die klaren Strukturen der agilen Vorgehensmodelle (z.B. Scrum), die gemeinsamen Ziele (Sprintziele oder

Roadmap-Themen der Teams) und die gelebte Selbstverantwortung helfen beim Operationalisieren der Remote-Arbeit. Es ist weiterhin klar, an welchen Themen wie gearbeitet wird. Das Vakuum der Orientierungslosigkeit ist sehr gering.

Was wir noch nicht wissen:

Wie sich eine mehrwöchige Home Office-Phase auf die Organisation auswirkt:

- Wie verändert sich unsere Art zu kommunizieren?
- Werden wir weniger effizient?
- Entstehen mehr Missverständnisse insbesondere in der teamübergreifenden Zusammenarbeit?
- Und werden wir mehr oder weniger Spass an unserer Arbeit haben?

Aktuell kann ich sagen: [Alleine ist man weniger zusammen](#) und das kann auch die Anzahl und Häufigkeit von Slack-Nachrichten nicht wirklich ändern.

Mehr zum Thema:

[*Both sides of the story - Thalia in Zeiten von Corona*](#)

Both sides of the story - Thalia in Zeiten von Corona



Auch Thalia hat die Corona-Krise getroffen. Und das in ganz unterschiedlicher Form. Unsere stationären Buchhandlungen sind geschlossen, das eCommerce hat aktuell Konjunktur und die Art, wie wir arbeiten hat sich praktisch von einem auf den anderen Tag verändert.

Waren wir gut vorbereitet? Schwierig, die Frage mit ‚Ja‘ zu beantworten. Das würde implizieren, dass wir mit so einem Szenario gerechnet hätten. Haben wir nicht, wie wahrscheinlich auch der Rest der Welt.

Aber können wir von Technologien, Strukturen und Arbeitsweisen profitieren, die wir uns in den letzten Jahren angeeignet, die wir geübt haben? Zumindest diese Frage können wir nach 3 Wochen zumindest vorsichtig optimistisch bejahen.

In den kommenden Wochen wollen wir in kleinen Beiträgen berichten, wie wir als OmniChannel-Händler, als eCommerce-Unternehmen, als Tech-Firma - eben als Thalia, mit dieser Ausnahmesituation umgehen.

[No 1] - [Alleine ist man weniger zusammen](#)

[No 2] - [Wenn IT-Operations in die Glaskugel schaut...](#)

[No 3] - [Corona Kommunikations-Knigge](#)

[No 4] - [Auf der Suche nach dem perfekten Termin](#)

[No 5] - [Was sagt eigentlich unser Management?](#)

<0110/> hackathon@thalia in Münster 2019

*Neue Technologie, die Dinge spielerisch einfach macht! Neuer Inhalt, der zählt!
Neues Modell, das uns und unsere Prozesse ein bisschen besser macht. Neue
Idee, die du entdecken möchtest! Welche Themen fallen dir ein? Dieser
Hackathon kennt keine Grenzen.*

Für den diesjährigen Hackathon haben wir uns bewusst für ein offenes Format entschieden und damit getreu dem agilen Prinzip ‚inspect and adapt‘ das Feedback zur Themeneinschränkung beim [letzten Mal](#) berücksichtigt.

Die Beteiligung sprach für sich: 18 motivierte Menschen, 5 interessante Pitch-Vorschläge und 1 Tag Zeit.



Pitch der Ideen



Der Preis!

Das Ergebnis: 3 aus 5. Die Gruppen hatten sich schnell formiert und ihren Weg jeweils nach Bullerbü, Springfield oder Entenhausen in der Meeting-Area in Speicher 6 gefunden.

Die Türen in Winterfell waren zu jeder Zeit für den uneingeschränkten Zugriff auf

die eiskalten Getränke weit geöffnet. Ebenso der Blick auf den Siegerpreis, der in den Katakomben unterhalb der Burg ... genug davon, ich schweife ab.

Kommen wir zu den Themen:



<https://www.okd.io>

Projekt: OKD - OpenShift Kubernetes Distribution

Worum ging es?

