

Infrastructure Engineer (m/w/d)



Bücher eröffnen Welten – wir tun es ihnen gleich. Als Deutschlands größter Sortimentsbuchhändler verknüpfen wir echte Kundennähe mit richtungsweisender Digitalinnovation. Wir denken das Buch als Erlebnis – ob vor Ort, online, als App oder über den e-Reader. Und versammeln rund 4000 Expertinnen und Experten mit einer Mission: Ein Kulturgut auf allen Kanälen erfahrbar zu machen.

Für unsere Zentrale in **Münster** suchen wir **ab sofort** einen

System Engineer (m/w/d) –
Schwerpunkt Microsoft/
Infrastruktur

Welche Aufgaben erwarten Dich:

- Im Team **Infrastructure Engineering** entwickelst und implementierst Du die technische Plattform für den Bereich eCommerce und unsere Buchhandlungen
- Zusammen mit Deinen Kollegen*innen entscheidest Du über die **Weiterentwicklung** der **IT-Architektur** und dessen Betrieb
- Du nutzt dein **Script-Foo**, um mit **PowerShell** alles zu automatisieren

- Gemeinsam im Team gehst Du auf die Jagd nach technischen Problemen und löst diese
- Du bringst deine **Ideen** ein, um unsere Plattform noch robuster, einfacher und schneller zu machen
- Du bleibst am Puls der Zeit, **teilst Dein Wissen** mit **den Teams** und berätst diese

Du willst mehr über die Teams, Technologien und Methoden wissen? Dann schau gerne hier vorbei: <https://tech.thalia.de/tag/peng/>

Was bringst Du mit:

- Bei **Microsoft Backend Systemen** und **IT-Netzwerken** (Routing, VLAN, WAN, etc.) macht Dir so schnell keiner etwas vor
- **Active Directory, Windows Server** und **VMware ESX** beherrscht Du richtig gut und **Microsoft 365** hast Du schon mal genutzt oder willst Du unbedingt kennenlernen
- „Cloud“ ist für Dich kein Buzzword: Du hast richtig Lust das sinnvollste aus dem eigenen Rechenzentren und der **Cloud** zu verbinden
- Du kannst andere für Deine Ideen begeistern und zeichnest Dich durch **proaktives Arbeiten** sowie eine gute **Teamfähigkeit** aus
- In Deiner Welt ist alles **automatisierbar** und Du findest Fehler, um daraus zu lernen
- Du hast ein sehr hohes **Verantwortungs- und Sicherheitsbewusstsein** für unsere Plattform

Diese Benefits bieten wir Dir:

Für uns selbstverständlich - Unbefristete Festanstellung mit flexiblen Arbeitszeiten, Remote Arbeit, 30 Tagen Urlaub im Jahr sowie Sonderurlaubstage.

Mit uns weiterkommen – eLearnings und Toolboxen, Coaching, Support und ein Fortbildungsbudget für die individuelle Entwicklung.

Täglich versorgt: Kaffee-, Tee- und Wasserflatrate, vergünstigtes Mittagsangebot in der Kantine sowie reichlich Platz für die gemeinsame Mittagspause.

Wir lernen von- und miteinander: Teamorientierte Unternehmenskultur, Kommunikation auf Augenhöhe sowie hoher Wissensaustausch.

Haben wir dein Interesse geweckt?

Dann bewirb dich unter Angabe deines Gehaltsrahmens und deines frühestmöglichen Eintrittstermins über unser [Online-Portal](#) oder ganz einfach über bewerbungen@thalia.de!

Den Bewerbungsprozess haben wir für dich entspannt und einfach gestaltet. Hier ein paar Einblicke am Beispiel Linux Administrator*in: <https://tech.thalia.de/wie-werde-ich-linux-administratorin-bei-thalia/>

Deine Ansprechpartner*innen:



Phillip Vojinovic

HR Ansprechpartner

p.vojinovic@thalia.de



Sören Schmitz

Teamleiter Infrastructure Engineering

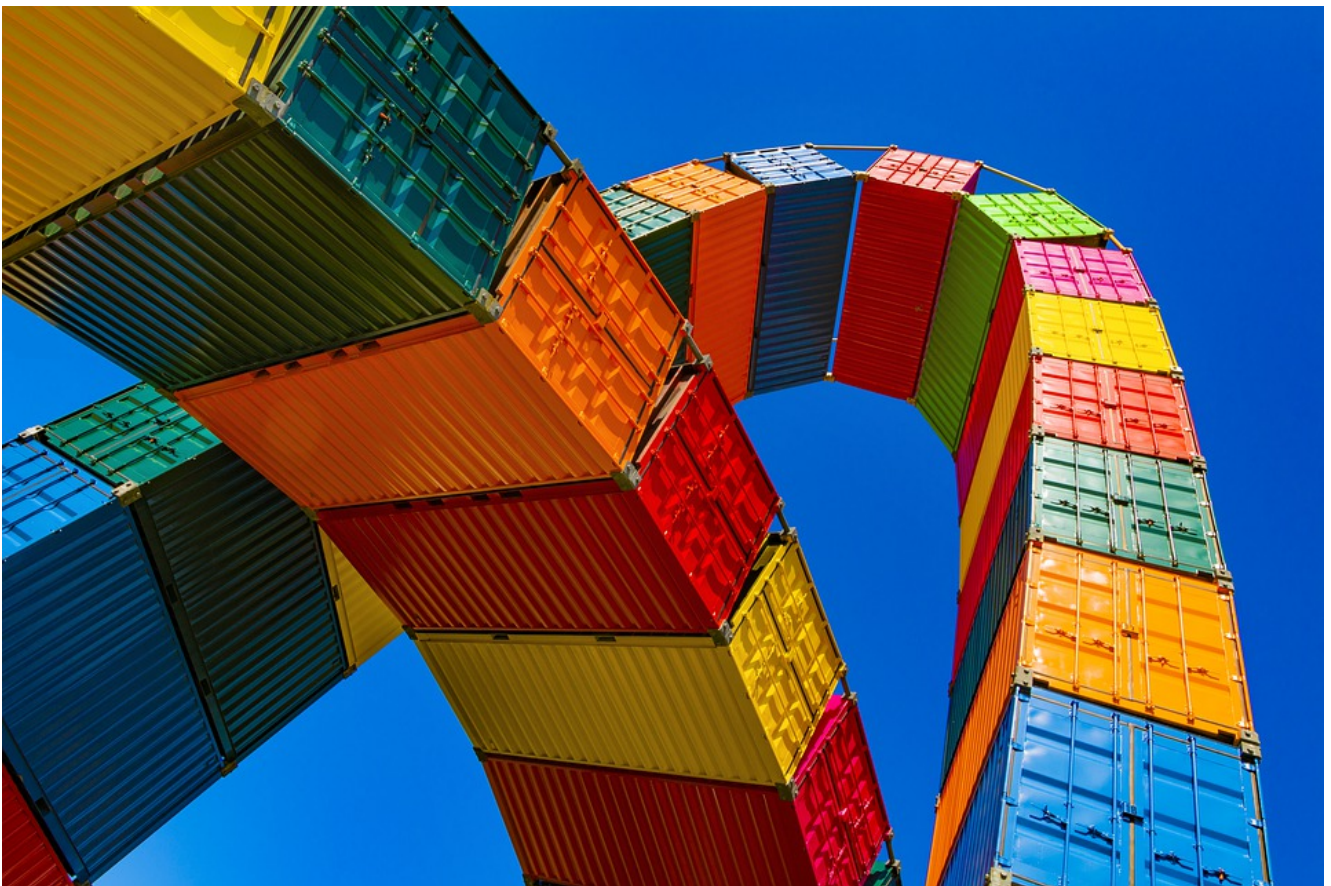
s.schmitz@thalia.de



Christoph Drosten

Head of IT-Operations

Kubernetes mal anders mit Tanzu



Container waren lange Zeit dem Warentransport vorbehalten – auch bei Thalia. Aber heutzutage wächst auch der Bedarf an Containern im Bereich der IT. Das vollständige Potential lässt sich erst erzielen, wenn auch ein entsprechender Orchestrator verwendet wird, um die Verwaltung zu übernehmen. Wo liegen jetzt die Vorteile von Containern und was bringt mir ein Orchestrator außer einer zusätzlichen und womöglich zugleich unbekannten Technologie?

Ein immer schnellerer Entwicklungszyklus und damit auch häufigere Deployments sind ein Treiber. Hierbei bieten Container die Möglichkeit, einen

vordefinierten und unveränderlichen Stand, auch als „Immutable Image“ bezeichnet, in unterschiedlichen Architekturen auszurollen. Somit ist es auch simpel, diese Images sowohl auf den Geräten der Entwickler, als auch in den unterschiedlichen Entwicklung-Stages zu verwenden.

Ein weiterer Treiber sind die Skalierungsmöglichkeiten, die durch einen Container Orchestrator wie Kubernetes entstehen. Anwendungen können hierbei anhand von Auslastungsmerkmalen wie der CPU- oder der RAM-Verbrauch automatisch skaliert werden, was ein manuelles Eingreifen überflüssig macht. Bei hohen Auslastungen können hierdurch weitere Container bereitgestellt, andersherum bei wenig Auslastung auch wieder reduziert werden, was Einsparungen ermöglicht.

Ideenfindung/Werdegang

Kubernetes ist in der heutigen Zeit als Orchestrator für Container eine gesetzte Technologie. Aber gleichzeitig birgt diese auch das Risiko, einen gewaltigen Mehraufwand im Betrieb zu erzeugen. Schnelle Entwicklungszyklen, die häufige Updates von Kubernetes selber erzwingen, um am Ball zu bleiben. Viele Komponenten innerhalb von Kubernetes, die ebenfalls verwaltet und aktualisiert werden wollen. Und das bei immer mehr Abstraktionsschichten, von der eigentlichen Hardware hin zu der eigentlichen Basis für die Anwendung. Insgesamt entsteht also ein hilfreiches, aber auch komplexes Konstrukt, dessen Management sichergestellt sein muss, um sich nicht kontraproduktiv auszuwirken.

Wie bekommt man also den schmalen Grad hin, bei einer vordefinierten Anzahl von Administratoren den Aufwand möglichst gering zu halten und trotzdem die Vorteile gewinnen zu können? Hierzu gibt es unterschiedliche Varianten, die wir uns angeschaut haben, bevor wir schlussendlich bei der jetzigen Lösung angekommen sind.

Die erste Herangehensweise war, sich mit Kubernetes selbst auseinanderzusetzen. Bezeichnet als „Vanilla Kubernetes“ machten wir uns mit entsprechender Automatisierung dran, Kubernetes auszurollen, miteinander zu verknüpfen und auch zu nutzen. Die ersten Anwendungen waren schlussendlich bereits in den Entwicklungsstages ausgerollt und aktiv, bevor wir die Reißleine

ziehen mussten. Mit der in der Zwischenzeit gesammelten Erfahrung war ein Betrieb in „Produktion“ schlichtweg nicht realistisch. Zu hoch war der Aufwand und das hiermit verbundene Betriebsrisiko.

Der nächste logische Schritt war, dass der Betrieb komplett ausgelagert wird. Somit könnten wir uns auf das Deployment der Anwendungen konzentrieren und müssten uns um die Verwaltung keine Gedanken machen. Somit würden auch keine Ressourcen unsererseits in Beschlag genommen. Klingt doch soweit super, könnte man denken? Aber auch das hat am Ende leider nicht die gewünschten Ergebnisse gebracht. Bei solchen „managed“ Lösungen gilt es einige Punkte zu beachten. Da wären z.B. das Kosten-Nutzen-Verhältnis aber auch der Abgang von Erfahrungen im Betrieb und damit das Verständnis für tiefere Ebenen der Technologie. Zusätzliche benötigte Funktionen müssen ebenfalls auf eigene Faust gestemmt werden oder gehen weiter zu Lasten der Kosten.

Schauen wir uns also einmal die Probleme an. Häufige Updates und Abhängigkeiten sind ein großer Treiber der Technologie. Auf der anderen Seite benötigen wir mehr als eine „managed“ Lösung uns im ersten Schritt bieten kann. Lässt sich also die grundlegende Verwaltung, wie z.B. Updates und Management von Abhängigkeiten, simpler gestalten? Aber gleichzeitig der eigentliche Betrieb und somit auch das tiefergreifende Wissen erhalten? Und das bei einem sinnvollen Kosten-Nutzen-Verhältnis?

Hier kommt „VSphere with Tanzu“ als Produkt von VMWare ins Spiel, welches im Nachfolgenden nur noch als „Tanzu“ bezeichnet wird.

Planung

Für die Recherche und die Prüfung des Produkts gab es verschiedene Schritte. Die Idee war aber immer eine pragmatische Herangehensweise. In einem ersten kurzen Test haben wir uns die Grundlagen angesehen, einen Cluster aufgesetzt, erste Objekte angelegt und dann Anwendungen hineingesetzt und die Funktionen geprüft. Soweit so gut, die erste Hürde war gemacht und bisher noch keine Blocker entdeckt.

Im nächsten Schritt haben wir uns dann noch einmal genauer mit den Details auseinandergesetzt, den technologischen Aufbau rekapituliert, Testfälle heruntergeschrieben und zusätzliche Kollegen eingeweiht und einbezogen. Mit

dieser Vorplanung ging es dann in den PoC (Proof of Concept). Das Setup konnten wir zum Glück noch wiederverwenden, hier gab es keine nötigen Anpassungen. Um einige Funktionen erweitert und die Testfälle abgeklappert ging die Stimmung weiter nach oben. Jetzt galt es noch die Entwicklungsteams abzuholen, eine ungenutzte Hülle hilft ja schließlich keinem. Hier kamen also die DevOps-Kollegen aus unseren Teams ins Spiel, um die finalen Bereiche abzuklären. Und wäre das nicht mit Erfolg gekrönt, würden diese Zeilen wohl nie gelesen werden.

