

Was machen wir mit unserem „Webshop?“

Unser Webshop

Start der Entwicklung für unseren Webshop war im Jahre 2004. Zu der Zeit waren wir (damals noch buch.de internetstores AG) noch ein reines [eCommerce Unternehmen](#), und mit diesem Fokus wurde unser Webshop entwickelt.



Thalia.de 2004
Webseite so sah die
aus

Seitdem ist viel passiert...

Unser Business Model und unser Unternehmen haben sich verändert

- Ebooks sind stark gewachsen und die Tolino Allianz hat sich gegründet
- Schon 2004 hatten wir Multichannel-Funktionalitäten (z.B. Filialabhohlung), aber spätestens mit dem Zusammenschluss von buch.de

& Thalia lag der Fokus auf „Cross-Channel“, und die Integration der unterschiedlichen Kanäle hatte begonnen.

Jetzt befinden wir uns auf dem Weg zum Omnichannel-Unternehmen und versuchen, die tatsächliche Verschmelzung der Kanäle zu realisieren. Die ersten Ansätze sind da, aber die Frage ist natürlich, wie weit können wir diese Reise mit unserem aktuellen Webshop gehen?

Die Methode, mit der wir Software entwickeln, hat sich verändert

Ähnlich wie sich das ganze Thema „Agile Softwareentwicklung“ entwickelt hat, haben wir uns natürlich auch verändert und setzen aktuell auf Scrum in einer [Produktorganisation](#).

Da [Conway's Law](#) eine große Rolle spielt, ist die Frage: Inwieweit passt unser Webshop in unsere neue Produktorganisation?

Die Technologien und Architekturen, mit denen wir Software entwickeln, haben sich verändert

Glücklicherweise wechselten wir 2004 auf das Spring Framework, und Spring wird immer noch aktiv weiterentwickelt – keine Selbstverständlichkeit, da ja Spring ebenfalls erst in [2004 in der Version 1.0](#) veröffentlicht wurde. Haben wir evtl. eine der ältesten noch aktiv weiterentwickelten Spring Anwendungen ☐ ? Natürlich haben wir unseren Webshop regelmäßig weiterentwickelt und sind nicht mehr auf Spring 1.0, aber ein Kern und die ursprünglichen Prinzipien blieben bestehen.

Auf der anderen Seite ist die „Best-in-Suite vs Best-in-Breed“ Debatte. Es gibt neuerdings immer mehr spezialisierte Tools / Frameworks, die in ihren Spezialgebieten besser sind (NoSQL Datenbanken, JavaScript Frameworks, alternative JVM Sprachen). Die Frage ist: Inwieweit können wir unseren Webshop mit den unterschiedlichen Tools kombinieren? Reichen uns die Möglichkeiten, die wir haben, aus oder müssen wir feststellen, dass sich die zukünftigen Business-Anforderungen nur mit Technologien umsetzen lassen, die wir nicht einsetzen können?

<#0010/> hackathon@thalia.de in Berlin

<#0010/>

<hackathon@thalia.de/> in Berlin zum Thema „Neue App Technologien“

Am 30. Juni 2017 fand der 2. Thalia Hackathon statt, diesmal mit dem Thema „Neue App-Technologien“ und am Standort Berlin. Mit dabei waren Kollegen aus den App- & Backend Teams, dem Digital Sales Management und dem Service Center Münster.

Wir starteten am Morgen gut gelaunt mit zwei **Lightning Talks** zum Thema „**Amazon Alexa**“ und „**Künstliche Intelligenz bei Thalia**“. Alexa ist seit Oktober 2016 in Deutschland verfügbar und hat bereits das Interesse einiger Mitarbeiter geweckt. Mit Alexa kann der Anwender digitale Dienste über eine Sprachsteuerung nutzen. Thalia beschäftigt sich im Service-Center schon seit längerer Zeit mit dem Thema „Künstliche Intelligenz“ (KI). Unter anderem im Kontext der Bearbeitung von Kunden-E-mails. Wir gehen davon aus, dass die KI in Verbindung mit Sprachsteuerung und Chatbots in den kommenden Jahren für Thalia stark an Bedeutung zunehmen wird. So lag es nahe sich damit zu beschäftigen.



Nach den Vorträgen wurden 3 Projektvorschläge präsentiert, Teams gebildet und die Themen bearbeitet. Am späten Nachmittag trafen sich die Gruppen, um ihre Erkenntnisse zu teilen und die Arbeiten zu präsentieren.

Thema 1: The Voice of Thalia - Eine Buchsuche im Dialog (am Beispiel Alexa)

Wie wäre es, Alexa nach Buchempfehlungen zu fragen? Vielleicht nach Empfehlungen unserer Thalia-Mitarbeiter? „Alexa, empfehle mir ein Buch zu Harry Potter“. Gesagt, getan...

Eine Gruppe nahm sich der Aufgabe an, die Buch-Empfehlungen zu realisieren. Dazu wurden **Intents** definiert und bestehende Services aus der **Thalia-E-Commerce-Plattform** angesprochen.

Eine 2. Gruppe beschäftigte sich damit einen **Kaffee-Experten** zu entwickeln. Also Alexa dazu zu nutzen, ganz alltägliche Problemstellungen zu lösen. Dazu wurden Skills für die Domäne „Kaffee“ realisiert, indem passende **Intents** im Zusammenspiel mit **AWS-Lambda-Functions** erstellt wurden. Alexa wurde unter anderem darauf trainiert zu folgenden Fragen Antworten zu geben:

- Welchen Kaffee kannst du mir aus einer Rösterei in Leipzig empfehlen?
- Wieviel Wasser brauche ich für <XXX> Gramm Kaffee?

Thema 2: Buchempfehlungs-Agent für Messenger

Ein weiteres Team nahm die Herausforderung der **Chatbots** an. Auch hier ging es darum Buch-Empfehlungen auszuspielen. Dazu nutzen wir die **API.AI** Plattform, die im September 2016 von Google übernommen wurde und mit der es möglich ist „**Conversational User Interfaces**“ zu gestalten. Die Plattform bietet unter anderem eine **Slack-Chat-Integration** an, die vom Team zusammen mit **Heroku** (node.js) genutzt wurde, um die Business-Logik zur Abfrage der Buchempfehlung zu implementieren. Wir trainierten die KI darauf eine Antwort auf folgende Frage zu geben (<XXX> steht für einen beliebigen Buchtitel):

- Empfehle mir ein Buch zu <XXX>

Die Herausforderung lag darin, das Buch zu identifizieren und anschließend mit unserer Empfehlungs-Engine einen Vorschlag zu ermitteln.



Weiterführende Links:

<https://api.ai/>

<https://www.heroku.com>

Thema 3: Anwendungen und Dialoge

Eine dritte Gruppe beschäftigte sich mit dem Thema **Dialogerstellung** bei Sprachsteuerungen und Chatbots. Eine Erkenntnis war, dass es für ein rundes Produkt wichtig ist, erst die Dialoge zu erstellen und anschließend die Skills zu

entwickeln. Die Komplexität der Dialoge ist nicht zu unterschätzen. Unsere UX/UI- und Fachexperten waren begeistert, sich mit dieser neuen Art der Kommunikation zwischen Kunde und Services auseinanderzusetzen.



Fazit

Alle Gruppen konnten am Ende des Tages beeindruckende Ergebnisse präsentieren. Das hatten die wenigsten zu Beginn erwartet. Die Ansteuerung von Alexa funktionierte ebenso, wie die Chatbot-Integration. Die Erkenntnisse aus der Dialog-Erstellung haben uns gezeigt, dass wir nicht nur technische Herausforderungen meistern müssen.

Vielen Dank an die Organisatoren des Events. Pizza, Snacks und Getränke lieferten die dringend benötigte Energie, um in der kurzen Zeit voranzukommen. Vielen Dank an Mirko, der uns mit Rat und Tat beim Thema Alexa zur Seite stand.

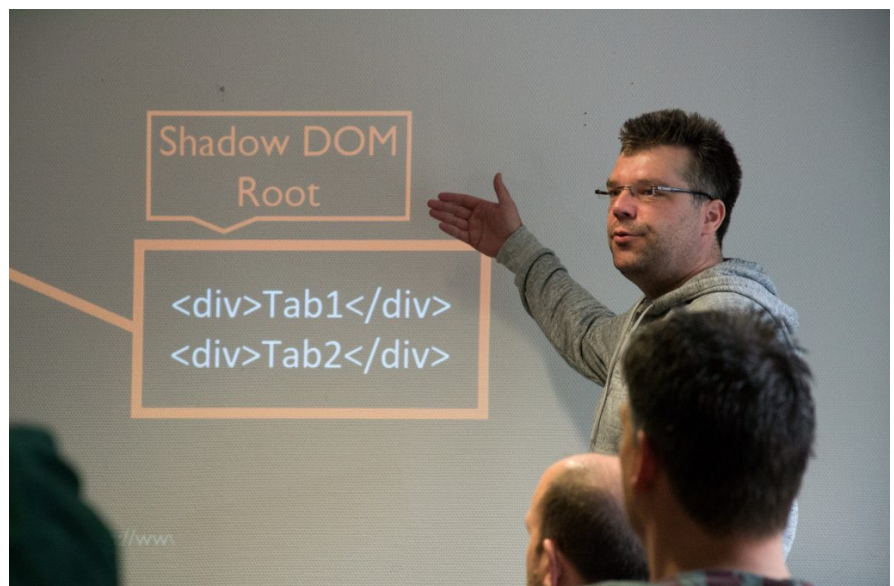


**<#0001/> <hackathon@thalia/> in
Münster**



Am 12.05.2017 fand unser Hackathon am Standort Münster statt. Auch unsere Berliner Kollegen waren mit vertreten.