Docker - auch bei uns ein Thema. Wie können wir die Administration und Orchestrierung unserer wachsenden Container-Infrastruktur besser und effizienter managen? Dieser Fragestellung will das Team mit 8 Leuten auf den Grund gehen und schaut sich das OpenSourceprodukt OKD genauer an.

[OKD](#) ist ein Upstream-Projekt für OpenShift aus der Produktpalette von Red Hat und setzt auf [Kubernetes](#) auf.

Ergebnisse

`oc cluster up` - mit diesem Kommando ist das Team in die Evaluierung gestartet. Das hat schon mal gut und vor allem schnell funktioniert. Ein SingleNode Cluster wird mittels Minishift und VirtualBox aufgebaut. In einem Gitlab-Projekt wurde eine Testapplikation mit Webfrontend und ein Dockerfile angelegt und deployed. Parallel haben Teammitglieder versucht, Minishift auf einem MacOS-Gerät zum Laufen zu bekommen. Nach anfänglichen Konflikten mit der

installierten VirtualBox gelang auch dies.

Während der Abschlusspräsentation konnte eine schlanke Webanwendung (Login-UI) gezeigt, eine neue Version gebaut und über eine eigene Jenkins-Pipeline deployed werden. Et voilà - es funktioniert.

Während man mit Kubernetes alles bauen kann, was man so haben möchte (Load Balancer, Network Policies, Benutzerverwaltung, ...), bringt OKD alle diese Features „out of the box“ mit. OKD ist somit eine gute open source Alternative zu OpenShift.



<https://developers.google.com/awareness>

Projekt: Google Awareness API

Worum ging es?

Unser eCommerce-Standort in Münster befindet sich in der Speicherstadt, wir arbeiten und treffen uns in unterschiedlichen Gebäuden. Es gibt zur Mittagszeit eine Kantine auf dem Campus, die ‚Hofrunde‘ ist ein allseits anerkanntes Kommunikationsinstrument und der Parkplatz ist ebenfalls auf dem Gelände. Das Team wollte analysieren, ob die [Google Awareness API](#) für die Erfassung von Zeiten genutzt werden kann, um somit anhand von Standorten zwischen Freizeit und Arbeit zu unterscheiden. Im ersten Schritt sollte dies auf Basis der geographischen Positionen von 2 Speichergebäuden erfolgen.

Ergebnisse

Die Zeit war eng bemessen. Ein Teil des Teams hat die App installiert und ein anderes die REST-Schnittstellen zum Sammeln und Anzeigen der Tracking-Daten entwickelt. Erkenntnis nach 6 Stunden: Die Anwendung wirkte instabil. Der Energieverbrauch durch fortlaufendes, aktives Ansprechen der App war hoch,

d.h. durch fehlende Automatisierung wurden die genutzten Geräte stärker beansprucht. Eventuell war auch der gedachte Anwendungsfall hier einfach nicht der richtige. An dieser Stelle unkritisch - bei Thalia gibt es Vertrauensarbeitszeit:-).

Einige Java-Entwickler waren Teil des Teams und konnten so die genutzte Programmiersprache [Kotlin](#) kennenlernen und im Zuge der REST-Schnittstellen sofort damit entwickeln. Viele Vorteile wurden im direkten Vergleich mit Java gesehen - insbesondere die Null-Safety-Fähigkeit und ‚Lines of Code-Reduktion‘ konnte begeistern.

Das Team hat bei der abschließenden Präsentation alles gegeben und im Rahmen einer Live-Performance Tracking-Daten gesammelt. Sehr sportlich!