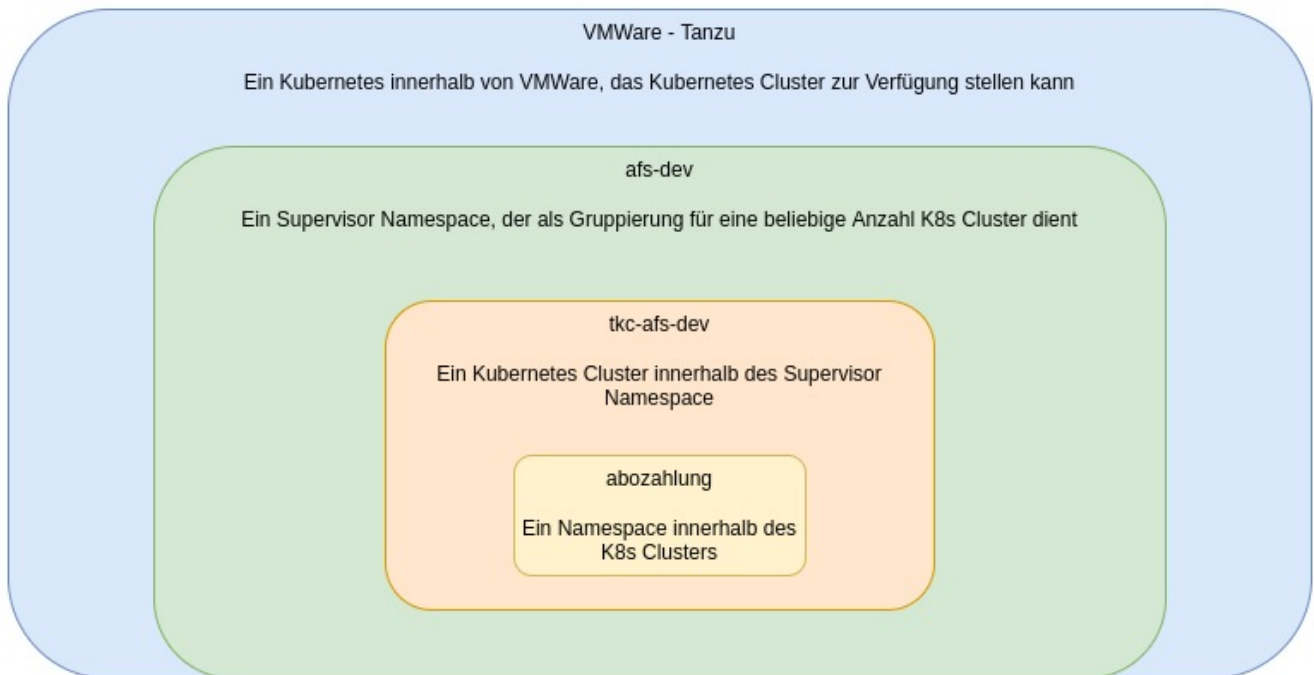
Jetzt ging es also an den größeren Brocken der Arbeit. Das Projekt muss geplant werden, um die Technologie auch sinnvoll für die Entwickler bereitzustellen. Viele Punkte wurden im PoC schon abgedeckt, aber der Teufel steckt ja bekanntlich im Detail. Auch Gedanken zu internen Abläufen, Prozesse zur Nutzung und das spätere Onboarding der Kollegen sollten noch die ein oder andere Stunde verbrauchen.

Was ist Tanzu eigentlich?

Aber noch einmal einen Schritt zurück. Erst ist von Kubernetes die Rede, dann Tanzu. Wo liegt denn jetzt eigentlich der Unterschied?

Wie zu Beginn schon erwähnt, bringt Kubernetes im Container Kontext viele Vorteile. Im Management aber auch einige Hürden mit sich. Tanzu bildet eine Abstraktionsschicht um einige dieser Hürden und versucht so, den Management Overhead zu minimieren. Es bildet unter anderem ein Rahmenwerk aus verschiedenen Technologien, die nach Prüfung von VMWare in einem Zusammenspiel gut miteinander funktionieren und auch unterstützt werden. Somit werden zwar ein paar Standards vorausgesetzt, was die ein oder andere Einschränkung mit sich bringt, aber im großen Ganzen noch alle Anforderungen erfüllt. Gleichzeitig gibt es jemanden, der einem bei Problemen weiterhelfen kann. Genau diese Standards ermöglichen es ebenfalls, dass Updates über wenige Klicks in Tanzu abgebildet sind. Im VCenter habe ich mit Tanzu einen Überblick über alle mit Tanzu gebauten Kubernetes Cluster und auch deren Versionsstand. Ebenso finde ich hier eine simple Möglichkeit alle Cluster auch auf einen neuen Stand zu bringen. Was in dem Kontext nur fehlt, sind die Updates für Plugins und Erweiterungen. Aber auch hier bietet Tanzu einen Download kompatibler Versionen inkl. detaillierter Anleitung zur Umsetzung, was das Vorgehen deutlich vereinfacht.

Was es nicht einfacher macht? Abstraktionsschicht um Abstraktionsschicht wird die Komplexität trotzdem nicht geringer. Um es sich zu verdeutlichen, hier einmal der Aufbau der mittlerweile entstandenen Schichten für die Kubernetes Cluster:



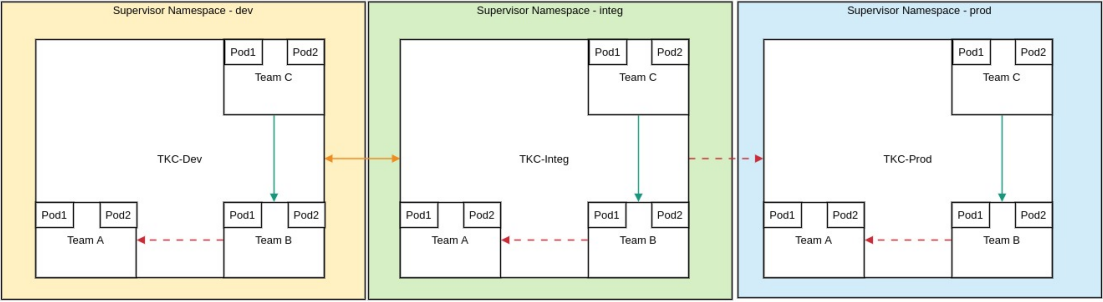
Architektur und Umsetzung

Die grundsätzlichen Überlegungen zu Abläufen etc. sind getroffen. Die Arbeitspakete geschnürt. Aber auch die Architektur soll wohl überlegt sein, bevor Sie Fallstricke enthält. Wie viele Cluster werden wir brauchen? Wie bauen wir das Netzwerk zwischen den Clustern auf? Brauchen wir ein Staging Konzept für die Cluster selber und nicht nur die Anwendungen? Das und vieles mehr galt es jetzt im konkreten Kontext anzugehen und zu klären.

