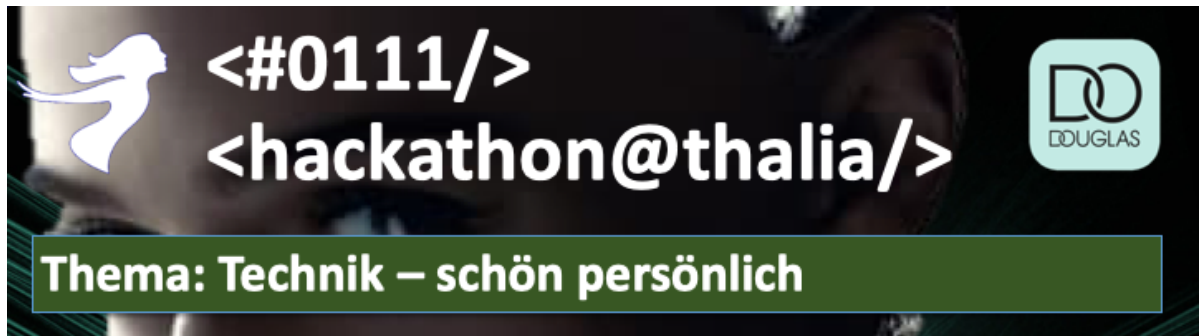


Hackathon 2019 in Berlin



Unseren **3. Hackathon** haben wir im Juni 2019 zusammen mit unserem B2B-Partner Douglas ausgerichtet. An zwei Tagen pitchten wir mit 25 Teilnehmern Ideen, probierten neue Methoden und Techniken aus und teilten unser Wissen. Das alles gut versorgt, mit jeder Menge Spass und Motivation. Mit dem Thema **Personalisierung** haben wir den Hackathon dieses Mal inhaltlich geschärft. Das Publikum hat nach den **Abschluss-Präsentationen** zwei Projektideen prämiert und die Gewinner durften sich über tolle **Preise** freuen. Über die Themen und Eindrücke möchten wir euch über unseren Techblog teilhaben lassen.

Thalia B2B-Entwicklung in Berlin

In Berlin entwickeln wir seit 2014 für B2B-Partner digitale Produkte im Retail-Umfeld. In der kontinuierlichen Produktentwicklung sind uns agile Arbeitsweisen und stabile Teams sehr wichtig. Unser Schwerpunkt liegt in der Frontend-Entwicklung für mobile Geräte auf Basis von Android und iOS.



Teilnehmer und Partner beim gemeinsamen Abschlussfoto - Yeahh! ☐



■ Vanessa Stütze – Managing Director Ecommerce & Omnichannel Douglas Group begrüßte die Teilnehmer mit einer Videobotschaft

Personalisierung - Unser Schwerpunktthema

Mit unseren Teams in Berlin entwickeln wir seit mehreren Jahren erfolgreich Apps für unseren B2B-Partner Douglas. In Absprache mit Douglas haben wir den

inhaltlichen Schwerpunkt auf das Thema **Personalisierung** gelegt und uns überlegt, wie wir diesbezüglich Projektideen generieren können. Die Teilnehmer haben wir vorab mit einer **Lightning-Demo Session** eingestimmt, in der aktuelle Apps bzw. Websites vorgestellt wurden, die inspirierende Personalisierungsfunktionen anbieten.

Lightning Demos

Eine Methode mit der sich ein Team mit Produkten und Services für die Konzeptphase inspiriert. Wird in der Methodik „Design Sprint“ angewendet.

Ablauf

- Jeder Teilnehmer recherchiert selbst 2-3 inspirierende Beispiele
- Jeder Teilnehmer präsentiert die Beispiele
- Die Kernideen werden auf Klebezetteln geschrieben und an ein Board angebracht

Am Ende hat das Team 10-20 Ideen zu dem Thema gesammelt. Es geht darum von allen Teilnehmern die besten persönlichen Ideen aufzunehmen und dabei nicht zu diskutieren.

<https://www.sessionlab.com/methods/lightning-demos>

<https://www.thesprintbook.com>

Die Projekte

Am Morgen des ersten 1. Tages stellten die Teilnehmer 6 Projektideen vor. Neben 5 technischen Vorschlägen aus dem Mobile-Umfeld hatten wir mit dem Design-Sprint-Projekt auch ein Methode aus dem UX-Umfeld am Start. So starteten wir nach dem Pitch mit 4 Teams in die Projekte.



Der Pitch – Nach den Vorstellungen wählten die Teilnehmer ihr jeweiliges Projekt



Team Mini Design Sprint - Solve Big Problem And Test New Ideas

Das Thema Personalisierung sind wir methodisch mit einem Mini Design-Sprint angegangen. Ein Design-Sprint ist **Konzeptionsmethode** für Produkte, die innerhalb von 5 Tagen mit dem Team durchgeführt wird. Da nur 2 Tage zur Verfügung standen, wurde das Format „kreativ“ komprimiert. Ziel der 2 Tage war das Kennenlernen der Methode und die Erstellung von interaktiven Prototypen für die Kundenapp unseres Partners. Bei diesem Non-Coding-Projekt stand also nicht die Programmierung, sondern die Methodik zur Generierung neuer Produktideen im Fokus.

Design Sprint

„The Design Sprint is a five-day process for solving problems and testing new ideas.“

Invented at Google by Jake Knapp, perfected with more than 150 startups at GV, then shared with the world in the bestselling book “Sprint - How To Solve Big Problems and Test New Ideas in Just Five Days”.

“On Monday, you make a map of the problem. On Tuesday, each individual

sketches solutions. On Wednesday, you decide which sketches are strongest. On Thursday, you build a realistic prototype. And on Friday, you test that prototype with five target customers.”

<https://www.thesprintbook.com/>

<https://www.youtube.com/watch?v=K2vSQPh6MCE>

In den 2 ziemlich durchgeplanten Tagen hat das Team im ersten Schritt in **Experteninterviews** die **Challenges** zum Thema Personalisierung zusammengetragen und priorisiert. Die Interviews wurden mit 2 Personen aus den Bereichen Business, Marketing/Sales und Tech durchgeführt.

How we might ...

„Wie könnten wir persönlich Kunden ansprechen, ohne das es creepy wird“?

Es wurden ambitionierte 2 **Jahresziele** und mögliche **Hindernisse** auf dem Weg identifiziert. Auf dieser Basis wurden **Lösungskontext** und **Fokusgebiet** geschärft und das Team hat eine **Lightning Demo** Session durchgeführt. Anschließend hat jeder Teilnehmer eine **eigene Produktidee** ausgearbeitet und als **Paper-Prototype** in **3 Schritten** visualisiert. Zwei Ideen wurden nach intensiven Diskussionen ausgewählt, um **Flows** auszuarbeiten und final **interaktive Prototypen** zu erstellen.

Mit den Ideen „Find your Style“ und “Love me Tinder“ ging das Team in die Abschlusspräsentation. Der Hackathon-Designsprint war damit abgeschlossen. Die Phase der Nutzer-Validierung möchte das UX-Team im Nachgang zusammen mit der Product-Ownerin angehen.