<https://www.testcontainers.org/>

Projekt: Testcontainers

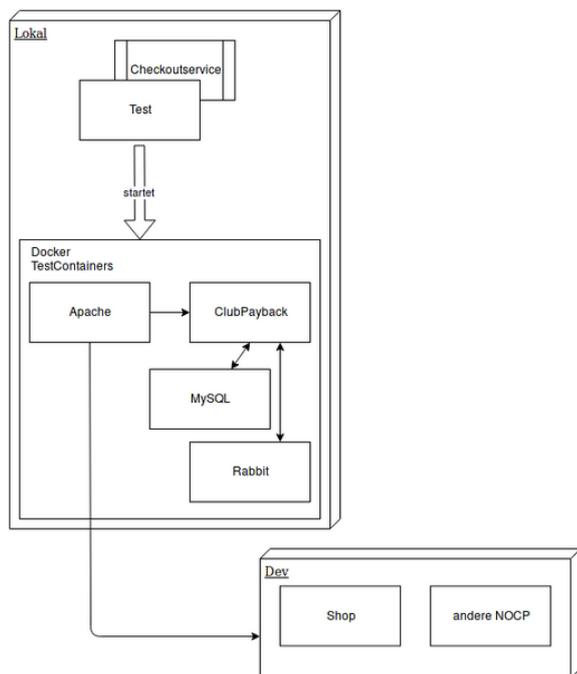
Worum ging es?

Mal wieder die Dev-Umgebung kaputt oder der Test schlägt erst auf der Integrationsumgebung nach dem Deployment fehl? Dieser Zustand kostet extrem viel Zeit. In [unserer Microservice-Architektur](#) haben unsere Produktteams zunehmend die Herausforderung, Test in Abhängigkeit zu [SCS-Services](#) anderer Teams auszuführen.

In unserem [Thalia-Club-Programm](#) schließt der Kunde die Mitgliedschaft im Rahmen eines Checkout-Prozesses ab. In diesem Prozess wird eine Payback-Kontenverknüpfung durchgeführt. Dieser Prozessbestandteil liegt in der Verantwortung eines separaten Produktteams. Wie kann nun das ‚Club-Checkout-Team‘ seine End2End-Tests (Front- und/oder Backend) mit möglichst hoher Unabhängigkeit zum ‚Payback-Team‘ gestalten? Beide Services wurden auf der Docker-Infrastruktur aufgesetzt.

[Testcontainers](#) ist ein Framework, um Docker-Container innerhalb von JUnit-Tests zu verwenden (Infrastructure-as-Code). Das Team möchte prüfen, ob hierdurch die Tests auf der Dev- oder Integrations-Umgebung reduziert werden und die Ausführung von Tests ohne Deployment auf die Teststages möglich ist.

Ergebnisse



Darstellung der beteiligten Services

Der [PoC](#) zeigt, wie von einander abhängige Services aus unserer Produktentwicklung im Test auf der lokalen Umgebung zu starten und auszuführen sind, dass wir vor dem Deployment auf den jeweiligen Umgebungen Fehler feststellen können. Und eben nicht hinterher;-).

Wichtig ist hierbei der Blick auf die transitiven Abhängigkeiten. Je mehr Kaskaden an inkludierten Testcontainern es gibt, desto komplizierter kann es werden. In einem Folgeschritt sollten noch die Laufzeiten angeschaut werden, die sich aufgrund der Start/Stop-Prozesse der Container erhöht haben. Das Team sieht in der Nutzung von Testcontainers eine Menge Potential. Klasse, dass sich hier Leute aus unterschiedlichen Produktentwicklungen zusammengefunden haben. So konnte schon der erste ‚Reality-Check‘ erfolgen.



Andreas, Christian, Benjamin und Julian haben die Zuschauer-Jury überzeugt.

Die Ergebnisse und die Präsentation haben dann auch am Schluss des Hackathons die Zuschauer-Jury überzeugt. Nach einer Stichwahl stand das Team als Gewinner fest. Herzlichen Glückwunsch!