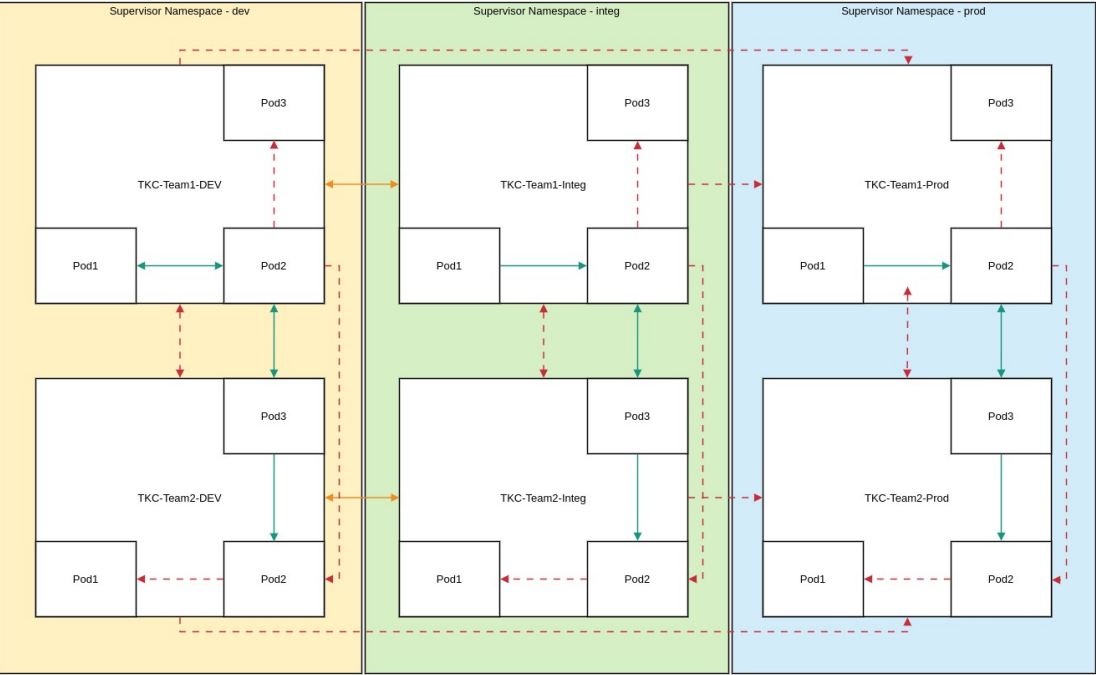
Einiges lässt sich einfach beantworten, anderes dann wiederum nicht. Schließlich sollen ja die unterschiedlichen Stages der Anwendungen wie Integration und Produktion bestmöglich voneinander getrennt sein. Auf der anderen Seite muss aber der Management Overhead überschaubar bleiben. Und dann gibt es ja nicht nur die unterschiedlichen Stages, sondern wir haben auch noch verschiedene Teams, die dann Ressourcen nutzen. Je mehr Cluster wir also aufbauen, desto komplizierter wird die Verwaltung des Konstrukts. Aber je weniger Cluster, desto schlimmer wird die Trennung – sowohl auf Netzwerk, der Ressourcen als auch der Berechtigungsebene.

Wie ein paar Überlegungen ohne nähere Erläuterungen aussehen können:

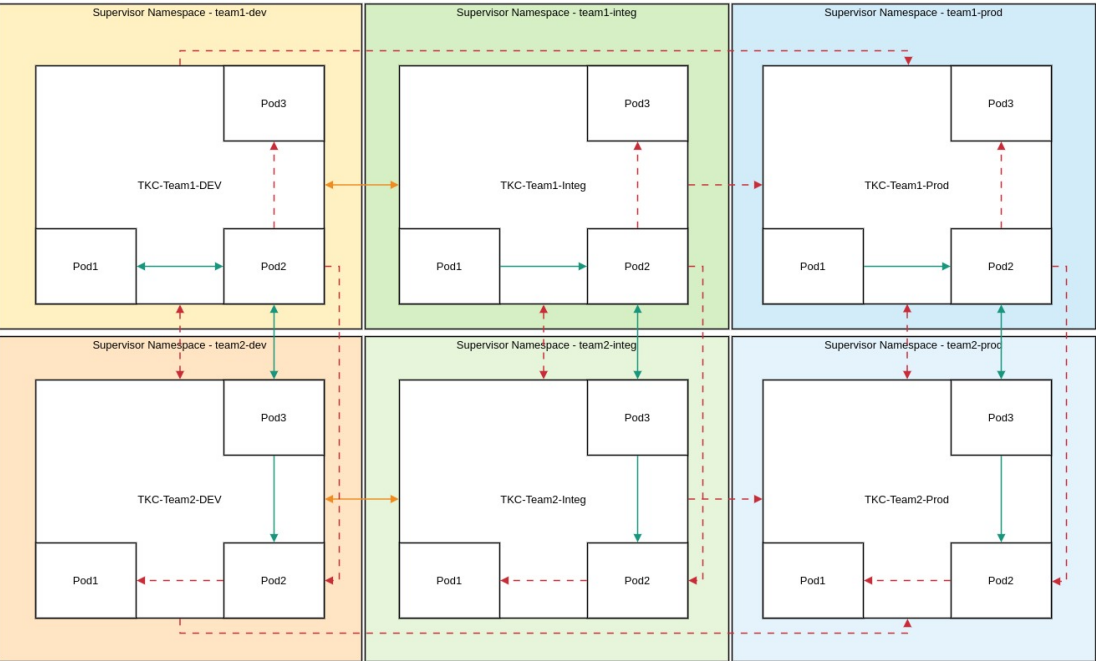
Option A



Option B



Option C



Architekturoptionen mit Kubernetes

Eines war klar, wir brauchen auf jeden Fall eine Automatisierungslösung. Eine Lösung, die uns so viel wie möglich von der Arbeit abnimmt und gleichzeitig dafür sorgt, dass wir die Infrastruktur reproduzieren können. Wie sonst auch soll dieser Ansatz basierend auf dem „Infrastructure as Code“ Prinzip abgebildet werden, womit die Wahl des Konstrukts am Ende unabhängig davon ist.

Ein wenig Recherche später qualmt der Kopf schon wieder. Sollten wir eine Lösung wie Flux oder ArgoCD nehmen? Genügt uns eine Jenkins oder Gitlab Pipeline? Ist Terraform die finale Lösung? Oder wird es womöglich eine Kombination aus mehreren Tools?

Wir haben ja schließlich mehrere Schritte, welche wir in der Provisionierung eines Clusters abdecken müssen. Zum einen der Bau eines Clusters über VSphere, die Einrichtung des Netzwerkes innerhalb des Clusters und auch die Anbindung der Services wie Monitoring, Logging und Alarming wollen in der Initialisierung bedacht werden. Und dann gibt es auch noch Sonderlocken von VMWare, wie z.B. Extensions, die einem teilweise die Arbeit abnehmen, durch die Standardisierung aber auch nicht für alle Fälle bei uns geeignet sind.

„Schuster bleib bei deinen Leisten“ heißt ein alter Spruch. Somit war die naheliegendste Lösung, vorerst eine Pipeline in unserem bestehenden Tool zu bauen. Die Daten sollen aus einem Git-Repository ausgelesen und alle nötigen Schritte hierin umgesetzt werden. Die Erfahrung wird zeigen, ob wir hier noch etwas nachzubessern haben und auf eine der anderen Lösungen schwenken, aber vorerst können wir alles abdecken. Und das ohne eine weitere neue Technologie, in die wir uns erst einarbeiten müssen.