Die 8 Teilnehmer waren von der Methode und den Ergebnissen begeistert. Unser UX/UI-Lead Nicolai hat den Design-Sprint sehr gut vorbereitet und moderiert. Das Team bestehend aus Designern, Product Owner, Entwickler, QA, Scrum Master und Team-Manager hat viele Perspektiven in die Konzeptphase eingebracht und so für unseren Kunden einen wertvollen Input für die Produktentwicklung geben können.

WORKSHOP

MONDAY

Define the
Challenge

Envision Long
Term Goal

Produce a mass
of solutions

Curate and vote on
best solutions

TUESDAY

Map the
User Flow

Design and build
the Prototype

Pitch Outcome



**"SOLVE BIG
PROBLEMS AND
TEST NEW IDEAS"**

From the guys who brought
you **Lightening Demos**

**MINI
DESIGN
SPRINT**



P E R S O N A L I Z A T I O N

Thalia AppTech
HACKATHON

Powered by **Douglas**

2019



Paper Prototypes

Team KRAS - Konferenz Raum Anzeige System

Mit der Projektidee sollte ein ganz praktisches Problem in unserem Office gelöst werden. Steht man vor einem unserer Meetingräume und möchte wissen ob oder ab wann dieser belegt ist, muss man erst eine Buchungsanfrage am Rechner durchführen, um an die Information zu gelangen. Da gehen schon mal ein paar Minuten ins Land. Das geht besser!

Die Informationen sollen mit einem kleinem Display an jedem Raum angezeigt werden. Darauf soll die aktuelle Tagesbelegung dargestellt werden.

Das Team hat sich für **Angular** als Frontendtechnologie entschieden. Die UI ist responsiv und kann auch auf kleinen **eInk-Devices** dargestellt werden. Neben der Darstellung der Rauminformationen hat das Team auch damit begonnen, eine Schnellbuchungsfunktion für den Raum zu implementieren. Die Web-UI wurde mit einer **Android-App** auf einem modifizierten **Tolino eInk-Reader** und auf einem Android-Tablet dargestellt. Aufgrund der deutlich langsameren Geschwindigkeit der eInk-Displays müsste die UI im nächsten Schritt weiter optimiert werden. Z. B. ist eine Alternative für das Scrolling durch Listen nötig.

Den Code hat das Team auf Github bereitgestellt:

Conference room assisting system

<https://github.com/jenszech/cras>

ePaper interface for Conference Room Assistant System (CRAS)

<https://github.com/jenszech/crasBadgeIt>

Neben der Web-UI hat das Projektteam auch eine **Arduino-basierte ePaper-Hardwarelösung** mit Mikrokontroller evaluiert. Die Lösung ist äußerst energiesparsam.

Die REST-API wurden über einen **Node.js-Server** realisiert, der auf einem **Raspberry Pi** läuft. Eine Herausforderung lag in der Anbindung des **Thalia Exchange**-Servers, der die Informationen zu den Räumen bereitstellt. Zusätzlich wurden Metadaten zu den Räumen wie z. B. vorhandene Meeting-Technik und Größe hinterlegt.

In den 2 Tagen konnte sich das Team gut in das umfangreiche Thema einarbeiten. Ein weiteres Learning war, dass für die Koordination der Subteams entsprechend Zeit eingeplant werden muss. Alle im Team waren hoch motiviert für ihre

Bereiche Lösungen zu erarbeiten und diese mit den anderen zu integrieren. Die erarbeitete Lösung ist nahezu fertig, um sie mit einem ersten Meetingraum zu testen.



 **<#0111/>**
<hackathon@thalia/>



UX Design *Angular*

Exchange *Node.js*

KRAS! *npm*

Konferenz Raum Anzeige System

wir
brauchen
DICH !!!

Microcontroller *eReader Hack*

ePaper



17. & 18.6.2019

mehr infos: <https://hackathon.thalia.de>
euer hackathon team: j.zech, d.okrent, s.wolff



Start 10:40 Uhr

Dauer	15 Min.	30 Min.	45 Min.	60 Min.	75 Min.
	90 Min.	105 Min.	120 Min.		

Jetzt buchen

■ [javascript:history.back\(\)](#)

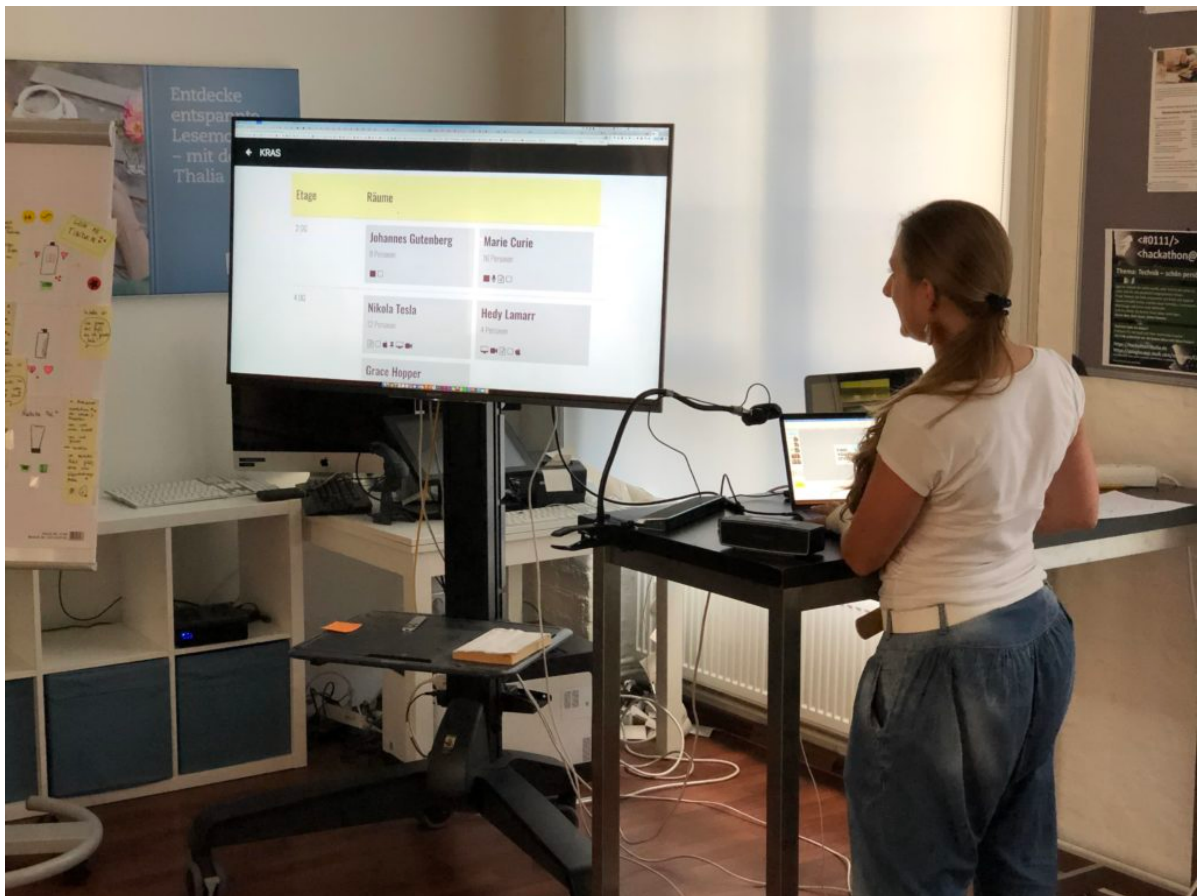
Raum: Ada Lovelace		Wednesday, 24.07. 10:49 Uhr
09:00 - 10:00	F R E I	
10:00 - 12:00	Neef, Johanna Hitze Planning	
12:00 - 14:00	F R E I	
14:00 - 15:00	Team OCS RED Tech Board	
15:00 - 21:00	F R E I	