Wir haben uns morgens weit entfernt vom Tagesgeschäft getroffen. Für Verpflegung war (reichlich) gesorgt, Kabel waren gelegt, Technik war eingerichtet, die Agenda stand - wir konnten ohne Verzögerung sofort „offkicken“!



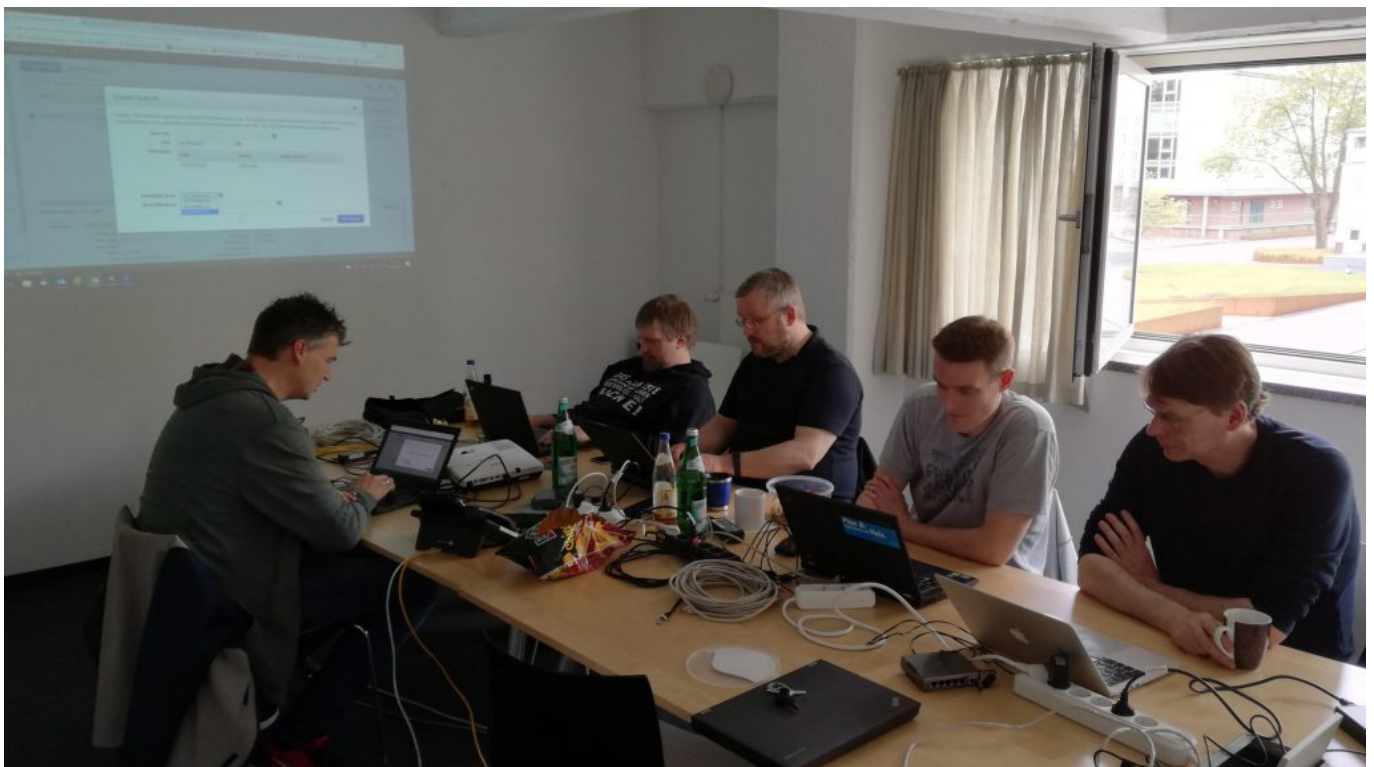
Lightning Talk „HTML 5 WebComponents“

17 Aktivposten, ein Organisator und ein paar Schaulustige (dazu gehöre dann wohl ich) konnten sich zunächst zwei informative „Lightning Talks“ zu den Themen „HTML 5 WebComponents“ und „WebGL“ anhören. Wer nicht teilnehmen konnte und sich trotzdem für eine grundlegende Einführung interessiert (Ursprung, Intention, Potential, Verbreitung), den Verweise ich gerne auf die beiden Referenten.

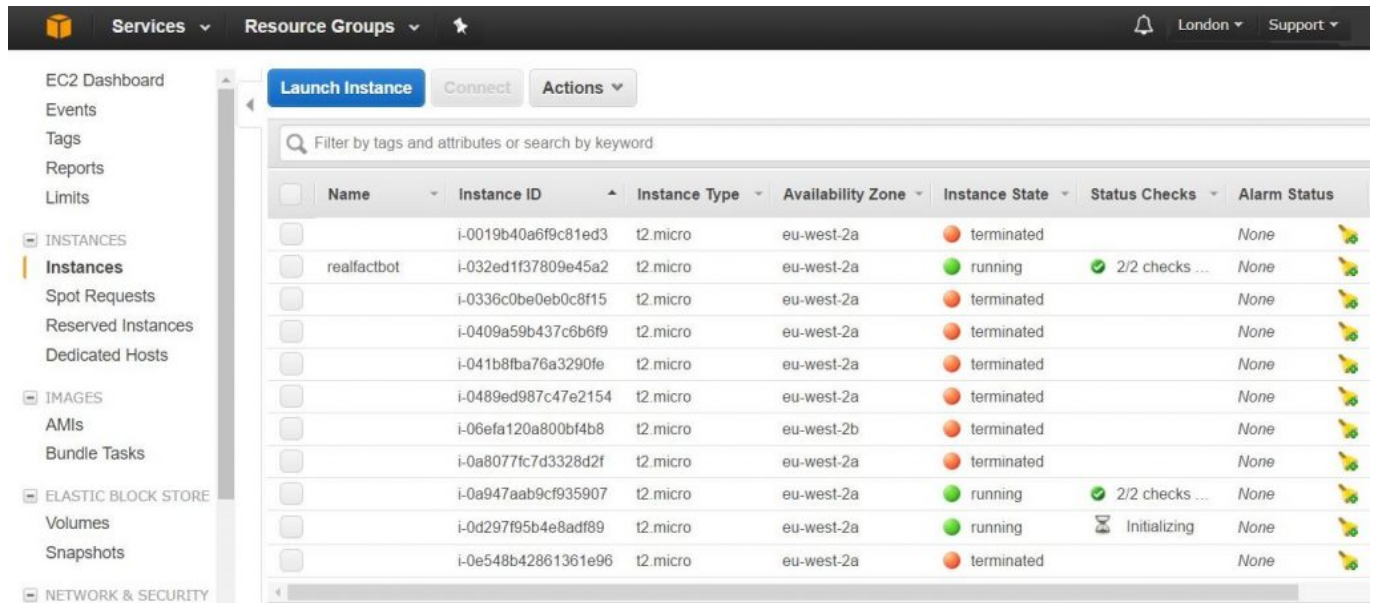
Danach wurden vier Projektvorschläge vorgestellt, von denen drei basisdemokratisch ausgesucht und in Folge in den gut besetzten Teams (5/6/6) bearbeitet wurden. Die Agenda sah eine Präsentation der Ergebnisse zu 16:30 Uhr vor, was alle Teams einhalten konnten.

„AWSome Scaling“ mit „Alternative Fact Provider“

Das Projekt hatte seinen Schwerpunkt in betrieblichen Aspekten. Es wurde eine Public Cloud im Hinblick auf Infrastruktur, RZ-Anbindung, Sicherheit, Automation, Deployment sowie automatische Skalierung evaluiert. Für einen reibungslosen Ablauf wurde das Teamgefüge zunächst mit Currywurst Chips geschmiert. Dann wurde von Ops Experten eine Infrastruktur aufgebaut und mit ersten Systemen bestückt, die möglichst nah an einem produktiven Einsatzszenario angelehnt war.