Das Endergebnis des gesamten Setups im Tanzu Kontext kann sich aus unserer Sicht auf jeden Fall sehen lassen!

Onboarding

Der Tag der Tage ist gekommen. Nach all der Planung, Konzeptionierung und Umsetzung steht die Vorstellung vor der gesamten Entwickler-Community an. Doch was soll schon groß passieren? Die Kommunikation mit den Teams lief bereits vorher regelmäßig durch kurze Präsentationen des Standes. Unsere DevOps-Kollegen in den Teams haben uns im Piloten unterstützt. Und insgesamt

war das Feedback bisher durchweg positiv.

Was hiernach trotzdem noch fehlt? Die Migration. Wir freuen uns schon auf die Zusammenarbeit mit den Teams und sind gespannt, wie die Lösung im weiteren Verlauf auch unseren Betriebs-Alltag verbessern wird.

Ein Artikel von



Matthias Efker

Linux Systems Engineer



Suginthan Rushiyendra

Linux Systems Engineer

Du hast Lust ein Teil des Teams zu werden?

Systems Engineer /

Systemadministrator - Retail IT (m/w/d)



Bücher eröffnen Welten – wir tun es ihnen gleich. Als Deutschlands größter Sortimentsbuchhändler verknüpfen wir echte Kundennähe mit richtungsweisender Digitalinnovation. Wir denken das Buch als Erlebnis – ob vor Ort, online, als App oder über den e-Reader. Und versammeln rund 4000 Expertinnen und Experten mit einer Mission: Ein Kulturgut auf allen Kanälen erfahrbar zu machen.

Für unsere Zentrale in **Hagen** suchen wir **ab sofort** einen

Systems Engineer /
Systemadministrator - Retail IT
(m/w/d)

Welche Aufgaben erwarten Dich:

- **Planung, Rollout und Betrieb** von innovativen Filialtechnologien
- **2nd Level Support** für über 350 Thalia Buchhandlungen
- **Koordination von Dienstleistern** z. B. bei der Eröffnung neuer Buchhandlungen

- Optimierung von **technischen Prozessen** im Filialbereich
- Einbringung von Ideen zur weiteren **Modernisierung der Filialtechnologie**

Was bringst Du mit:

- Eine abgeschlossene Ausbildung und/oder Studium im Bereich IT
- Fundierte Kenntnisse im Bereich **Microsoft 365, Window Server/Client sowie Active Directory**
- Erfahrungen im **Betrieb von Netzwerken** (LAN/WLAN/WAN/MPLS) sowie Telefonanlagen
- Idealerweise KnowHow im Bereich Tablets (iOS) sowie Mobile Device Management
- **Pragmatische und lösungsorientierte** Arbeitsweise, **ausgeprägte Kommunikationsfähigkeit** sowie eine hohe Serviceorientierung
- Interesse an neuen Retail-Technologien

Diese Benefits bieten wir Dir:

Für uns selbstverständlich - Unbefristete Festanstellung mit flexiblen Arbeitszeiten, Möglichkeit zur Remote Arbeit, 30 Tagen Urlaub im Jahr sowie Sonderurlaubstage.

Nicht nur für Leseratten - Thalia Shopping Benefits sowie weitere Mitarbeitervorteile.

Täglich versorgt: Kaffee-, Tee- und Wasserflatrate, vergünstigtes Mittagsangebot in der Kantine sowie reichlich Platz für die gemeinsame Mittagspause.

Wir lernen von- und miteinander: Teamorientierte Unternehmenskultur, Kommunikation auf Augenhöhe sowie hoher Wissensaustausch.

Haben wir dein Interesse geweckt?

Dann bewirb dich unter Angabe deines Gehaltsrahmens und deines frühestmöglichen Eintrittstermins über unser [Online-Portal](#) oder ganz einfach über bewerbungen@thalia.de!

Den Bewerbungsprozess haben wir für dich entspannt und einfach gestaltet. Hier ein paar Einblicke am Beispiel Linux Administrator*in: <https://tech.thalia.de/wie-werde-ich-linux-administratorin-bei-thalia/>

Deine Ansprechpartner*innen:



Phillip Vojinovic

HR Ansprechpartner

p.vojnovic@thalia.de



Safwan Shunnar

Teamleiter Retail-IT

s.shunnar@thalia.de



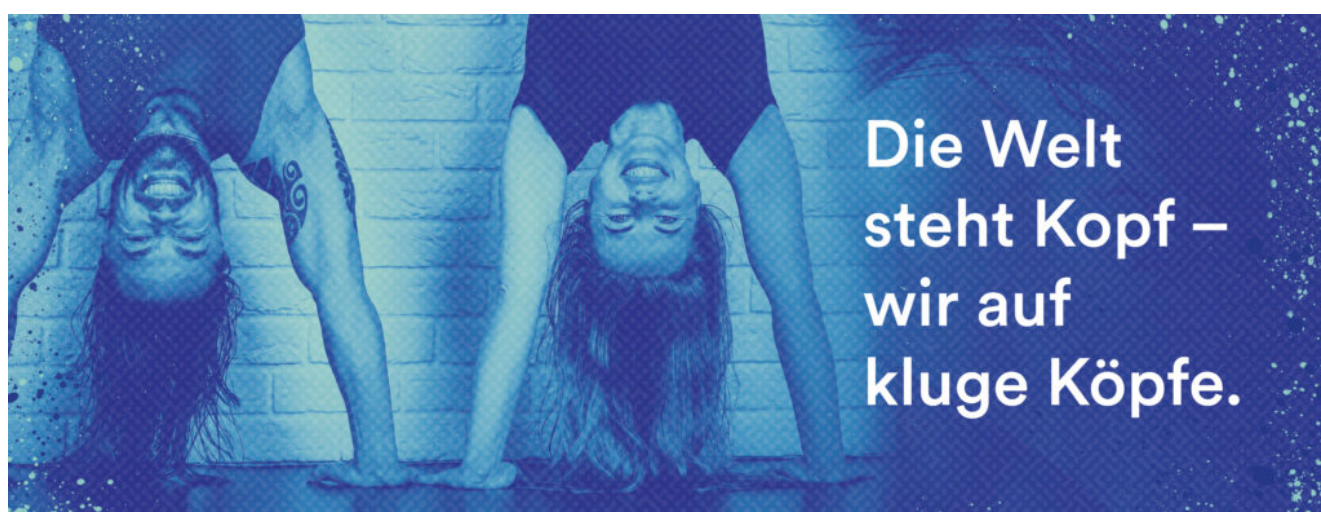
Christoph Drosten

Head of IT-Operations

c.drosten@thalia.de

DevOps Engineer Münster (m/w/d)

DevOps Engineer / Linux Administrator (m/w/d)



Bücher eröffnen Welten – wir tun es ihnen gleich. Als Deutschlands größter Sortimentsbuchhändler verknüpfen wir echte Kundennähe mit richtungsweisender Digitalinnovation. Wir denken das Buch als Erlebnis – ob vor Ort, online, als App oder über den e-Reader. Und versammeln rund 4000 Expertinnen und Experten mit einer Mission: Ein Kulturgut auf allen Kanälen erfahrbar zu machen.