■

← KRAS

Etage	Räume		
2 OG	Johannes Gutenberg 8 Personen  	Marie Curie 16 Personen   	
4 OG	Nikola Tesla 12 Personen     	Hedy Lamarr 4 Personen    	Grace Hopper 6 Personen  
3 OG	Ada Lovelace 10 Personen   	Alan Turing 4 Personen  	





Team Flutter Touch - We want to Flutter you!

Mit dem Thema Cross-Plattform Entwicklung unter Anwendung von **Google's Flutter** hat sich auch dieses Jahr ein Team beschäftigt. Flutter ist mittlerweile in Version 1.7 veröffentlicht. Neben iOS und Android existiert auch ein **Technology Preview für das Web**. Seit Ende 2018 ist Flutter production-ready und die Flutter Community wächst kontinuierlich. Das Team hatte 2 Teilnehmer die bereits Projekterfahrung mit Flutter hatten.

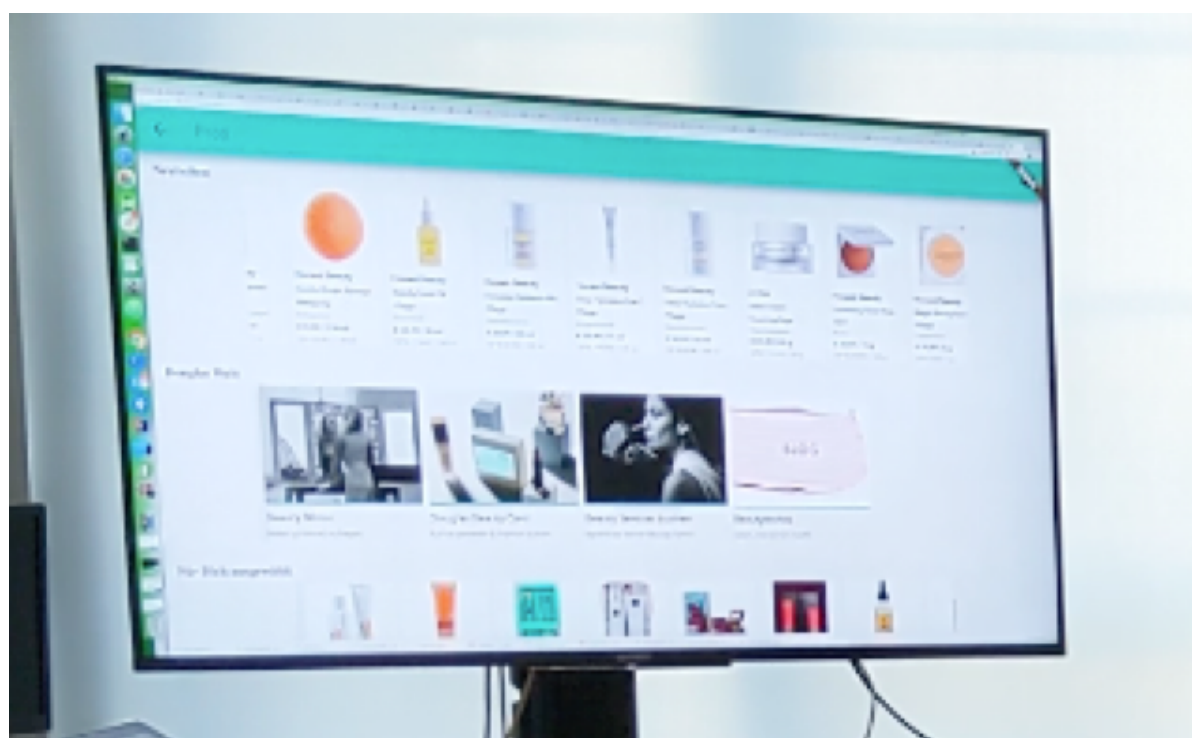
Unter dem Topic „**Feed-Configurator**“ hat das Team für die Kundenapp unseres Partners eine Individualisierungsfunktion geschaffen, um die Produktbereiche wie „Für dich ausgewählt“, „Kategorien“ oder „Marken“ in der Reihenfolge zu ändern. Mit dem „**Tinder-Swipe für Produkte**“ Feature kann der Kunde im Tinder-Stil Produkte durch Wischen nach links oder rechts als interessant bzw. uninteressant markieren. Die Daten könnten später in eine Recommendation-Engine überführt werden, um bessere Produktvorschläge zu machen. Präsentiert wurden die Funktionen im **Web**, auf **iOS** und auf **Android**.

Das Tinder-Feature hat das Team sogar auf einem **Android-Wearable** (Uhr) im Emulator gezeigt. Großartig!

WE WANT TO **FLUTTER** YOU!



NEAREST RECRUITING STATION JUST AROUND THE CORNER!
THALIA HACKATHON 17./18. JUNI 2018 - MELDE DICH BEIM
TEAM FLUTTER TOUCH





Wearables - Wie können wir Wearables im Douglas Kontext einsetzen?

Das Team hat sich einen Usecase im Kontext Personalisierung überlegt. In diesem werden Produktempfehlungen als personalisierte **Push-Notification** auf dem **Wearable** dargestellt und sollen über **Smart-Actions** auf den Merktzettel oder in

den Warenkorb gelegt werden können. In der Filiale könnte man sich auch vorstellen die Produkte direkt über einen Paymentdienst wie Google Pay zu bezahlen.

Entwickelt wurden die Prototypen auf **iOS, Android und Tizen**. Über ein **Google Firebase**-Setup sollten die Push Notifications auf die Geräte gesendet werden. Die Anwendungen liefen jeweils in den Emulatoren der Plattformen. Die Produktdetails wurden von den Partner-APIs abgerufen und auf den kleinen Displays dargestellt. Die Entwickler waren davon überrascht, wie schnell sie auf den Wearables entwickeln können.







Unser Fazit

Das Feedback der Teilnehmer und unseres Partners Douglas war äußerst positiv. Den Hackathon auf 2 Tage zu erweitern, gab den Teams mehr Zeit sich inhaltlich intensiver mit den Themen zu beschäftigen und ihre Präsentationen vorzubereiten. Die Bandbreite der Themen war technologisch und inhaltlich groß und brachte den Teilnehmern viele interessante Einblicke in die Themen, mit denen wir uns vielleicht schon morgen beschäftigen werden.