Fazit nach der ersten Stunde: „Wir haben Frankfurt gelöscht.“ Das war unsere Chance, ungeplant auch den Support des Cloud Anbieters zu testen. Nach einer steilen Lernkurve stand dann die Infrastruktur und wartete hungrig auf eine ganz besondere Anwendung.



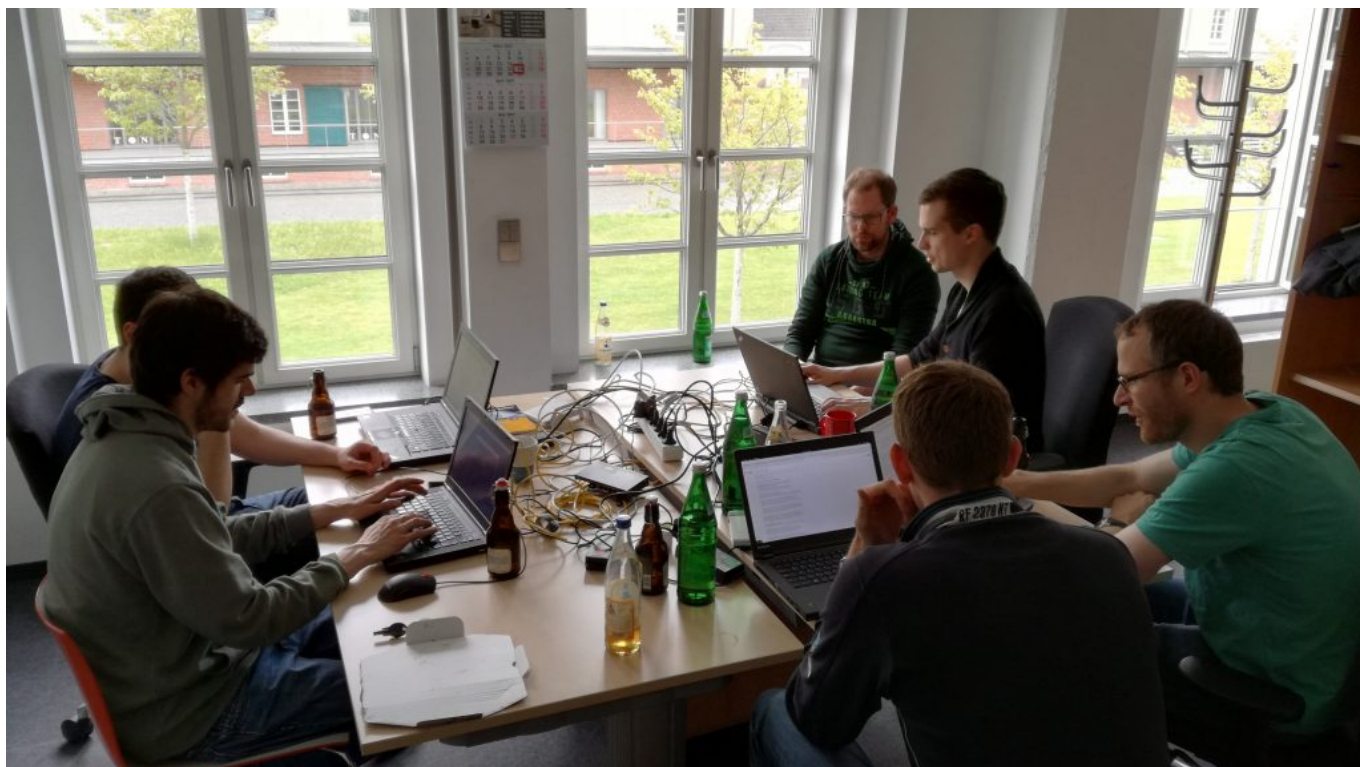
Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status
	i-0019b40a6f9c81ed3	t2.micro	eu-west-2a	terminated		None
realfactbot	i-032ed1f37809e45a2	t2.micro	eu-west-2a	running	2/2 checks ...	None
	i-0336c0be0eb0c8f15	t2.micro	eu-west-2a	terminated		None
	i-0409a59b437c6b6f9	t2.micro	eu-west-2a	terminated		None
	i-041b8fba76a3290fe	t2.micro	eu-west-2a	terminated		None
	i-0489ed987c47e2154	t2.micro	eu-west-2a	terminated		None
	i-06efa120a800bf4b8	t2.micro	eu-west-2b	terminated		None
	i-0a8077fc7d3328d2f	t2.micro	eu-west-2a	terminated		None
	i-0a947aab9cf935907	t2.micro	eu-west-2a	running	2/2 checks ...	None
	i-0d297f95b4e8ad8f9	t2.micro	eu-west-2a	running	Initializing	None
	i-0e548b42861361e96	t2.micro	eu-west-2a	terminated		None

Um das angestrebte Ziel zu komplettieren, steuerten die Dev Experten die hochaktuelle Zeitgeist-Anwendung „Alternative Fact Provider“ bei. Die Anwendung wurde dabei in die Cloud deployed und kommunizierte mit einer Datenbank im Thalia-Rechenzentrum. Aspekte wie das automatisierte Ausrollen neuer Softwareversionen inklusive einer automatischen, bedarfsorientierten Skalierung der Rechenkapazität, Kommunikation mit unserer eigenen Infrastruktur und Bewerten der dabei (natürlich) auftretenden Latenzen wurden betrachtet.

Dass das Thema Cloud nicht so trivial zu handhaben ist, wie es im Hochglanz Marketing dargestellt ist, wussten wir vorher. Als eines von vielen Werkzeugen in unserem Portfolio ist „Public Cloud“ aber definitiv eine spannende Ergänzung.

Spring 5 - Functional Reactive Programming

Am Beispiel einer Chat-Funktion im Browser, mit der der Kundensupport die Kunden im Webshop unterstützen kann, hat das zweite Team neue Funktionalitäten aus dem stark ersehnten [Spring 5](#) angetestet.



„Functional Reactive Programming“ in Spring 5 ändert maßgeblich die Art und Weise, wie Client und Server miteinander kommunizieren. Während bei gewohnter REST-Kommunikation der Client die gesamte Response erwartet, eröffnet die Reactor-Adaption von Spring 5 ein sukzessives Verarbeiten von Teil-Responses. Im Rahmen des Hackathons nutzte das zweite Team die starke Spring-Abstraktion der Websocket-Implementierung von Reactor. Dabei wurde eine „ereignisgetriebene“ Anwendung entworfen, die auch bei hoher Last mit wenigen Threads auskommt, und gleichzeitig durch die anhaltend geöffnete TCP-Verbindung per Websocket nicht (wie bei alternativen Technologien) die Nutzung der Anwendung blockiert.

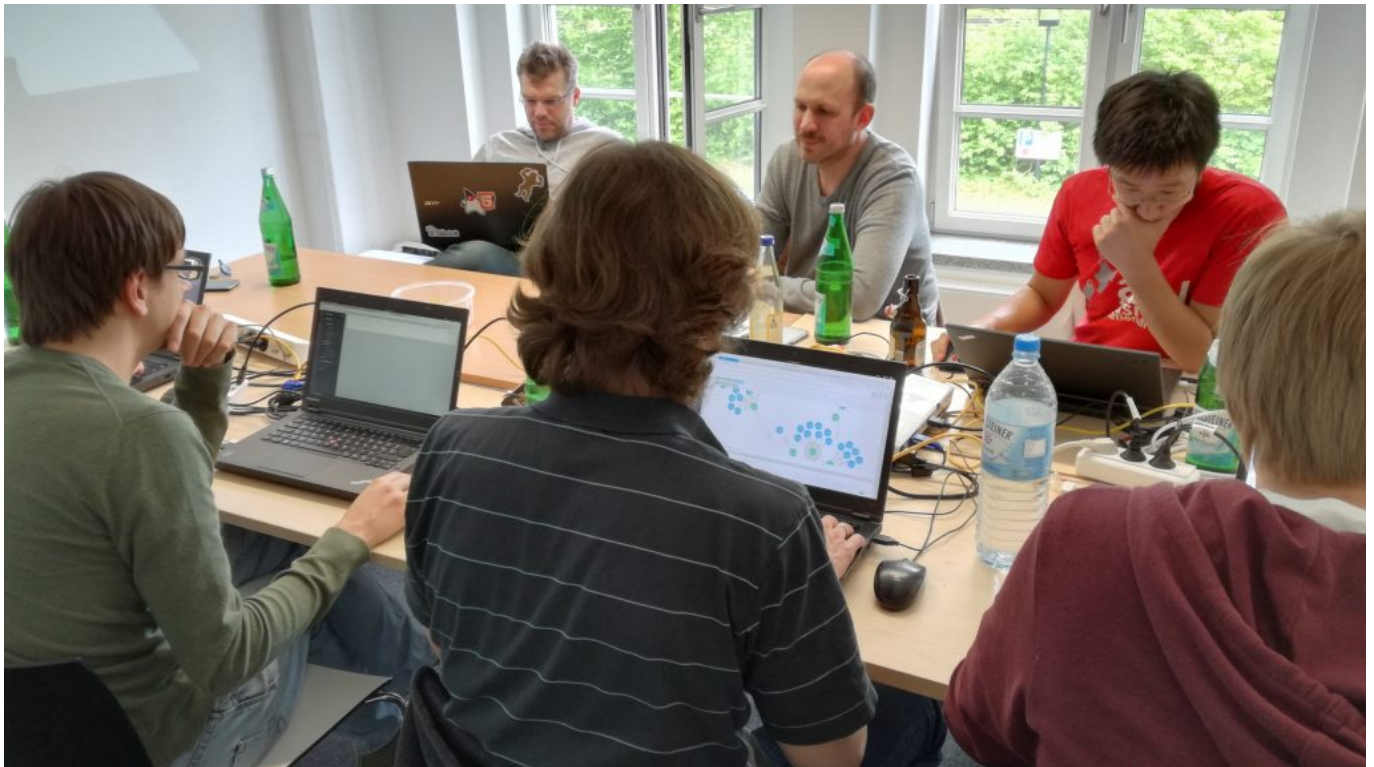
Fazit: Die Technologie ist sehr grundlegend, die Anwendungsfälle dafür zahlreich. Spring 5 wird freudig erwartet. Das würde uns erlauben, „neu über Sachen nachzudenken“.