Für unsere Zentrale in **Münster** suchen wir **ab sofort** einen

DevOps Engineer / Linux Administrator (m/w/d)

Welche Aufgaben erwarten Dich:

- Du bist Teil unserer **agilen Entwicklerteams** und bringst dein **Operations-Know How** ein um so eine Plattform für **qualitativ hochwertige Anwendungen** zu ermöglichen
- Du unterstützt die eigenverantwortlichen Teams bei der **Bereitstellung** von Server-, Logging-, Monitoring-, Datenbank-Infrastrukturen
- Du unterstützt die Teams bei der Verbesserung der **Continuous Delivery** Pipelines, automatisierten Tests und automatisierten Deployments
- Du hilfst bei der Einführung von **Kubernetes & Docker** und treibst die Automatisierung kontinuierlich voran
- Du hilfst ein Entwickler Team in ein **DevOps Team** zu wandeln
- Du bringst deine Ideen ein und unterstützt bei der Einführung von **neuen Technologien** und Trends um deine Produkte noch besser, schneller und robuster zu machen

Du willst mehr über die Teams, Technologien und Methoden wissen? Dann schau gerne hier vorbei: <https://tech.thalia.de/tag/peng/>

Was bringst Du mit:

- IT hat dir schon immer Spaß gemacht und ist mehr als nur ein Job für dich
- Du bist in der **Linux Welt zuhause** und hast Lust auf Automation, Infrastrukturentwicklung & Containerisierung
- **Apache, Nginx, Tomcat, Scripting** kennst du gut und Puppet, Ansible und Docker hast du schon mal genutzt oder willst du unbedingt kennenlernen
- Erfahrung mit **Datenbanken** (MySQL), Caching sowie Last- & Performancemessung
- Du kennst die **Zusammenarbeit mit Entwicklern** und **JAVA** ist kein

Fremdwort für dich

- Du kannst Leute von deinen Ideen begeistern. In deiner Welt ist alles **automatisierbar** und du findest Fehler um daraus zu lernen
- Eigeninitiative, **Kommunikationsstärke**, **analytisches Denken** und **sehr gute Deutschkenntnisse**
- Ständige Weiterentwicklung auf technologischem und persönlichem Gebiet

Diese Benefits bieten wir Dir:

Für uns selbstverständlich - Unbefristete Festanstellung mit flexiblen Arbeitszeiten, Remote Arbeit, 30 Tagen Urlaub im Jahr sowie Sonderurlaubstage.

Mit uns weiterkommen - eLearnings und Toolboxen, Coaching, Support und ein Fortbildungsbudget für die individuelle Entwicklung.

Täglich versorgt: Kaffee-, Tee- und Wasserflatrate, vergünstigtes Mittagsangebot in der Kantine sowie reichlich Platz für die gemeinsame Mittagspause.

Wir lernen von- und miteinander: Teamorientierte Unternehmenskultur, Kommunikation auf Augenhöhe sowie hoher Wissensaustausch.

Haben wir dein Interesse geweckt?

Dann bewirb dich unter Angabe deines Gehaltsrahmens und deines frühestmöglichen Eintrittstermins über unser [Online-Portal](#) oder ganz einfach über bewerbungen@thalia.de!

Den Bewerbungsprozess haben wir für dich entspannt und einfach gestaltet. Hier ein paar Einblicke am Beispiel Linux Administrator*in: <https://tech.thalia.de/wie-werde-ich-linux-administratorin-bei-thalia/>

Deine Ansprechpartner*innen:



Phillip Vojinovic

HR Ansprechpartner

p.vojinovic@thalia.de



Christoph Drosten

Head of IT-Operations

c.drosten@thalia.de

Security Engineer / Linux Administrator (m/w/d)



Bücher eröffnen Welten – wir tun es ihnen gleich. Als Deutschlands größter Sortimentsbuchhändler verknüpfen wir echte Kundennähe mit richtungsweisender Digitalinnovation. Wir denken das Buch als Erlebnis – ob vor Ort, online, als App oder über den e-Reader. Und versammeln rund 4000 Expertinnen und Experten mit einer Mission: Ein Kulturgut auf allen Kanälen erfahrbar zu machen.

Für unsere Zentrale in **Münster** suchen wir **ab sofort** einen

Security Engineer / Linux Administrator (m/w/d)

Welche Aufgaben erwarten Dich:

- Im Team **Platform Engineering** entwickelst und implementierst du

technische **Sicherheitsmaßnahmen** sowie **-richtlinien**

- Du unterstützt die **agilen Teams** beim Ausbau von **Sicherheitsmaßnahmen** gegen unbefugte Zugriffe
- Zusammen mit unseren Teams und externen Pentestern jagst und **behebst du Schwachstellen**
- Du bringst deine **Ideen** ein, um unsere Produkte uneinnehmbar zu machen
- Du bleibst am Puls der Zeit, **teilst dein Wissen** mit **den Teams** und **schaffst ein Bewusstsein** für Sicherheitsrisiken

Du willst mehr über die Teams, Technologien und Methoden wissen? Dann schau gerne hier vorbei: <https://tech.thalia.de/tag/peng/>

Was bringst Du mit:

- Du hast eine abgeschlossene **Ausbildung** im Bereich **IT** und bringst Berufserfahrung mit
- Du stellst **Sicherheit** an erster Stelle und liebst es **Sicherheitsstrategien** zu entwickeln
- Dein fundiertes **Linux-** und **IT-Netzwerk Wissen** helfen dir bei Themen wie Serverhärtung, Verschlüsselungen, **IDS** oder **Sicherheitszonen**, um z.B. SQL Injections zu verhindern
- Du hast idealerweise schon mal mit **agilen Methoden** gearbeitet und kannst dich beim Thema **Sicherheit durchsetzen**
- Du zeichnest dich durch **proaktives Arbeiten** sowie eine gute **Teamfähigkeit** aus
- Du hast ein sehr hohes **Verantwortungs- und Sicherheitsbewusstsein** für die entwickelten Produkte

Diese Benefits bieten wir Dir:

Für uns selbstverständlich - Unbefristete Festanstellung mit flexiblen Arbeitszeiten, Remote Arbeit, 30 Tagen Urlaub im Jahr sowie Sonderurlaubstage.

Mit uns weiterkommen - eLearnings und Toolboxen, Coaching, Support und ein Fortbildungsbudget für die individuelle Entwicklung.

Täglich versorgt: Kaffee-, Tee- und Wasserflatrate, vergünstigtes Mittagsangebot in der Kantine sowie reichlich Platz für die gemeinsame Mittagspause.

Wir lernen von- und miteinander: Teamorientierte Unternehmenskultur, Kommunikation auf Augenhöhe sowie hoher Wissensaustausch.

Haben wir dein Interesse geweckt?

Dann bewirb dich unter Angabe deines Gehaltsrahmens und deines frühestmöglichen Eintrittstermins über unser [Online-Portal](#) oder ganz einfach über bewerbungen@thalia.de!