Da auch Douglas-Mitarbeiter in Berlin dabei waren, konnten wir uns persönlich besser kennenlernen und austauschen. Die Projekte waren allesamt spannend und wir haben unser Ziel erreicht mit dem Schwerpunkt Personalisierung einen direkten Mehrwert in Form von **Prototypen**, **Methodik** und **Inspiration für Produktideen** zu schaffen.

Die Abschlussphase mit den Präsentationen und die Preisverleihung hat den Hackathon-Event gekonnt abgeschlossen. Wir freuen uns schon auf das nächste Jahr!

Und noch ein paar weitere Inspirationen ...



Team KRAS - 1. Preis



Team miniDesignSprint - 2. Preis





Möchtest du mehr über uns erfahren? Dann empfehle ich dir auch die folgenden Beiträge und Links:

[B2B-Entwicklung in Berlin](#)

[Hackathon 2018](#)

[Hackathon 2017](#)

Continuous Integration in der App-Entwicklung

Am Standort Berlin entwickeln wir für unseren B2B-Partner Douglas unter anderem die Kunden-App [2], [3]. Einhergehend mit dem Ausbau der App-Entwicklungsaktivitäten haben wir in den letzten Monaten den CI-Ansatz überarbeitet. Ein zusätzliches Team sollte am selben Produkt – der KundenApp – mitarbeiten und die App sollte öfter veröffentlicht werden. Mit Methoden und Tools aus dem Bereich Continuous Integration wollten wir dafür sorgen weiter zuverlässig und mit hoher Qualität zu liefern. Und das natürlich automatisiert. Neben der Technik geht es auch um die Teams. Wie sind sie vorgegangen und welche Hürden haben sie genommen.

Komplexität und Feedback

Je mehr Personen gleichzeitig an einem Produkt entwickeln, desto größer wird die Wahrscheinlichkeit, dass unbeabsichtigte Seiteneffekte auftreten. Gleichzeitig steigt der Umfang der App, da wir konstant neue Funktionen hinzufügen und bestehende Funktion ändern. Um die Komplexität weiter zu beherrschen, ist schnelles Feedback zu Änderungen ein entscheidender Faktor, um Probleme schnell zu korrigieren. Wie wäre es, automatisiert ein Feedback nach jedem Commit zu bekommen und darauf nur kurze Zeit warten zu müssen? Genau hier setzten wir an.

Schnelles Feedback erhalten wir durch den Einsatz von **Feature-Toggles** und durch die Ausführung von **automatischen Tests** im CI-Prozess.

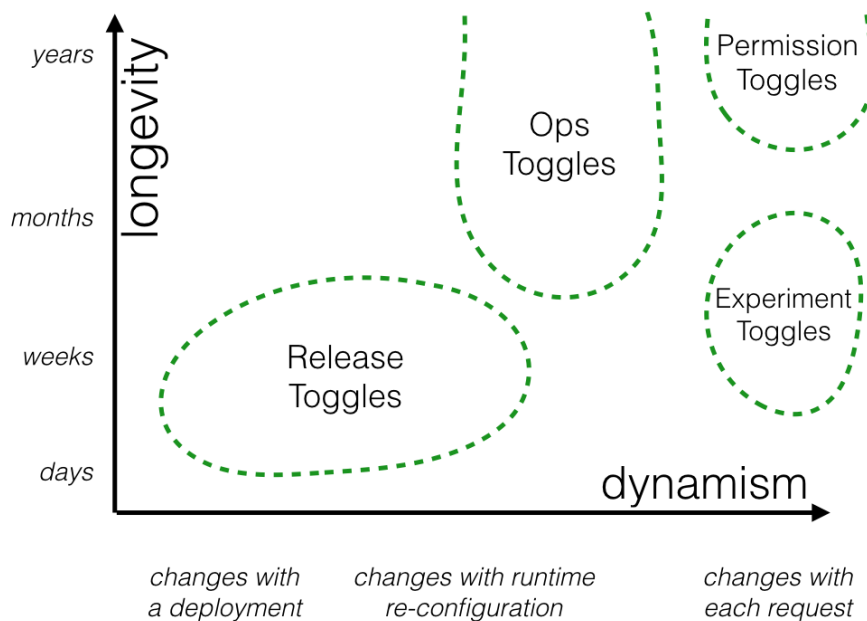
Feature Toogles

Feature Toogles ermöglichen es uns Codeänderungen aller Entwickler kontinuierlich in einen gemeinsamen Integration-Branch zusammenzuführen. Und das auch, wenn Features noch nicht fertig bzw. für den Kunden nicht sichtbar sein sollen. In der Vergangenheit haben wir solche Features für mehrere Tage, manchmal Wochen, in separaten Branches entwickelt und erst am Ende der Entwicklung in den Integration-Branch gemergt. Das Feedback kam entsprechend spät. Traten Probleme auf, war es durch die Vielzahl der Änderungen mitunter schwer festzustellen, welche konkrete Änderung zum Bruch geführt hat. Die Integration hatte das Potential unsere Zeitplanung empfindlich zu stören. Diese Bing-Bang-Szenarien sollen durch Toggles und kontinuierliche Integration abgefedert werden.

Toggles und Diskussionen

Der Einsatz der Feature-Toggles wurde im Team intensiv diskutiert, denn die Einführung erhöht erstmal die Komplexität – und lieb gewonnenen Pfade, wie die isolierte Arbeit im Feature-Branch, standen auf einmal auf dem Prüfstand. Es gab diverse Pros und Cons. Auch musste ein gemeinsames Verständnis beim Thema Toggle-Mechanik erarbeitet werden. In Bezug auf Dynamik und Langlebigkeit der Toggles gab es unterschiedliche Auffassungen, da viele schon mal irgendetwas mit Toggels gemacht hatten.

Wir haben uns am Ende auf die Nutzung von Feature-Toggles zur Entwicklungszeit – auch Release-Toggles genannt – geeinigt. Sie werden für den Zeitraum weniger Tage/Wochen genutzt. Ist das Feature fertig entwickelt, wird der Toggle aus dem Code komplett entfernt. Der Artikel auf martinfowler.com [1] sei dem interessierten Leser an der Stelle empfohlen.



<https://martinfowler.com/articles/feature-toggles.html>

2 bis 3 Feature-Toggle sind im Durchschnitt parallel im Einsatz. In unserem Jenkins haben wir durch einen manuellen Schritt in der Build-Pipeline die Möglichkeit geschaffen, einen einzelnen Toggle zu aktivieren und somit App-Artefakte für Features (apk, ipa) für das Testing zu bauen. Ist das Feature komplett entwickelt, wird der Toggle aus dem Code entfernt. Mit dem nächsten App-Release ist das Feature dann für den Kunden sichtbar.

Was ist ein Feature Toggle?

Ein Feature Toggle ist eine Programmiertechnik, die es erlaubt ein Feature oder eine Funktion vor Kunde ein- bzw. auszuschalten. Also die Sichtbarkeit zu ändern. Entwicklungsteams aktivieren Features beispielsweise um noch nicht fertige Funktionen integrieren und testen können. Ist ein Feature fertig, kann es ohne großen zusätzlichen Merge-Aufwand veröffentlicht werden, da die Arbeit in separaten Branches entfällt. Feature Toggles können auch dafür genutzt werden, die Sichtbarkeit von Funktionen zur Laufzeit der Anwendung zu ändern. Z.B. im Rahmen von A/B Tests oder wenn die Sichtbarkeit zu einer bestimmten Zeit geändert werden soll.