Auswertung von Artikelbewertungen mit Hilfe einer Graphendatenbank

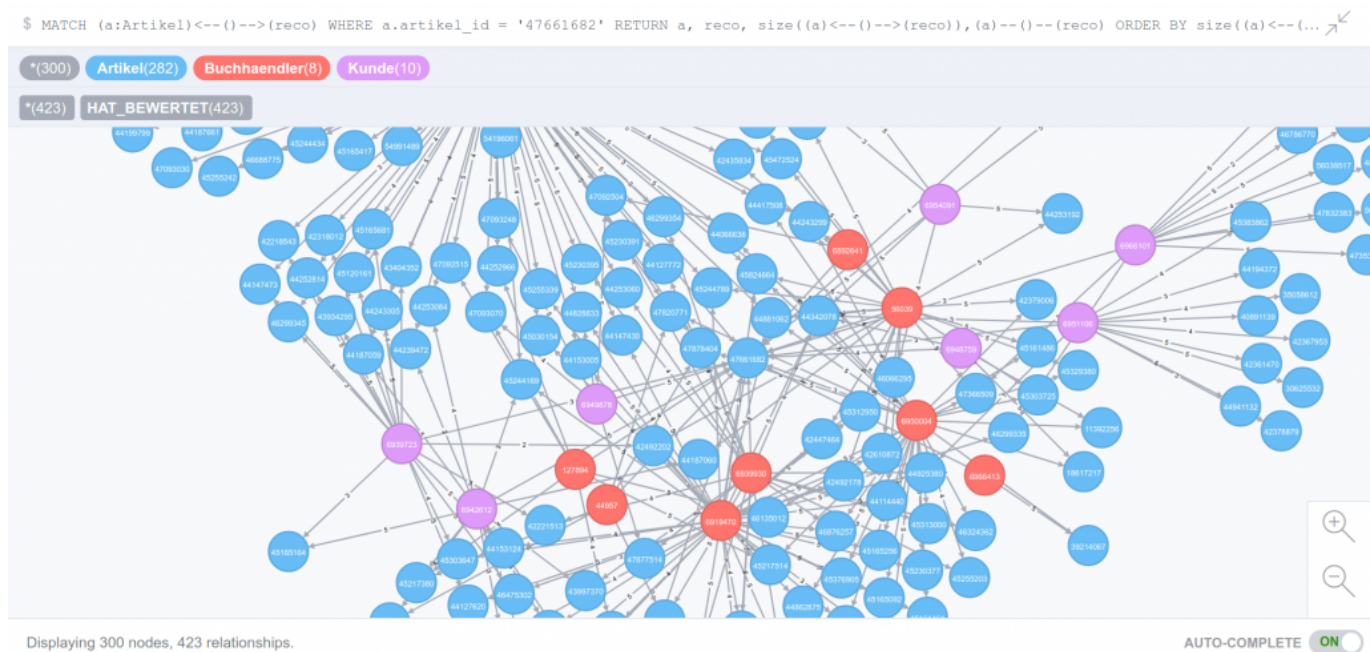
Auch das dritte Projekt hat zwei Themen (technologisch und fachlich) miteinander kombiniert. Neben dem Aufbau von technischen Erfahrungen im Bereich der [Graphendatenbanken](#), in diesem Fall „[Neo4j](#)“, sollte die Frage beantwortet werden, wie die zahlreichen Artikelbewertungen sowohl unserer Kunden als auch unserer Buchhändler miteinander in Beziehungen stehen und inwiefern man

daraus gute Empfehlungen ableiten kann.

Weiterhin wurde eine exemplarische Spring Anwendung hochgezogen, um eine Anbindung an diese Datenbank zu prüfen.



Fazit: Das Team war beeindruckt, wieviel in so kurzer Zeit machbar ist. Die Fachobjekte „Kunde“, „Artikel“ und „Buchhändler“ wurden in diversen Darstellungen korreliert, die Ergebnisse sind plausibel. Das Reindenken in die proprietäre Abfragesprache war allerdings nicht ohne Tücken (aka lange Laufzeit). Ideen zu weiteren verwendbaren Datenquellen existieren. Hier lohnt es sich, weiter zu forschen!



Abschluss

Auf die Auswertung eines „Applaus Voting“ für das beste Projekt und damit einhergehendes Schulterklopfen und Preise wurde einhellig verzichtet. Die Erfahrungen allerdings haben die Teilnehmer gerne mitgenommen.

Das verwendete GitLab bleibt für Interessenten erstmal bestehen, die Links findet ihr im Wiki.

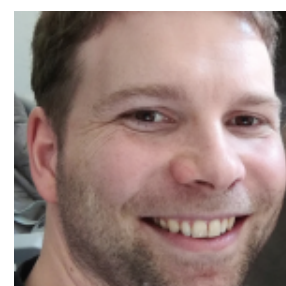
Als Verbesserungsvorschlag wurde das Thema „T-Shirts“ genannt, ich selber möchte noch „griffige Team Namen“ einwerfen – ich will aber natürlich nicht an einer eventuellen Eskalation bei der Namensfindung schuld gewesen sein. (Zwei Projekte hatten immerhin schon Namen mitgebracht.)

Die vorgeschlagenen und bearbeiteten Themen waren alle technologisch hochinteressant und gleichzeitig nah „am Business“, so dass sich ihre Bearbeitung für uns doppelt gelohnt hat. Ich freue mich schon, die ersten Ergebnisse in der Produktion bewundern zu können.

Müssen wir wiederholen!

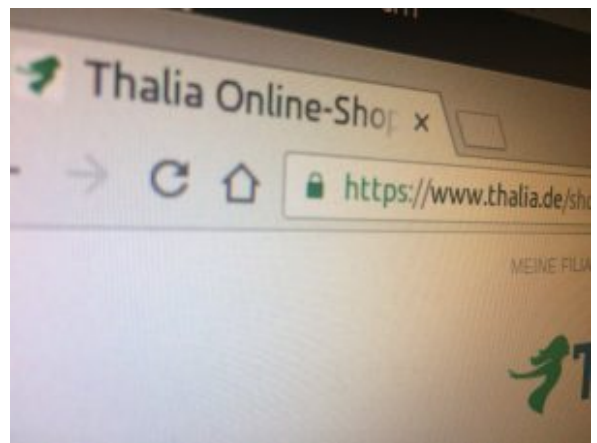


Jonas de Buhr



Martin Ernst

Alles auf „grün“ - Unser Weg hin zu „SSL für Alles“



Alles auf „grün“: hier ist die komplette Umstellung unserer Mandanten / Domain thalia.de gemeint

HTTPS ist nicht nur ein von Google bestätigter Rankingfaktor, auch einige Internet-Browser weisen inzwischen Webseiten, die noch nicht über HTTPS erreichbar sind, als unsicher aus. Aus Nutzersicht macht es also ganz klar Sinn Websites auf HTTPS umzustellen. Niemand möchte, dass sensible Daten von Dritten abgefangen werden können. Sieht ein Nutzer, dass eine Website vom Browser als unsicher eingestuft wird, überlegt er es sich zweimal, ob er einen Kauf abschließt oder seine Daten für einen Newsletter oder andere Dienste hinterlegt. Eine Einstufung als sicher kann ihn hingegen in seiner Kaufabsicht bestärken. Natürlich sind bei Thalia die Seiten, auf denen sensible Daten sind

bzw. erhoben werden, schon immer verschlüsselt. Wir haben uns jedoch bewusst entschieden alle Seiten zu verschlüsseln was neben den erwähnten positiven Faktoren für den Kunden auch die Performance optimiert als auch die Administration der Seite vereinfacht.