Den Bewerbungsprozess haben wir für dich entspannt und einfach gestaltet. Hier ein paar Einblicke am Beispiel Linux Administrator*in: <https://tech.thalia.de/wie-werde-ich-linux-administratorin-bei-thalia/>

Deine Ansprechpartner*innen:



Phillip Vojinovic

HR Ansprechpartner

p.vojinovic@thalia.de



Christoph Drosten

Head of IT-Operations

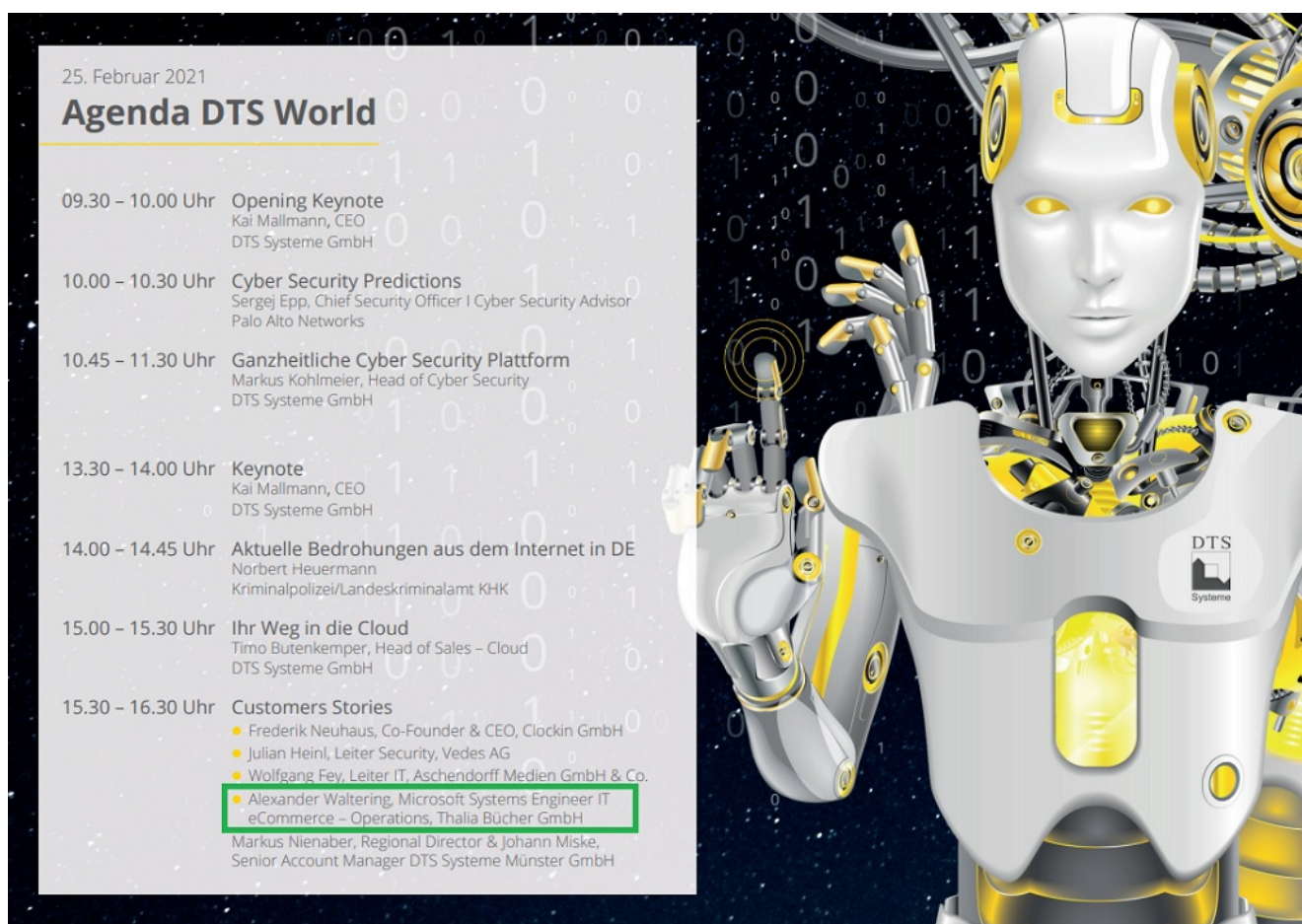
c.drosten@thalia.de

Thalia bei der DTS-World

Am 25.02.2021 findet im virtuellen Showroom die DTS-World unseres Dienstleisters DTS-Systeme statt.

Im Rahmen der DTS-World wird es auch ein Interview mit unserem Kollegen Alexander Waltering zum Thema Security geben.

Wer sich dafür interessiert, der findet hier weitere Information und kann sich auch anmelden: <https://www.dts.de/dts-world-anmeldung.html>



The image shows a promotional graphic for the 'Agenda DTS World' event. On the left, a white box with a grey background contains the event schedule for February 25, 2021. The schedule lists various sessions from 09:30 to 16:30, including an opening keynote, cyber security predictions, a keynote by Kai Mallmann, and a session on cloud security. The 'Customers Stories' section at the bottom lists several speakers, with Alexander Waltering's entry highlighted by a green rectangular border. On the right, a stylized illustration of a white and yellow humanoid robot is shown against a dark background with floating binary code (0s and 1s). The robot has a friendly expression and is gesturing with its hands. A small 'DTS Systeme' logo is visible on the robot's chest.

25. Februar 2021
Agenda DTS World

09.30 – 10.00 Uhr	Opening Keynote Kai Mallmann, CEO DTS Systeme GmbH
10.00 – 10.30 Uhr	Cyber Security Predictions Sergej Epp, Chief Security Officer Cyber Security Advisor Palo Alto Networks
10.45 – 11.30 Uhr	Ganzheitliche Cyber Security Plattform Markus Kohlmeier, Head of Cyber Security DTS Systeme GmbH
13.30 – 14.00 Uhr	Keynote Kai Mallmann, CEO DTS Systeme GmbH
14.00 – 14.45 Uhr	Aktuelle Bedrohungen aus dem Internet in DE Norbert Heuermann Kriminalpolizei/Landeskriminalamt KHK
15.00 – 15.30 Uhr	Ihr Weg in die Cloud Timo Butenkemper, Head of Sales – Cloud DTS Systeme GmbH
15.30 – 16.30 Uhr	Customers Stories - Frederik Neuhaus, Co-Founder & CEO, Clockin GmbH - Julian Heini, Leiter Security, Vedes AG - Wolfgang Fey, Leiter IT, Aschendorff Medien GmbH & Co. Alexander Waltering, Microsoft Systems Engineer IT eCommerce – Operations, Thalia Bücher GmbH Markus Nienaber, Regional Director & Johann Miske, Senior Account Manager DTS Systeme Münster GmbH

Wie werde ich Linux Administrator*in bei Thalia?

Wie sieht eigentlich der Bewerbungsprozess bei einem Buchhändler aus? Was ist Thalia wichtig? Welcher Dresscode wird erwartet? Wird ein Assessment Center durchgeführt?

Transparenz gehört zu unseren Werten, und so haben wir uns überlegt, euch einen Einblick in unseren Bewerbungsprozess zu geben. Das wichtigste vorab: Wir bei Thalia sind auch nur Menschen und freuen uns immer über tolle neue Kollegen*innen.