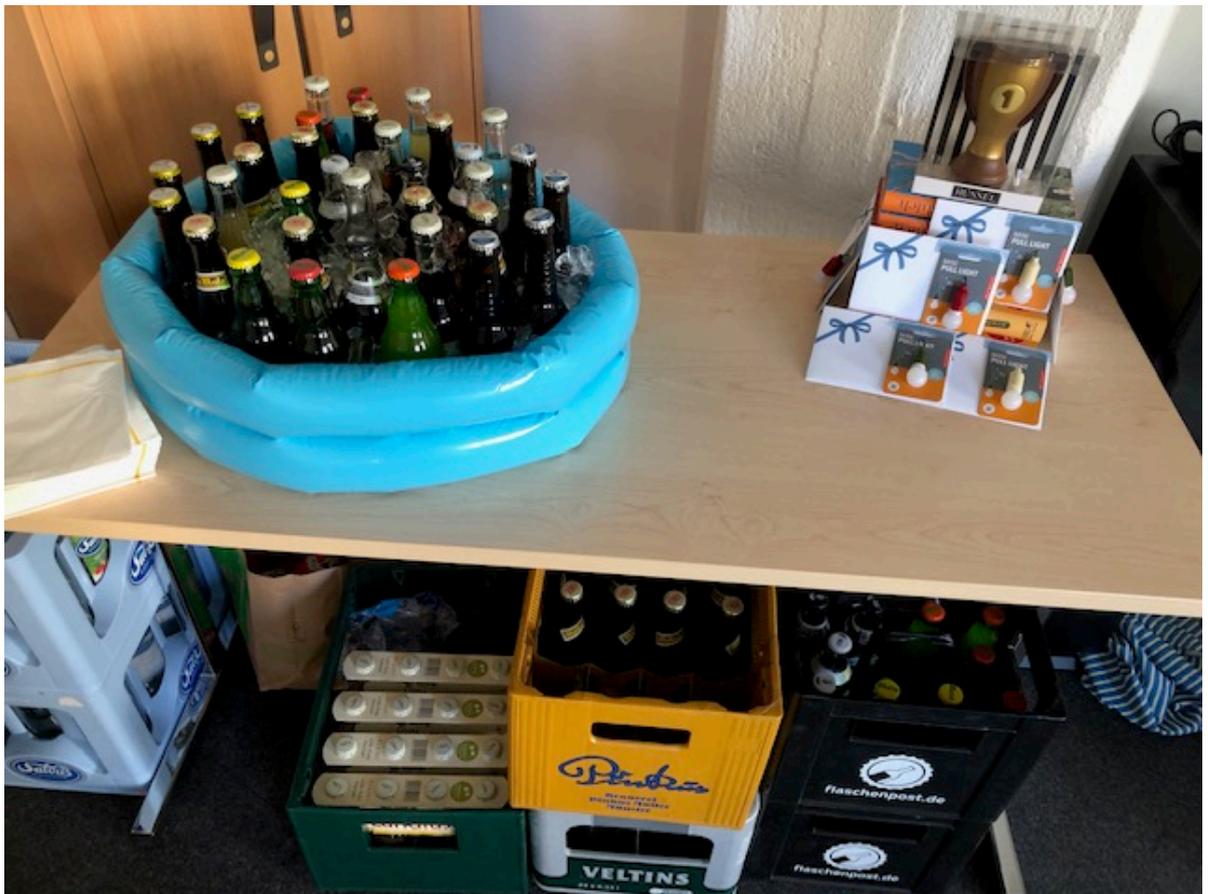
Fazit

Ist es Zufall, dass sich 2 von 3 Themen rund um die Docker-Infrastruktur konzentrieren? Wahrscheinlich nicht - wir beschäftigen uns im Moment sehr mit dem Thema und wollen eine steile Lernkurve über Einsatz, Handling und Nutzen dieser Infrastruktur erreichen. Die Erfahrungen aus dem Hackathon werden uns hierbei unterstützen. Well done!