Auch im Bezug auf die **Sichtbarkeit** in Suchmaschinen ist die Nutzung von HTTPS anzuraten. Schon seit August 2014 ist HTTPS als Rankingfaktor bei Google etabliert. Seit Ende 2015 versuchen sämtliche Suchmaschinen automatisch Websites unter HTTPS aufzurufen. Ist eine Website unter HTTP und HTTPS aufrufbar, wird in der Regel die HTTPS-Version indexiert. Eine Studie des SEO-Tool-Anbieters Searchmetrics von 2015 bestätigt einen Zusammenhang zwischen der Nutzung von HTTPS und der Sichtbarkeit in Suchmaschinen einer Website. Laut der Studie haben Websites mit HTTPS tendenziell bessere Rankings.

Doch nicht nur die Sichtbarkeit und / oder Indexierung von Inhalten ist maßgeblich für eine erfolgreiche Umstellung auf HTTPS. Unsere gesamte eCommerce Plattform unterliegt regelmäßigen PCI ([Payment Card Industry Data Security Standard](#)) Audits. Mithilfe dieser extern beauftragten Audits wird sichergestellt, dass sämtliche Zahlungsmodalitäten (Visa, Paypal, etc.), sowie Verfahren innerhalb unserer eCommerce Plattform, die [Anforderungen und Regeln des PCI](#) erfüllen. Einer der wichtigsten Punkte ist dabei die Absicherung sämtlicher Kommunikation zwischen Plattform, Endbenutzer sowie Drittanbieter. Dieses ist für uns ein Baustein um Kundendaten maximal zu schützen.

Es ist ziemlich sinnvoll SSL auf allen Webseiten zu haben. Lasst es uns angehen! Aber nicht ohne die Fachabteilungen.

Für die Umsetzung haben wir uns bewusst für den Wasserfall-Ansatz entschieden. Gestartet mit dem Thema „SSL für Alles“ sind wir im März 2016 mit einer großen Kickoff Veranstaltung, bei der alle Fachbereiche (insgesamt 12), die Berührung mit der eCommerce Plattform hatten, eingebunden waren. Auch wenn es schnell und einfach möglich gewesen wäre auf dem Apache ein HTTPS-Forwarding einzurichten, ohne die einzelnen Fachbereich wäre es auf keinen Fall gegangen. Der Shop wäre dann zwar technisch auf HTTPS gewesen, jedoch hätten sich die

Kollegen gefragt, wo plötzlich die Umsätze hin sind.

Ausgehend von der großen Kickoff Veranstaltung begann anschließend die erste Planung der für die Umstellung notwendigen Ressourcen. Dabei kamen unglaubliche Zahlen zutage: 195 PT und 31.09.2016. So viel schon mal vorweg: das Datum konnten wir nicht halten ☐

„Vorbereitung & Planung ist alles“...

...hat mal ein schlauer Mensch gesagt und lag damit verdammt richtig. Doch auch eine noch so genau Planung und Vorbereitung, in der man(n) denkt, auch wirklich an alles, absolut alles gedacht zu haben, relativiert sich, wenn die Umstellung naht und der „Schalter umgelegt“ wird.

Es versteht sich von selbst, dass eine solche Umstellung von HTTP auf HTTPS nicht auf den produktiven Systemen „per Hand“ durchgeführt werden kann und darf. Zur Absicherung der Plattform und aus Gründen der Stabilität sowie aus Sicherheitsgründen haben wir uns dafür entschieden die bisherige produktive Webstrecke bestehend aus diversen Webservern, Loadbalancern und Firewalls nicht zu verändern. Stattdessen haben wir parallel eine komplett neue Webstrecke mit den eben genannten Komponenten aufgesetzt. Mithilfe dieser sind wir in der Lage die Umstellung schnell, sicher und ohne Ausfall zu planen und im Vorfeld testen.

Mit der Bereitstellung einer dedizierten Webstrecke — bestehende aus neuen IP Adressen, frischen Webservern auf Basis von Apache — sowie einer Vielzahl von Erweiterung / Änderungen an Firewall sowie Loadbalancer haben wir die Möglichkeit geschaffen, diese neue Strecke mit Anbindung an das produktive Backend ausgiebig zu testen und weiter zu optimieren, ohne dabei den eigentlichen produktiven Traffic in irgendeiner Weise negativ zu beeinflussen. Eine gute Ausgangsbasis, wenn man bedenkt, dass wir durch den Aufbau einer dedizierten bzw parallelen Strecke jederzeit hin und her schalten könnten.

Unsere Domain thalia.de sowie die dazugehörigen (virtuellen) Webserver sind nicht nur der Endpunkt für den Einstieg in den bereits genannten Online-Shop. Vielmehr bietet thalia.de eine Vielzahl von Diensten für die

unterschiedlichsten Geräte bzw. Konsumenten — wie beispielsweise OAuth, API, Partner-Integration, Mobile Apps (XCA), Hörbuch Downloads, etc... — und natürlich auch sämtliche Microservices, die in den letzten Monaten durch die einzelnen Teams bereitgestellt worden sind. Nicht zu vergessen sind hier auch die Abhängigkeiten zu externen Partnern, wie Paypal, Payback, Visa, usw., um nur einige zu nennen.

Die Umstellung ist nicht nur ein simples HTTPS-Forwarding auf dem Apache

Und spätestens an dieser Stelle dürfte jedem klar sein, dass eine scheinbar triviale Anforderung (HTTP zu HTTPS) weitreichende Auswirkungen haben kann. Daher ist es umso wichtiger das Gesamtbild aller Funktionalitäten und Abhängigkeiten vor Augen zu haben. Hand auf's Herz: Ist dazu wirklich irgendjemand in der Lage? Die Antwort darauf ist so nüchtern wie erschreckend: Nein. Aus der Theorie wissen wir, dass eine oder maximal eine handvoll Personen einen sehr guten Überblick haben. Der Rest *weiß* einfach, an wen man sich wenden kann und muss, wenn der Fall eintritt. Aus diesem Grund haben wir uns dafür entschieden alle wichtigen Vertreter der einzelnen Services (Shop-Management-System, App Services, IT Betrieb,) für den Tag der Tage in einem Raum zu vereinen. Die Umstellung ist ein Team-Erfolg.

SSL und SEO - so passt beides zusammen

Um eine möglichst Suchmaschinen-konforme Umstellung zu erreichen, mussten die folgenden Punkte zwingend eingehalten werden. Wie gut, dass wir dafür unseren Spezialisten am Tag der Umstellung an Board hatten. Eine zentrale Rolle spielt dabei die Google Search Console (ehemals Google Webmaster Tools). Hier ist es unbedingt notwendig, direkt nach der Umstellung auch die neue HTTPS-Version der Webseite einzurichten.

Umstellung auf HTTPS: Die wichtigsten Tipps

- Alle alten URLs sollten mit einem 301-Redirect auf die HTTPS-Variante weitergeleitet werden
- Auch Canonical Tags sollten in die SSL-Variante umgeschrieben werden
- Sofern bei den internen Links keine relativen Links verwendet wurden,

sollten diese ebenfalls umgebaut werden, um unnötige Redirects zu vermeiden

- Die robots.txt Datei sollte auch via HTTPS erreicht werden
- Es sollte sichergestellt sein, dass alle internen und externen Ressourcen (Bilder, Scripte, CSS etc.) über HTTPS laden – ansonsten werden im Browser für den Nutzer abschreckende Warnungen angezeigt
- In der [Google Search Console](#) sollte eine neue Property für die HTTPS-Version der Seite angelegt sein
- Die Sitemap sollte neu erstellt und bei der Google Search Console eingereicht werden
- [HTTP Strict Transport Security \(HSTS\)](#) sollte eingeführt werden: Dieser Sicherheitsmechanismus schützt HTTPS-Verbindungen vor der Aushebelung der Verbindungsverschlüsselung durch downgrade-Attacken

„Harry, leg den Schalter um !“

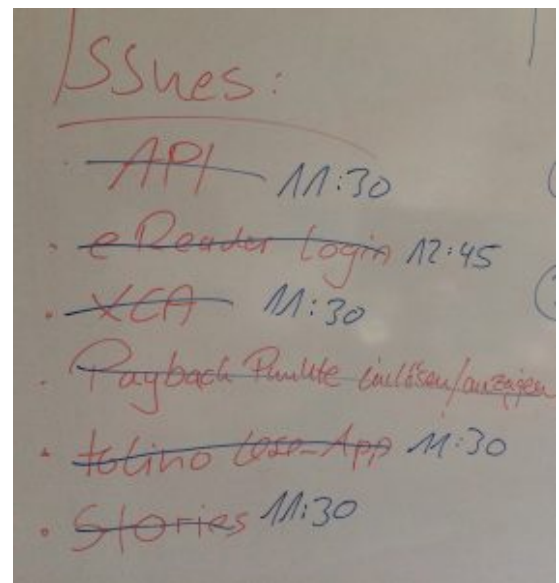
Eine Umstellung in der Nacht kam für uns nicht in Frage. Wir sind mutig und haben ein Top-Team welches am Morgen deutlich frischer ist. Ein beliebtestes Ritual darf an einem solchen Tag jedoch nicht fehlen: ausgiebig gemeinsam Frühstücken.