Toogles im Code

iOS

Unter iOS wird ein Feature in der App über eine Environment-Variable in der **Launch-Konfiguration** aktiviert (z.B.: USE_NATIVE_PRODUCT_LIST = 1). Im Code wird dann an relevanten Stellen über eine Abfrage entschieden, ob bestimmte Codestellen zur Ausführung kommen oder nicht.

```
if toggleRouter.isNativeProductListEnabled() {  
    // Feature Code  
}
```

Android

Es gibt ein Interface, in dem alle Toggles als Methoden definiert werden. Diese Methoden werden mit Java-Annotations annotiert und geben immer ein Boolean zurück - TRUE für Feature aktiv, FALSE für nicht aktiv.

```
@ReleaseToggle(BuildConfig.FEATURE_PRODUCT_LIST)  
fun isProductListEnabled(): Boolean
```

Eine eigens dafür entwickelte Library mit einem **Annotation-Prozessor** wird während der Build-Phase ausgeführt: Dieser schaut in einer Konfigurations-Datei (json) nach, ob das jeweilige Feature getoggelt werden soll. Wenn das Feature eingeschaltet werden soll, muss der String, der sich in der Annotation befindet, hier eingetragen werden.

```
[  
    "FEATURE_PRODUCT_LIST"  
]
```

Der Prozessor baut dann jeweils die Implementation für das Interface zusammen. In diesem Fall würde die implementierte Methode TRUE zurück liefern. Wäre der String FEATURE_PRODUCT_LIST nicht in der Datei, wäre es FALSE.

So kann man auf jedem lokalen Rechner die Features beliebig ein- und ausschalten. Auf dem Jenkins kann man das genauso machen, hier editieren wir nicht manuell die Datei sondern sagen ihm über das **Blue Ocean Plugin**, welches Feature getoggelt werden soll.

Und die jeweiligen Code-Stellen togglen wir, in dem wir die Interface-Implementation prüfen:

```
if (ReleaseToggleConfiguration_Impl().isProductListEnabled())  
{  
    // Mach was mit der Product list  
}
```

Ein gemeinsames Traffic Light für Build und Test

Eine weitere zentrale Komponente im CI-Prozess stellt die Testautomatisierung dar. Das Feedback, dass Build und Test erfolgreich nach einem Commit auf Integration durchgelaufen sind, wird durch eine Ampel visualisiert. Diese ist für jedem im Team sichtbar. Ist sie rot, ist das **gesamte Team angehalten** den Grund zu ermitteln und die Ampel wieder auf „grün zu bekommen“. Also Fehler zu korrigieren, Tests oder die Automatisierung anzupassen.



CI-Build-Status für Android und iOS

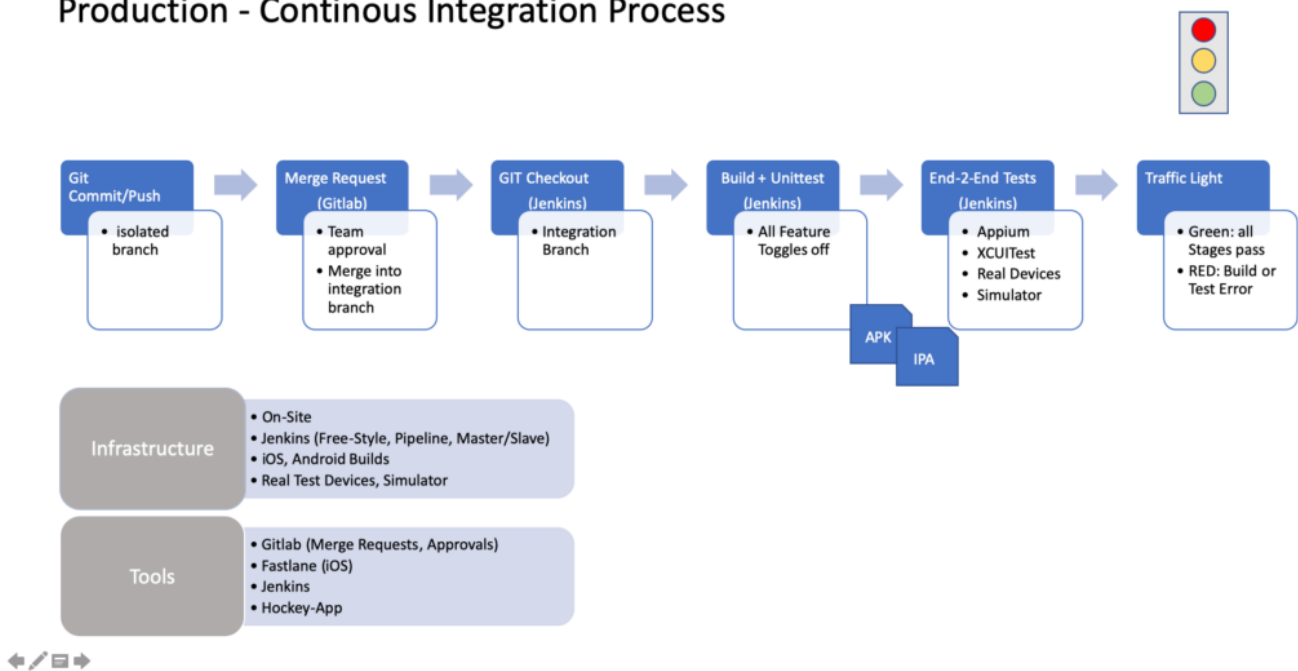
Die Tests sind eine Kombination aus Unit-Tests und End-2-End-Tests (Akzeptanztests). Die End-2-End-Tests laufen auf echten Geräten bzw. Simulatoren im Zusammenspiel mit dem Backend.

Continuous Integration Process

Unser CI Prozess sieht wie folgt aus:

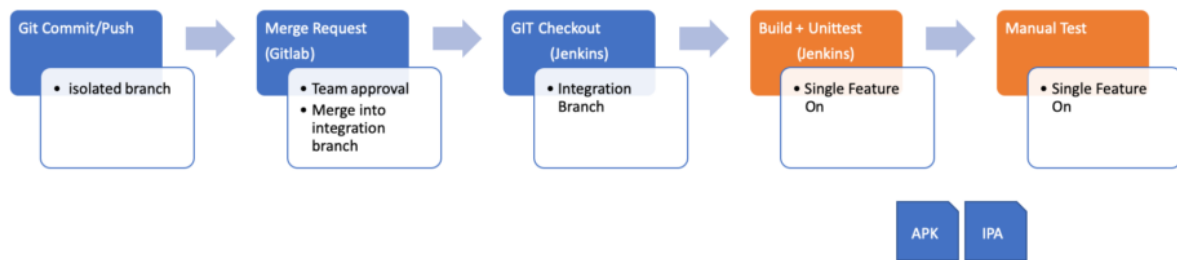
Nach dem Approval der Codeänderung in Gitlab und Integration in den Integrations-Branch baut der Jenkins das App-Artefakt, führt die Unittests aus und startet die End-2-End Tests. Das kombinierte Ergebnis aus Build/Unitests und End-2-End Test wird auf der Ampel dargestellt.