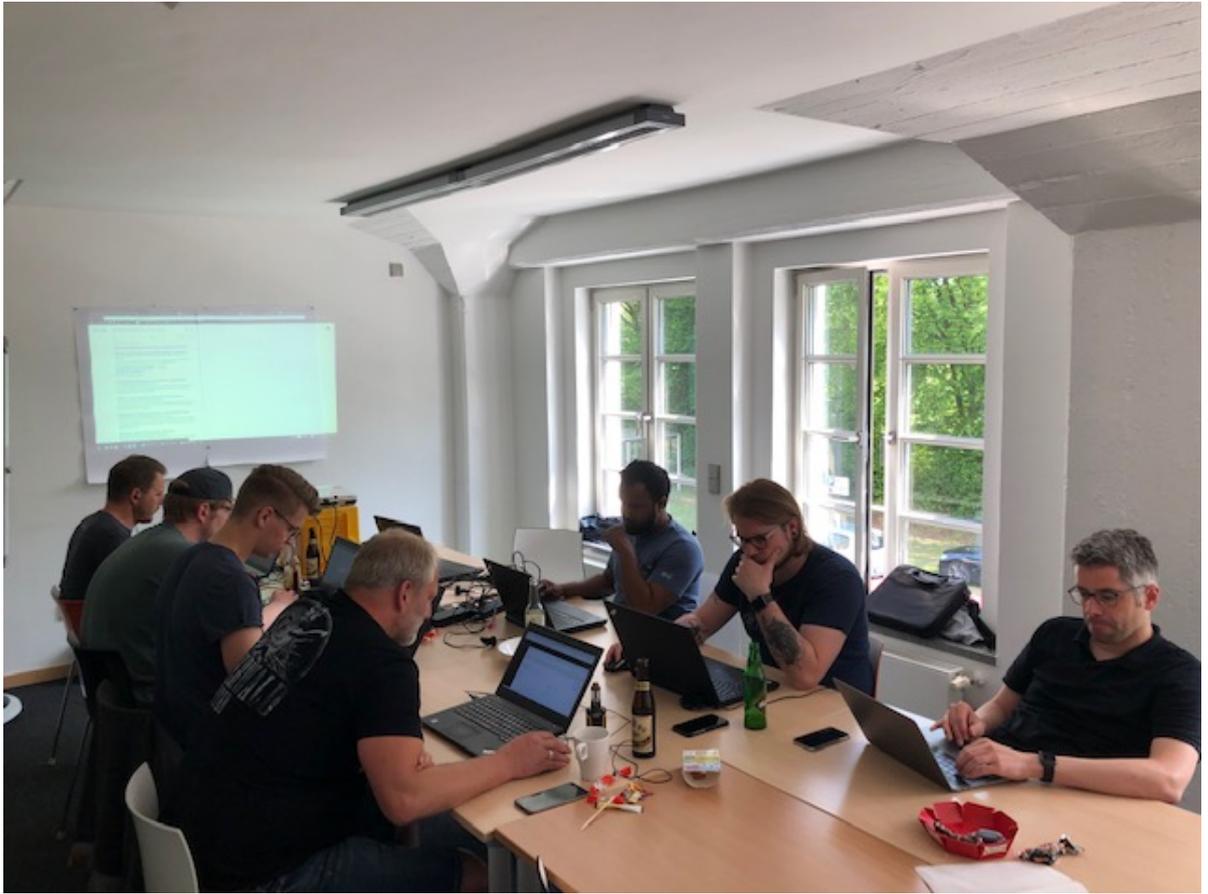
Im letzten Jahr haben wir die [App-Entwicklung](#) von Thalia von Berlin nach Münster umgezogen. Jetzt sitzen die Kollegen/innen im Büro nebenan. Macht richtig Spaß, diese teamübergreifende Zusammenarbeit zwischen App-Experten und Java-Spezialisten. Kotlin wird aktuell bei uns in der App-Entwicklung genutzt. Mal sehen, was jetzt in den ‚Java-Teams‘ passiert:-).

Mit folgenden Stimmungsbildern verabschieden wir uns und schließen mit den Worten:

Das machen wir wieder - see you in 05/2020.









Scrum Deutschland 2018 - Domain Driven Scrum

Crossfunktionale, selbstorganisierte und -verantwortliche Teams sind das Herzstück unserer Produktorganisation. Um den damit verbundenen Herausforderungen, insbesondere im Hinblick auf die Verantwortungsübergabe- und -übernahme, gerecht zu werden haben wir auch unsere bisherigen Führungsrollen verändert.

Zum Startpunkt unserer agilen Transformation vor 2 Jahren gab es weder Scrum Master, Agile Coaches oder erfahrene Servant Leader bei uns - aber natürlich Teamleiter und andere Führungskräfte:

- Wie sind wir in dieser Situation gestartet?
- Welche Annahmen in Bezug auf Trennschärfe und mögliche Überschneidungen haben wir zwischen den beiden Rollen getroffen?
- Sind für unsere Teams (und uns) die jeweiligen Aufgabenbereiche klar definiert?
- Wie hat sich in dieser Zeit die Zusammenarbeit zwischen Coaches und Leader entwickelt?