Käse, Wurst, Lachs, Mett und 40 Brötchen waren es zu Beginn :-)

Gut gestärkt geht es dann in die letzte Vorbereitungsphase, in der alle anwesenden Teilnehmer(/-innen) noch einmal alle notwendigen Punkte durchgehen. Für die vorliegende Umstellung von HTTP auf HTTPS haben wir uns für einen klassisch-moderaten Ansatz entschieden: Anstatt Änderungen im DNS zu einem gegebenen Zeitpunkt durchzuführen und darauf zu warten, dass diese

Änderungen weltweit propagiert sind (Notiz am Rande: Ein TTL von 5 Minuten kann auch sehr sehr lang sein), entschieden wir uns für eine Anpassung bei unserem vorgeschalteten DDoS-Schutz. Mit Hilfe des dazugehörigen Webportals kann im Handumdrehen der Upstream (in diesem Fall unser Endpunkt) aktualisiert werden, ohne dabei auf TTL oder sonstiges warten zu müssen, denn diese Art von Änderungen werden sofort aktiv. Hinzu kommt der charmante Vorteil, dass wir jederzeit und auf direktem Wege wieder zur ursprünglichen Konfiguration hätten zurück kehren können. Das passt und befriedigt auch die verbleibenden kritischen Stimmen.



Auszug aus unserem Whiteboard vom Tag der Umstellung: Störungen sowie die Lösungszeit sind hier vermerkt.

Wer jetzt erwartet hat, dass diese Umstellung ohne irgendwelche „Ruckler“ oder sonstigen Seiteneffekte vollzogen worden ist, den müssen wir hier enttäuschen. Auch Thalia kennt Murphys-Law. Es gibt immer Punkte, die zuvor nicht berücksichtigt werden konnten oder schlichtweg auf der Strecke geblieben sind. Doch eine gute Vorbereitung heißt auch, genau diese Punkte in kürzester Zeit anzugehen sowie diese Probleme aus dem Weg zu räumen. Denn hey, dafür haben wir doch „alle Mann an Board bzw. in einem Raum eingesperrt“.

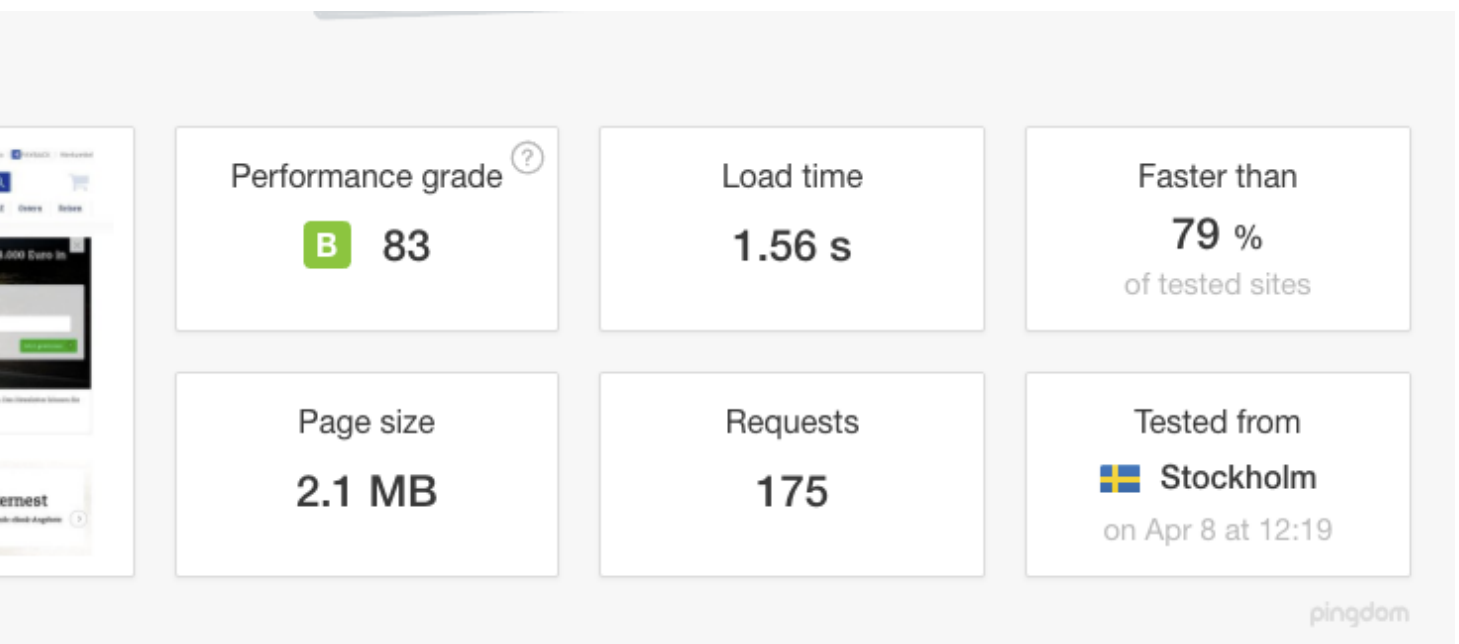
Bereits Wochen zuvor haben wir in (in teils penetranter Art & Weise) sämtliche Bereichen, Abteilungen sowie interne und externe Partner auf diese

Umstellung sensibilisiert, wodurch genügend Raum und Zeit gegeben worden ist, sämtliche Probleme zu klassifizieren, zu adressieren und schnellstmöglichst lösen zu können.

Fazit & Lesson's Learned

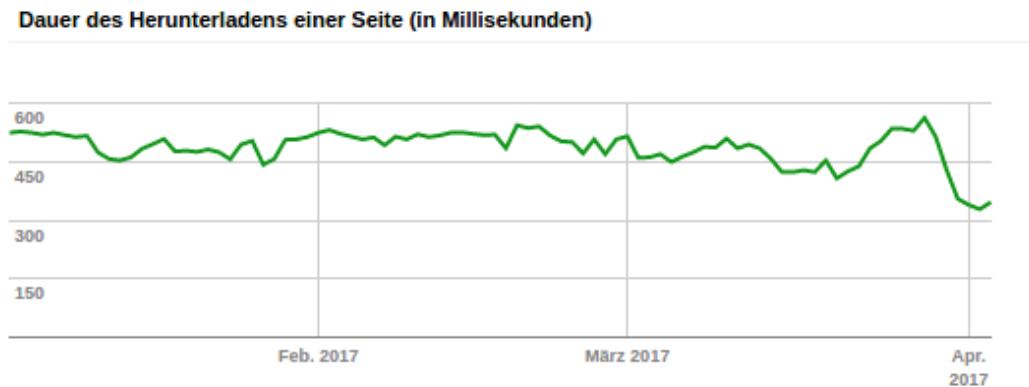
Unsere Vorgehensweise sämtliche Stakeholder und notwendigen Ressourcen an einem (physikalischen) Ort zu bündeln (bzw einzusperren) erwies sich wieder einmal als sehr nützliche Methode / Methodik, denn alle gemeldeten Problemchen konnten während des normalen Tagesbetriebs gelöst werden. Dank unserer ASB (=Automatische Service Bereitstellung) konnte ein Hotfix noch am gleichen Tag erstellt, getestet und in Produktion ausgerollt werden.

Die Umstellung auf HTTPS (SSL/TLS) hat darüber hinaus noch weitere Vorteile: Wir sind ab sofort in der Lage [HTTP/2](#) Verbindungen zu unterstützen, welches eine effizientere Nutzung der Netzwerk-Ressourcen und kürzere Latenzen verspricht (und auch einhält). Verbesserungen sollen aber auch eine Kompression der Header der IP-Pakete sowie Multiplex- und Push-Techniken bringen. Browser können mittels [Multiplexing](#) mehrere Messages gleichzeitig über eine Verbindung beziehen und Server können Elemente im voraus, also ohne Anfrage des Browsers pushen. Dies wird umso deutlicher, wenn wir uns vor Augen führen, wie viele Bilder beim Aufruf der sämtlicher Seiten geladen und übertragen werden müssen: HTTP/2 erweist sich hier als bemerkenswert effizient, da sämtliche Anfragen parallel und mit nur einer Verbindung übertragen werden. Dies wiederum hat zur Folge, dass die Ladezeit erheblich verbessert werden konnte.



stellung auf HTTPS inkl HTTP/2 (Quelle: tools.pingdom.com)

weiterer Vorteil für den Client, über den Sicherheitsgewinn hinaus: Durch die entfallende Umleitung verkürzen sich die Antwortzeiten.