Production - Continuous Integration Process



Für den Test eines Features, dass sich noch in der Entwicklung befindet, wird ein Feature-Toggle manuell im Jenkins aktiviert, die App gebaut und die Unittests ausgeführt. Die End-2-End Tests werden zum jetzigen Zeitpunkt noch nicht ausgeführt. Zum einen müssten die Tests für das Feature bereits angepasst und erweitert sein. Das ist noch nicht der Fall. Ein weiterer Grund sind die noch fehlenden Ressourcen in Form von Hardware und Testgeräten. Ein nächster Schritt.

Feature Test - Continuous Integration Process



Learnings zu Tools, Integrationslevel und Verantwortung

Auf drei Learnings möchte ich an der Stelle speziell eingehen.

Der Leser soll dazu wissen, dass unsere Entwicklungsteams Cross-funktional aufgebaut sind. Ein Entwicklungs-Team besteht aus iOS-Entwicklern, Android-Entwicklern, Backend-Entwicklern und einem Quality Engineer. Die Backend-Entwickler sitzen an einem anderen Standort und sind wie wir Dienstleister für den B2B-Partner. Folgende Tests führen wir zur Zeit durch:

Manuell	• Ticketabnahmen • Release-Tests • Test-Objekt: App und Backend
End-2-End	• automatisiert (Teil des CI-Pipeline) • Appium Tests (Android) • XCUI Test (iOS) • Test-Objekt: App und Backend
Unit-Tests	• Android, iOS • Mocking • Test-Objekt: wenigen Klassen (App)

Testtypen

Beim Tooling ging es vor allem um die Wahl des **Testautomatisierungstools**. Da die QA in der Vergangenheit auf Appium gesetzt hat, wollten wir die Technik auch für unsere CI-Tests im gesamten Team nutzen (Dev+QA). Bisher wurden die Appium Tests ausschließlich von der QA geschrieben und waren nicht in einem gemeinsamen CI-Prozess zusammen mit dem Build integriert. Es stellte sich heraus, dass die Akzeptanz des Tools für iOS unter den Entwicklern sehr gering war. Stabilität, Funktionsumfang, Integrierbarkeit und Ausführungsgeschwindigkeit überzeugten nicht. Unsere Teams haben sich daher entschieden, für iOS die End-2-End Tests auf Basis von XCUITest neu zu schreiben. Für den Android Bereich setzen wir vorerst weiter auf Appium.

Ein weiteres Learning gab es beim **Integrationslevel**. Unsere End-2-End Tests weisen ein hohes Integrationslevel auf: die App wird im Zusammenspiel mit dem Backend getestet. Fehler im Backend oder eine schlechte Verbindungsqualität können dazu führen, dass Tests fehlschlagen, obwohl die App „korrekt“ funktioniert. Die Ampel zeigt rot, obwohl „mit der App alles in Ordnung ist“. Flaky Tests bzw. instabile Tests senken die Aussagekraft der Ampel deutlich und führen dazu, dass die Teams einer roten Ampel weniger Aufmerksamkeit schenken. Neben dem End-2-End Test planen wir daher einen zusätzlichen Testtyp einzuführen, der vom Integrationslevel zwischen Unittest und End-2-End Test liegt. Ziel ist es, die App ohne Backend zusätzlich zu verifizieren. Dafür sollen Backend-Responses „gemockt“ und die Tests auf der UI-Ebene durchgeführt werden. Die Tests sollen eine Ergänzung zu den End-2-End Tests werden.

Beim Thema Verantwortung gehen wir mit dem CI-Ansatz ebenfalls neue Wege. Das Ergebnis aus Build + Test in einem gemeinsamen Ampelstatus zu visualisieren und damit jeden im Entwicklungs-Team zu aktivieren, Probleme im **Test oder Build** zu analysieren, erfordert, dass sich Entwickler mehr mit dem Thema Testing und sich die QA mehr mit der Automatisierung auseinandersetzt. Dieser Veränderungsprozess benötigt Zeit und den Willen aller Beteiligten, sich zu verändern. In unserem Produktteam ist diese Veränderung explizit gewünscht und alle im Team sind angehalten, für den Prozess **Verantwortung** zu übernehmen und ihn aktiv weiterzuentwickeln.

Ausblick

Im Bereich Testing steht der Ausbau der Testautomatisierung für die End-2-End

Tests und die Einführung der zusätzlichen Test-Verifikationsstufe für die Android-App mit Espresso als UI-Testing Tool an.

Um die Qualität zu steigern, möchten wir automatisiert statische Codeanalysen durchführen und Metriken wie beispielsweise die technische Schuld ermitteln.

Verweise

[1] <https://martinfowler.com/articles/feature-toggles.html>

[2] Douglas App iOS:
<https://itunes.apple.com/de/app/douglas-parfüm-kosmetik/id394685685?mt=8>

[3] Douglas App Android:
<https://play.google.com/store/apps/details?id=com.douglas.main&hl=de>

Die innovative Idee der Voice App „Thalia Buchtipps“ kommt an!

Seit ein paar Monaten ist der Thalia Alex Skill erst online und erfreut sich bereits einer kontinuierlichen Nutzung. Vor allem die abwechslungsreiche und neckische Art kommt bei den Kunden gut an und konnte nun auch die Jury des „Deutsche Exzellenz-Preis“ überzeugen.

<#0100/> <hackathon@thalia/> in

Berlin

Ende Mai fand der 2. Hackathon in Berlin statt. Zum Thema waren diesmal alle neuen Technologien gesetzt.

Berlin macht Agil !



Dank der Unterstützung unseres Kollegen Matthias Hochschulz und seiner Organisation [Münster macht Agil](#), konnten wir gemeinsam im August auch in Berlin einen kostenlosen Scrum Workshop organisieren. Ursprünglich getrieben von der Notwendigkeit einer Reihe von neuen Kollegen die Grundwerte und -prinzipien von Scrum nahezubringen, entwickelten wir gemeinsam die Idee den Workshop auch für externe zu öffnen. So konnten wir am Ende fast 30 Interessierten aus und um Berlin dazu ermutigen, an einem realen Produkt, mehrere kurze Sprints zu durchlaufen.

Vorallem der Schritt zu einer öffentlichen Veranstaltung war für uns neu. Passte aber sehr gut zu unseren Bemühungen unsere Kompetenz in der Softwareentwicklung nach außen zu tragen. Für viele passt das Bild eines modernen agilen Softwarehauses nicht zu dem angestaubten Bild eines Buchhändlers. Daher haben wir uns sehr gefreut, das wir zeigen konnten, das dieses Bildniss in der Tat sehr veraltet ist und wir eine langjährige Erfahrung im eCommerce und der agilen Softwareentwicklung haben.