Entlang dieser Fragen haben wir bei [Scrum Deutschland 2018](#) berichtet - aus den beiden Sichtweisen Scrum Master und Servant Leader mit konkreten Erfahrungen, Erkenntnissen und Vorschlägen.



Eine tolle Gelegenheit für uns in den Austausch mit anderen Interessierten zu gehen und nach dem Vortrag weiter zu diskutieren und Erfahrungen auszutauschen - es hat Spaß gemacht!



Matthias Hochschulz |
m.hochschulz@thalia.de



Claudia Landmesser |
c.landmesser@thalia.de

Software Delivery ist keine Abteilung!

„Software Delivery“ geht weiter als Analyse und Entwicklung. Sie hört auch nicht nach der Qualitätssicherung auf. Nach dem Deployment ist vor dem Deployment. Und der Betrieb ist sowieso inklusive, das ist mal klar.

Dieses Ziel erreichen wir nicht mit ausschließlich Analysten und Softwareentwicklern. Ohne QA-Spezialisten, Operations-Experten und Product Ownern in diesen Teams ist die Herausforderung nicht zu schaffen.

Daher besteht „Software Delivery“ bei uns aus selbstorganisierten, cross-funktionalen Produktteams mit dem Ziel, autonom und selbstorganisiert Software zu entwickeln, zu deployen und vor allem Mehrwerte für unsere Kunden zu

generieren.

Sind wir da schon am Ziel? Nein. Wollen wir da hin? Definitiv.



Tägliche Besprechung im crossfunktionalen Team. Im Hintergrund das Monitoring der Systeme in Produktion.

Um im Bereich der agilen Softwareentwicklung richtig durchstarten zu können, haben wir uns in den Produktteams für Scrum als Vorgehensmodell entschieden. Was uns dabei sehr wichtig ist:

- Verantwortung für den Gesamterfolg
- Erzielen von gemeinsamen Ergebnissen
- Regelmäßige, kritische Überprüfung der Qualität der Zusammenarbeit & der Prozesse
- Messen der eigenen Produktivität
- Transparenz im Tun und in der Kommunikation
- Kundenorientierung
- Mut haben: zur Einfachheit und Offenheit

Sind wir bei all dem schon Experten? Nein. Wollen wir das werden? Definitiv.

„You build it, you run it“ – aus dem klassischen Legacy-Umfeld kommend bauen wir unsere Umgebung und unsere Prozesse aktuell radikal um und haben eine Menge Zukunftsbilder im Blick:



- Umbau vom ‚Release-Train‘ hin zur ‚Release-Subway‘
- Wissensaufbau im Team vom Datenbankindex bis zum Frontend-Skript
- Umbau von Infrastrukturanforderungen hin zu automatisierter Servicebereitstellung
- Erweiterung automatisierter Qualitätssicherung vom JUnit-Test zum Regressionstestautomaten
- u.v.m.

Haben wir das alles schon erreicht? Nein. Wollen wir das? Definitiv.

Wir befinden uns gerade in einer der spannendsten Phasen einer Transformation. Die Wege sind nicht ausgetreten, vieles ist neu, muss definiert oder überhaupt erst erschaffen werden. Alle müssen dazulernen.

Unsere produktbezogene Softwareentwicklung bedeutet ständige Veränderung in Bezug auf den Kunden, den Markt und auf uns. Keine kleine Herausforderung, aber eine sehr motivierende.

Also: Software Delivery ist keine Abteilung – es ist eine ganze Menge mehr.



Martin
Ernst -
Teamleiter
Software
Delivery



Claudia
Landmesser
-
Teamleiterin
Software
Delivery

Interesse & Lust bekommen, uns auf diesem Weg zu begleiten und Dich aktiv einzubringen? Dann freuen wir uns auf Deinen Kontakt.