Auch Google indiziert die Domain www.thalia.de nun deutlich schneller als zuvor (Unsere Änderung wurde Ende März in Produktion ausgerollt)

Ausblick

Der erste Schritt in die richtige Richtung ist vollbracht. In den kommenden Wochen werden wir das Verhalten sowie die Performance weiter im Auge behalten und ständig weiter optimieren, um unseren Kunden und Besuchern eine best mögliche User Experience zu präsentieren. In Sachen Performance und User Experience stehen unsere Edge-Provider (MyraCloud Security sowie Akamai) ganz oben auf der Liste der Agenda. Hier gilt es zu bewerten, inwiefern wir das Caching weiter verfeinern können.

Tipps & Tricks für JIRA als SCRUM Board

1. Einleitung

Die Software Entwicklung bei Thalia ist agil und wird von mehreren Teams per SCRUM durchgeführt. Zur generellen Verwaltung der einzelnen Aufgaben wird in der gesamten IT das Ticketsystem JIRA von [Atlassian](#) verwendet. Dieser Artikel soll einige Erfahrungen aus einem unserer Teams weitergeben, wie sich der SCRUM Alltag mit Hilfe von JIRA organisieren lässt. Er richtet sich dabei vor allem an Anwender, die bereits erste Erfahrungen mit SCRUM und JIRA haben.

Im folgenden sind einige Anregungen, wie ein SCRUM Team Daily Meetings digital abhalten kann, der PO beim Planning durch JIRA unterstützt wird bzw. die Durchführung der Meetings organisiert werden kann.

2. Daily

Für die Dailys verwendet das Team das Standard JIRA SCRUM-Board mit der Ansicht des aktiven Sprints. Dieses wird auf einem großen 40" Monitor dargestellt. Die einzelnen Team-Mitglieder können das Board während des Meetings direkt per Trackpad bedienen.

Vorteile:

- Das digitale Board wird den ganzen Tag angezeigt und gibt damit einen permanenten Überblick
- Einfache Zusammenarbeit
 - Tickets können während der Arbeit vom Platz aktualisiert werden.
 - Standortübergreifende Teams haben ein einheitlichen Blick auf das Board. Das war vorher bei der Verwendung von Pinnwänden schwierig.
 - Flexible Zusammenarbeit auch bei Teammitgliedern im HomeOffice
- Interaktion im Daily bleibt erhalten
 - Das Trackpad wird als „Staffelstab“ an den jeweiligen Redner übergeben
 - Tickets können schnell und einfach direkt während des Daily in andere Statusspalten geschoben werden.

- Tickets können schnell und einfach direkt während des Daily anderen Personen zugewiesen werden.

3. Planung

Bei Thalia hat jedes Entwickler-Team sein eigenes JIRA-Projekt. Die Planung für unser Team erfolgt daher innerhalb unseres eigenen Projekts und dem dort zur Verfügung gestellten Backlog. In diesem werden die Tickets des Teams erfasst und verwaltet.

The screenshot displays the JIRA interface for 'Team RED - Ideenspeicher'. At the top, there's a navigation bar with 'Startseite', 'Projekte', 'Vorgänge', and 'Boards'. A search bar and an 'Erstellen' button are also present. The main section is titled 'Backlog' and shows a list of tickets. The tickets are organized into columns: 'Ideen', 'Neu', 'Frage', 'DoR', 'Refinement', 'Offen', 'In Work', '!Sprint', 'UC', 'BW', 'BH', and 'CMS'. The 'Backlog' column contains 295 items. Each ticket entry includes a JIRA ID (e.g., RED-332, RED-389, RED-145, RED-455, RED-39, RED-364, RED-654, RED-648, RED-626, RED-429), a description, and an assignee (e.g., LieblingsBuchhändler, Thardis-Optimierung, AreaManager). The tickets are color-coded by priority (e.g., yellow for low, orange for medium, red for high).

Grundsätzlich lässt sich mit dem Backlog und einzelnen Sprints mit JIRA ohne weitere Veränderung arbeiten. In der alltäglichen Arbeit zeigt sich aber, dass die Bedürfnisse von PO und SCRUM Team in einigen Punkten abweichen und die Arbeit durch kleine Anpassungen erleichtert werden kann.

Das Backlog wird im Laufe der Zeit mit verschiedensten Tickets befüllt, die ab einer bestimmten Menge das SCRUM Team überfordern können. Es geht der Überblick verloren, welche noch aktuell sind oder eventuell neu geschätzt werden müssen. Daher ist eine weitere Unterteilung unter Umständen hilfreich.

3.1 Stadien eines Tickets

Eine genauere Betrachtung der Tickets zeigt, dass diese während ihrer Lebenszeit verschiedene Stadien durchlaufen. Hierbei wird deutlich, dass es unterschiedliche Standpunkte bzw. Schwerpunkte gibt, was genau der Status eines Tickets ist.

Aus Sicht des Teams

Das Team hat vor allem die Bearbeitung des Tickets im Blick. Am Anfang ist das Ticket noch *Offen* und wird dann *Bearbeitet*. Von dort wandert es über ein *Review* in den *Test* bis es nach einer *Abnahme geschlossen* werden kann.

Bearbeitung: *Offen > In Bearbeitung > [Review] > Test > [Abnahme] > Geschlossen*

Aus Sicht des PO

Die Arbeit des PO fängt bereits lange vor der Bearbeitung durch das Team an. Es kommen neue Tickets ins Backlog, die eventuell auch von außerhalb des Teams kommen können und daher als *Neu* kenntlich sein müssen. Diese können ins Backlog aufgenommen oder abgelehnt bzw. verändert werden. Alle Tickets, die im Backlog sind, müssen zur *Feinplanung* an das Team gegeben werden und geschätzt werden. Dabei können sie als *Sprint tauglich* (Definition of Ready erfüllt) bewertet werden. Alle Tickets, die den DoR-Status haben, stehen damit für einen nächsten Sprint zur Verfügung.

Planung: *Neu > Ins Backlog aufgenommen > Feinplanung > Sprint Tauglich > In Sprint übernommen.*

Zwischenfazit

Es gibt also zwei Ebenen, die **Bearbeitung** und die **Planung**. Da Tickets aus einem laufenden Sprint auch wieder ins Backlog wandern können, existieren unter Umständen Tickets im Backlog, die bereits bearbeitet wurden und einen entsprechenden Status haben. Das zeigt, dass diese beiden Zustandstypen parallel existieren können. Daher haben wir uns dafür entschieden, die Planungszustände nicht über den Status abzubilden, sondern als parallelen Zustand zu implementieren.

Im Folgenden wird daher gezeigt wie sich mit einfachen Mitteln in JIRA diese beiden Stränge parallel verwalten lassen.

3.2 Organisation

Die Organisation der Tickets findet in einem einzigen Projekt Backlog statt. **In diesem Backlog erfolgt die Zuordnung zu den einzelnen Phasen der Bearbeitung über den Ticket Status. Die Zuordnung zu den einzelnen Phasen der Planung erfolgt durch Stichwörter.**



Eine Aufteilung kann recht einfach durch „virtuelle Trennlinien“ in Form von Tickets mit Titeln wie „=====“ visualisiert bzw. organisiert werden. Da JIRA aber einige Eigenheiten beim Sortieren von Tickets aufweist, können sich schnell Tickets in andere Bereiche schummeln. Durch die Verwendung von Stichwörtern lässt sich das vermeiden.

Stichwörter

Das Backlog wird durch die Stichwörter in einzelne Planungsphasen aufgeteilt. Diese helfen bei der Organisation und unterstützen die zielgerichtete Vorbereitung der Scrum Meetings.



Ursprung	Ziel	Stichwort	Wer macht Was ?
Neu	Backlog	+ RED-Backlog	Der PO prüft alle neuen Tickets und verschiebt diese ins Backlog.
Backlog	Refinement	+ RED-Refinement	PO und TL bereiten im Rahmen der Backlog Pflege das Refinement Meeting vor und verschieben ausgewählte, kurzfristig relevante Tickets ins Refinement. Der Refinement Bereich soll nach jedem Meeting wieder aufgeräumt werden.