Nach einer kurzen Einführung (Stichworte: Sprint,

Review, Retrospektive, Daily-Stand-Up, Inspect & Adapt, Fokus und Transparenz) ging es direkt ans *Selbermachen*! Passend zum heißen Sommer sollten die Teams eine Wasserpistole aus einem Akkuschrauber bauen. Diese konnte über mehrere Sprints verbessert werden. Das ermöglichte, ganz im agilen Sinne, aus den ersten Erfahrungen zu lernen und auf neue Anforderungen zu reagieren.

Nach dem erfolgreichen Event konnten wir noch alle Beteiligten noch auf ein Bier und Pizza in unsere Büroräume einladen und uns so für künftigen Erfahrungsaustausch vernetzen.

[Link zum Xing Event](#)

Nachtrag:

Matthias hat [hier](#) zusätzlich einen sehr umfangreichen Erfahrungsbericht zu dem Workshop verfasst. Dieser reflektiert sehr ausführlich die Erfahrungen die Matthias während des Workshops als Moderator gesammelt hat.

Kids Programming - Thalia fördert bereits die Jüngsten!



Nachdem wir in den letzten Jahren einige Sommerpartys und eine riesige Weihnachtsfeier hatten, hat sich unsere Geschäftsleitung entschieden, für Ende 2017 einen „Family Day“ abzuhalten. Und so durften nicht nur die Lebensgefährten sondern auch alle Kinder mal schauen, was Papa / Mama so im Büro machen. Es gab an den einzelnen Standorten ein umfangreiches Rahmenprogramm. Dabei

wurde stets versucht, auch einen gewissen Bezug zum Unternehmen zu wahren. So bot sich die Bücherecke mit Vorlesungen für Kinder natürlich an, was vor

allem einige der jüngeren Kinder sehr genossen haben. Aber was macht man an einem Software Standort und wie erklärt man die Entwicklung von Programmen seinen Kindern?

Inspiriert von der „[Langen Nacht der Wissenschaft](#)“ in Berlin, wurde die Idee des Kids Programming geboren. An einzelnen Plätzen konnten so die Kinder bereits ab 4 Jahren in die Welt des Programmierens eingeführt werden und eine Idee entwickeln, was ein Software Entwickler eigentlich macht.

Die grundlegende Idee

Wir haben dafür mit Hilfe von [code.org](#) einige Programme für die Altersgruppe 4-6 und 6-12 rausgesucht und unterschiedliche Themen gewählt. Dadurch konnten wir später nicht nur Jungen sondern auch einige Mädchen für die Kurse begeistern. Für die fortgeschrittene Programmierung stellten wir darüber hinaus noch Scratch vom MIT zur Verfügung und konnten das Ganze durch einige, speziell für Kinder gestaltete Lehrbücher, unterstützen.

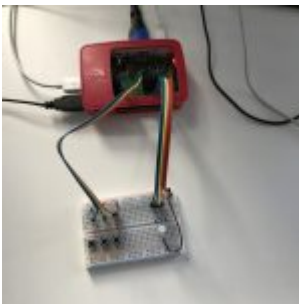
Darüber hinaus wollten wir bei den Kindern aber auch das Interesse für Softwareentwicklung wecken und sie neugierig auf mehr machen. Deshalb haben wir unsere Raspberry Farm ebenfalls zur Scratch Spielwiese umfunktioniert und mit einigen Hardware Projekten ergänzt. Das wurde dann zum echten Blickfang und sorgte für die Aufmerksamkeit von jung und alt. So bekamen wir am Ende mindestens genauso viel interessierte Eltern wie Kinder in den Kurs.

Programmieren mit code.org und Scratch

Die Idee dahinter ist nicht eine echte Programmiersprache zu erlernen, sondern ein Verständnis von strukturierten Abläufen und Funktionsbausteinen zu schaffen. So können die Kinder mit einzelnen Code-Blöcken, im Lego Baukasten Prinzip, ihre Programme zusammenklicken. Dabei geht es vom Erkennen von Wiederholungen / Schleifen bis zum Erstellen eigener Funktionen. Während Scratch eine reine Entwicklungsumgebung bietet, unterstützt code.org das mit stückierten, auf einander aufbauenden Kursen.

Einsteiger beschrieben. Bis auf den Raspberry wird auch alles benötigte Zubehör mitgeliefert.





Wer Interesse bekommen hat, das Ganze mal selber auszuprobieren, wird hier fündig:



Christian Immler

Der kleine Hacker: Programmieren für Einsteiger

Mit Scratch schnell und effektiv programmieren lernen

Der kleine Hacker

★★★★★

eBook
ab 14,99 €

Taschenbuch
29,99 €

Programmieren ist langweilig und trocken? Nicht mit dem kleinen Hacker! Die grafische Programmiersprache Scratch macht's möglich. Programmieren lernen mit Spaß. Egal, ob du nur die Katze tanzen lassen oder ein richtiges Spiel programmieren willst - Scratch eignet sich sowohl zum Einstieg in die Programmierung als auch für anspruchsvollere Projekte.

Links zu den

Seiten: code.org, scratch.mit.edu

Ein Team sieht Rot

Da das letzte Team Event, ihr erinnert euch: im [JumpHouse](https://www.jumphouse.de), schon wieder ein paar Tage zurück lag, war es Zeit für ein neues Event. Da passte es gut, dass wir herausfanden, dass manche noch nie Raclette gegessen hatten und andere noch nie den Film R.E.D.2 gesehen hatten! Was liegt näher als beides zu verbinden?! So entstand die Idee zum Kinoabend mit gemeinsamen Raclett essen und wir haben den Arbeitstag mit einem gemütlichen Kineoevent ausklingen lassen.



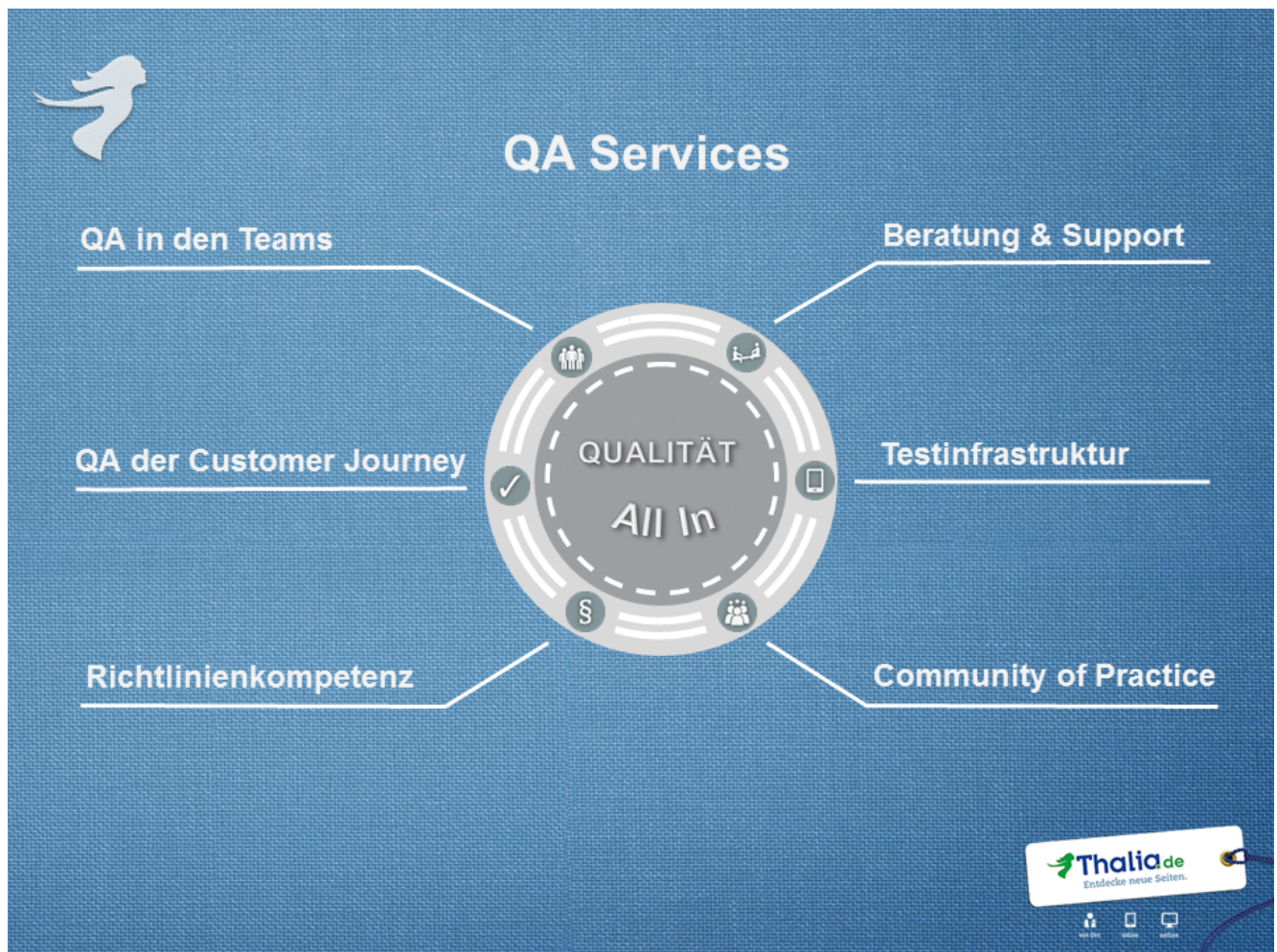
[R.E.D. 2 - Noch Älter. Härter. Besser](#)

[Trailer](#)

QA Services - Mittendrin statt nur am Ende

Der Bereich Thalia eCommerce Quality Assurance Services unterstützt die anderen Bereiche bei dem Ziel, gegenüber den Kunden und Kundinnen ein hochwertiges Produkt zur Verfügung zu stellen. Wir sind dabei entweder direkt einem Team zugeordnet oder in der zentralen QA tätig. Durch die permanente Weiterentwicklung von Produkten und Prozessen werden wir stets gefordert,

neue passende Lösungen zu entwickeln. Derzeit stehen bei uns sechs Themen im Fokus:



QA in den Teams

Wir sind als QA-Lead direkter Bestandteil eines cross-functional Teams und nehmen vielfältige Aufgaben wahr:

- Abstimmung der teaminternen Qualitätskriterien
- Erster Ansprechpartner zum Thema Qualitätssicherung und Test
- Themen zur Testautomation im Team bündeln und vorantreiben
- und natürlich das normale Testgeschäft

Beratung und Support

Wir beraten die einzelnen Teams und Fachbereiche bei Themen wie Testfallermittlung, Testkonzeption, Tooleinsatz etc.

QA der Customer Journey

Wir prüfen zyklisch, dass die Customer Journey für unsere Kunden keine Stolpersteine enthält. Die Prüfung durch automatisierte Tests bauen wir stetig aus.

Testinfrastruktur

Wo und wann kann ich testen? Woher bekomme ich Testdaten? Womit kann ich Lasttests ausführen? Das sind einige Fragestellungen, die wir beantworten. Und falls wir die Antwort nicht kennen, sorgen wir für Lösungen.

Richtlinienkompetenz

Wir definieren Spielregeln, damit die Flexibilität und hohe Innovationsgeschwindigkeit der agilen Produktentwicklung nicht zu Lasten von Stabilität und Qualität geht.

Community of Practice

Wir tauschen uns zu spannenden Themen im Bereich der QA aus, um voneinander zu lernen und besser zu werden.

Berlin live for the code



Time	Session Title	Speaker
07:30	Registration	
08:30	Introduction to the Oracle Code Event	
09:00	Keynote: The Future of Cloud Native Applications	
09:30	Breakfast	
10:00	Session 1: Introduction to Oracle Cloud Native	
10:30	Session 2: Introduction to Oracle Cloud Native	
11:00	Session 3: Introduction to Oracle Cloud Native	
11:30	Session 4: Introduction to Oracle Cloud Native	
12:00	Session 5: Introduction to Oracle Cloud Native	
12:30	Lunch	
13:00	Session 6: Introduction to Oracle Cloud Native	
13:30	Session 7: Introduction to Oracle Cloud Native	
14:00	Session 8: Introduction to Oracle Cloud Native	
14:30	Session 9: Introduction to Oracle Cloud Native	
15:00	Session 10: Introduction to Oracle Cloud Native	
15:30	Session 11: Introduction to Oracle Cloud Native	
16:00	Session 12: Introduction to Oracle Cloud Native	
16:30	Session 13: Introduction to Oracle Cloud Native	
17:00	Session 14: Introduction to Oracle Cloud Native	
17:30	Session 15: Introduction to Oracle Cloud Native	
18:00	End of Event	

Agenda - Berlin

Ein paar Kollegen aus allen Berliner Entwicklerteams, haben die Gelegenheit genutzt und die [Oracle Code](#) besucht. Natürlich zog sich Oracles neues Produkt, die Oracle Plattform Cloud, als roter Faden durch alle Präsentationen. Die Plattform möchte Oracle demnächst starten und bietet alles was das Entwicklerherz begehrt (von Source Verwaltung, Scrum Board, Ticket System, Container Orchestrierung bis zum Deploy). Allerdings lassen bereits die großzügig spendierten 300\$ Startkredit (für alle Teilnehmer zum ausprobieren) erahnen, das die Plattform nicht umsonst ist.

Erfreulicherweise gab es aber über die Plattform hinaus auch zahlreiche interessante Vorstellungen verschiedener Tools und Mechanismen. So zeigte Adam Bien sehr beeindruckend das eine Source Codeänderung inkl. Maven Build und Deploy per Docker nur eine Frage von Sekunden ist. Darüber hinaus gab es noch Einführungen in Kafka und Redis und jeweils eine Erläuterung wie „Failover“ Lösungen bzw. ein redundanter Service aussehen könnte.

So bleibt am Ende ein spannender Tag der uns viele neue Anregungen mit in die Teams nehmen lässt.

Und für alle die nicht dabei sein konnten: [Download der Vorträge in New York](#)

<https://youtu.be/O3gMnVrOyuU>