Ursprung	Ziel	Stichwort	Wer macht Was ?
Refinement	DoR	+ RED-DoR - RED-Refinement	Das Entwickler Team übernimmt alle vollständig geschätzten und geklärten Tickets in das Sprint Backlog. Aus diesem werden die folgenden Sprints im Planning Meeting befüllt.
Refinement	Refinement	+ RED-Question	Sollte das Entwickler Team ein Ticket nicht schätzen können, wird dies zur Klärung zurückgestellt und markiert.
Refinement	Backlog	- RED-Refinement	PO und TL räumen im Anschluss an das Refinement Meeting den Bereich wieder auf und versuchen die offenen Fragen zu klären

3.3 Boards

Für die alltägliche Arbeit stehen mehrere Boards zur Verfügung. Das Entwickler-Team nutzt dabei für die tägliche Arbeit nur das SCRUM Board, während der PO überwiegend im Ideenspeicher unterwegs ist. Als Schnittstelle zwischen beiden dient das Refinement Board.



Board	Beschreibung
Ideenspeicher	Das Board zeigt alle Tickets, die im gesamten Projekt existieren. Es kann über Filter eingeschränkt werden.
Refinement	Es werden nur die zu besprechenden Refinement Tickets angezeigt.
SCRUM Board	Es werden nur die aktuell geplanten Sprints angezeigt und alle Tickets, die reif für die Umsetzung sind.

3.4 Kleine Helfer im Alltag

Filter

Vor allem der PO steht vor der Herausforderung das Board stets aktuell zu halten, das Team mit priorisierten Aufgaben zu versorgen und einen Überblick über neue und abgeschlossene Tickets zu behalten. Das führt zu häufigen Wechseln zwischen Planung und Sprint Übersicht.

Um das zu vereinfachen, haben wir im Ideenspeicher-Board eine Reihe von Filtern eingerichtet, durch die es je nach Bedarf weiter eingeschränkt werden kann. Dadurch kann die Sichtbarkeit in den einzelnen Boards geprüft werden, ohne dass die Boards gewechselt werden müssen.

Filter	Regel	Zweck
Neu	Label not in (Backlog)	Zeigt alle neuen Tickets, die noch nicht geprüft wurden
Idee	Label not in (Refinement, DoR)	Zeigt nur Ideen, die noch nicht zur Umsetzung ausgewählt worden sind
Frage	Label = Question	Zeigt alle Tickets mit offenen Fragen
Refinement	Label = Refinement	Zeigt nur Tickets, die für das Refinement vorgesehen sind
DoR	Label = DoR	Zeigt alle Tickets im Sprint Backlog
Offen	Status = Open	Zeigt alle offenen Tickets
In Work	Status != Open	Zeigt alle nicht offenen Tickets
!Sprint	Sprint != OpenSprint()	Blendet alle Tickets aus, die einem Sprint zugeordnet sind

Farbkennzeichnung der Tickets

Ein weiterer Aspekt hat sich in der Sichtbarkeit des Planungsstandes der Tickets gezeigt. Für den PO ist es wichtig zu wissen, welche Tickets beim Team liegen, neu sind oder wo noch Fragen offen sind. Um dafür nicht jedes Mal die Filter benutzen zu müssen, lässt sich der Status auch als Farbkennzeichnung realisieren.

Farbe	Filter	Beschreibung
orange	(labels not in RED-Backlog) or labels is Empty	Neue Tickets
rot	labels = RED-Question	Tickets mit Rückfragen aus dem Team
blau	labels = RED-Refinement	Tickets, die im Refinement Board für das nächste Refinement anstehen
grün	labels in (RED-DoR)	Tickets, die reif für die Umsetzung sind

Meetup@Thalia: Vortrag Johannes Mainusch: „Otto.de - wie die Titanic den Eisberg verfehlte“

Am Donnerstag war es endlich so weit, wir traten aus dem Schatten der internen Fachvorträge ins Rampenlicht der offenen MeetUp Kultur.

Die offene Kultur war für diese erste Veranstaltung auch prägend. Nicht nur, dass sie inmitten unserer Räumlichkeiten stattfand, sie war auch ein Beispiel für die enge Zusammenarbeit und Vernetzung der Branche.

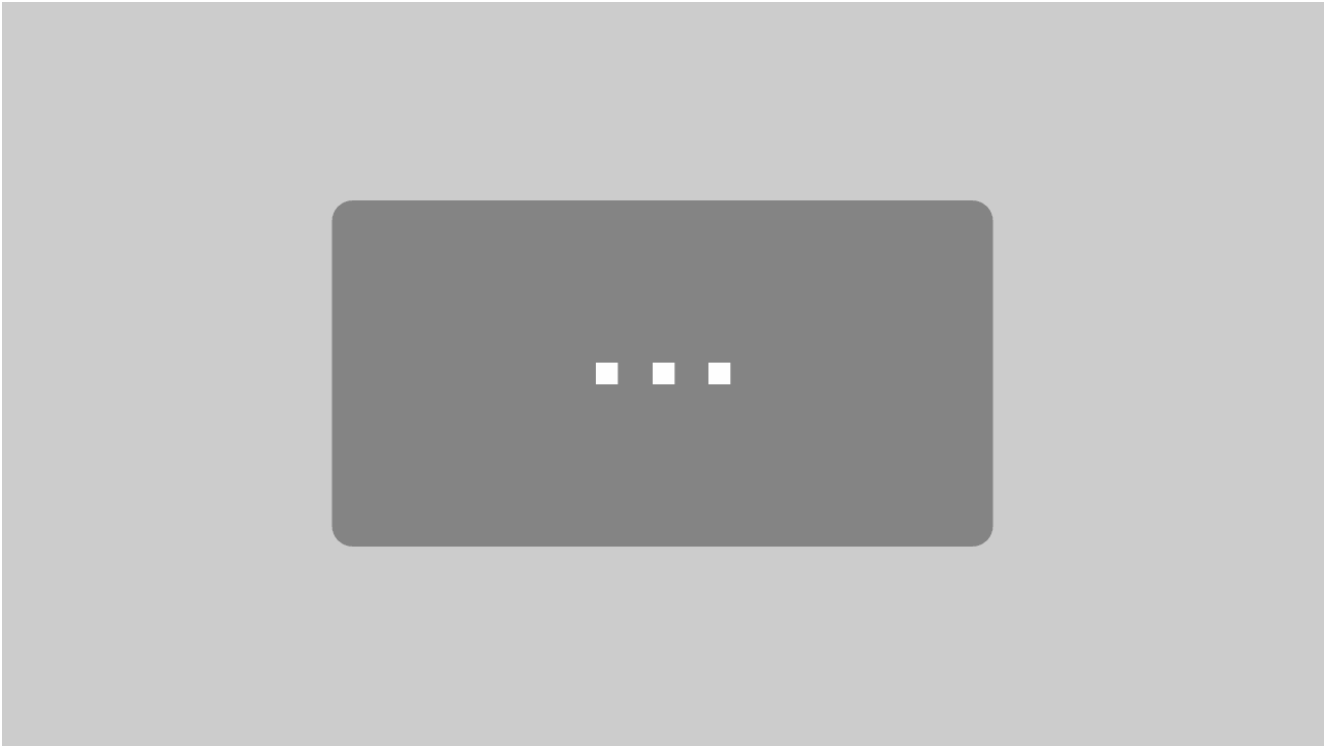
So konnten wir Johannes Mainusch dafür gewinnen seinen Vortrag „Warum die Titanic den Eisberg verfehlte“ in unserem Haus erneut zu präsentieren. Der Vortrag konnte bereits im Rahmen der Keynote der Continuous Lifecycle Konferenz begeistern.



Vortrag durch Johannes Mainusch

Der Vortrag wurde neben den Berliner Kollegen, auch von Kollegen aus Münster und Hagen, sowie einigen externen Gästen besucht. In rund zwei Stunden bekamen wir so einen spannenden Einblick, wie otto.de die langsame und schwer veränderbare Plattform modernisieren konnte. Aus einer 10 Jahre alten Intershop Installation mit mehr als 750.000 modifizierten Code Zeilen wurde ein Musterbeispiel für agile Entwicklung. Firmen wie Quelle und Neckermann waren unter ähnlichen Bedingungen in der Insolvenz gelandet. Otto.de konnte dagegen mit dem 2012 gestarteten Lhotse- Projekt zu einer der größten E-Commerce Plattformen in Europa wachsen. Welche Entscheidungen letztendlich dafür notwendig waren und was ein Unternehmen befähigt mehr als 340 Releases pro Woche Live zu bringen, schilderte Johannes Mainusch anschaulich und untermalt von vielen Beispielen.

Wer Interesse hat, kann sich den vollen Keynote Beitrag auch bei Youtube ansehen:



Mit dem Laden des Videos akzeptieren Sie die Datenschutzerklärung von YouTube.

[Mehr erfahren](#)

[Video laden](#)

☐ YouTube immer entsperren



Johannes Mainusch

ist seit April 2016 selbstständig bei der kommitment GmbH & Co. KG tätig. Vorherige Stationen waren unter anderem Lufthansa, Xing, Otto.de bis hin zur E-Post. Author des Titels [„Scrum mit User Stories“](#)



Start mit einem Pizza Buffet



Vortrag durch Johannes Mainusch



Diskussionsrunde nach dem Vortrag