Der erste Kontakt

Am Anfang steht immer die erste Kontaktaufnahme. Unsere aktuellen Stellenausschreibungen findest du neben unserer Karriereseite z.B. auch auf Stepstone, Xing, LinkedIn oder sonstigen Portalen. Wenn etwas Spannendes dabei ist, dann schicke uns deine Unterlagen entweder direkt oder über die Tools der jeweiligen Seite. Alle Wege führen dich zu den freundlichen Leuten von der Thalia Personalabteilung. Sie werden dich im weiteren Verlauf begleiten, Termine vereinbaren, deine Fragen beantworten, Feedback entgegennehmen/geben usw.

Das erste Vorstellungsgespräch

Ein sehr spannender Moment für uns ist es, dich persönlich im ersten Gespräch kennen zu lernen. Das Gespräch findet im Thalia Bürogebäude in Münster statt

und dauert ca. 1h - 1,5h. Von Thalia sind meistens eine Person aus der Personalabteilung, ein/e Fachspezialist*in und ein/e Teamleiter*in dabei. Keiner der Thalia Mitarbeiter*innen wird Geschäftskleidung tragen. Wir sind in der Regel leger gekleidet. Ziehe einfach etwas an, in dem du dich wohl fühlst.

Nach der Begrüßung werden wir dir als erstes das „Du“ anbieten. Im Bereich IT-Operations - und in vielen anderen Abteilungen von Thalia - duzen wir uns. Warum sollten wir es also in den Gesprächen anders machen?

Zu Beginn werden wir uns vorstellen und etwas über Thalia, den Bereich IT-Operations an den verschiedenen Standorten von Thalia und natürlich auch zu uns selber erzählen. Dann sind wir aber sehr gespannt auf dich! Erzähle uns einfach wer du bist, was du bislang gemacht hast und warum du glaubst, richtig gut auf die ausgeschriebene Stelle zu passen.

Der Übergang in das Fachgespräch ist dann fließend. Wir wollen gerne verstehen, inwieweit du unsere eingesetzten Technologien schon beherrscht bzw. schon mal gesehen hast. Und eines ist sicher: Du wirst nicht alles können und beherrschen. Und das ist auch OK. Es gibt nichts, was man nicht lernen könnte. Also hab den Mut auch mal zu sagen, dass du etwas noch nicht kennst oder in einer Technologie noch Lücken hast.

Es geht uns aber nicht nur um Technologie. Besonders wichtig ist uns auch, wie sehr du mit dem Team harmonierst. Was nützt uns der/die beste Techniker*in der Welt, wenn es im Team nicht funktioniert? Nicht zuletzt ist es auch für dich sicherlich wichtig, dass du dich im neuen Team wohl fühlst. Spaß an der Arbeit ist für uns keine Floskel.

Ein weiterer wichtiger Punkt ist bei uns die Methodik. Im eCommerce von Thalia setzen wir auf agile Methoden. Auch wenn es hilfreich ist, schon mal was über Agilität gehört zu haben oder eine grobe Idee zur DevOps Kultur zu haben, so musst du kein SCRUM Master sein, um dich zurechtzufinden. Wenn du bei Thalia startest, dann wird es u.a. einen Workshop zum Thema „Agilität“ geben.

Im Gespräch stellen wir viele Fragen zu all diesen Themen. Natürlich kannst du auch schon während des Gesprächs deine stellen. Sollten dennoch Fragen bei dir offen sein, so ist nachher noch einmal explizit Zeit, um all deine offenen Punkte zu beantworten.

Nach etwa einer Stunde ist das erste Gespräch auch schon vorbei. Und wir haben einen ersten gegenseitigen Eindruck gewonnen. Genau wie wir solltest auch du den ersten Eindruck verarbeiten und dir mit den neu gewonnen Informationen zu Thalia überlegen, ob du richtig Lust auf den Job hast.

Zeitnah nach dem Gespräch wirst du wieder kontaktiert, und du erhältst Feedback zu der ersten Runde.

Die zweite Vorstellungsrunde

Wenn es weiter geht, dann werden dich in der zweiten Runde weitere Personen kennenlernen wollen. So können z.B. die IT-Leitung oder weitere Team-Mitglieder beim Gespräch dabei sein. Im Gespräch stellen wir weitere Fragen. Auch kann es vorkommen, dass dich in der zweiten Runde eine kleine Aufgabe erwartet, die du natürlich zuhause in Ruhe vorbereiten kannst und im Gespräch vorstellst. Uns ist dabei nicht wichtig, ob die Aufgabe perfekt gelöst wurde. Wir wollen verstehen, wie du mit Aufgaben dieser Art umgehst.

Wenn du möchtest, führen wir dich auch gerne noch einmal durch die Räumlichkeiten, so dass du dir selber einen Eindruck von den potenziellen Arbeitskollegen*innen und vom möglichen neuen Arbeitsplatz machen kannst. Gerne können wir auch einen Schnuppertag vereinbaren.

Mit dem zweiten Gespräch hast du dann auch schon das „Schlimmste“ überstanden. Wenn wir uns einig sind, dann stimmt die Personalabteilung mit dir noch die vertraglichen Rahmenbedingungen ab. Hier geht es dann um Themen wie z.B. Urlaub, Vorteile durch Thalia, Gehalt, Unterschriften und sonstige vergleichbare Themen.

Es geht los!



Ab jetzt freuen wir uns auf den Tag, an dem du startest, und bereiten alles für dich vor. Am Tag 1 bei Thalia begrüßen dich die Personalabteilung und weitere Personen, die zusammen mit dir durchstarten wollen. In einer Willkommensveranstaltung informieren sie dich über die wichtigsten organisatorischen Themen. Danach kommst du in dein neues Team, wo dein vorbereiteter Arbeitsplatz dich bereits erwartet. Während der Einarbeitung in die technische Welt von Thalia begleitet dich ein erfahrener und fester Ansprechpartner aus dem Team. Jetzt kannst du bei Thalia mit deinen Ideen richtig durchstarten.

Hast du Lust den Prozess einmal selber auszuprobieren? ☐

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



Die Corona-Krise hat uns alle früher ins Home Office geschickt, als wir es alle erwartet haben. Es war jedoch keine Überraschung, dass vermehrt Mitarbeiter/innen ins Home Office gehen würden. Daher haben wir schon einige Zeit vorher mit den Planungen begonnen.

„Wer“ war an der Planung beteiligt?

Aber wer ist „wir“ überhaupt für dieses Thema? Bei Thalia haben wir mehrere IT-Teams, die bei der Planung und Vorbereitung unterstützen konnten. So haben wir zunächst überlegt, wen brauchen wir zur Vorbereitung eigentlich. Daraus entstanden ist eine teamübergreifende Gruppe mit Personen aus vier Zentralstandorten. Natürlich war es auch wichtig, die Vorbereitung nicht nur auf die IT zu reduzieren. Was nutzt es, wenn die IT einfach ins Home Office gehen kann, das CallCenter jedoch noch vollständig im Bürogebäude sitzt, weil die Home Office Fähigkeit nicht gegeben ist? Hier war es hilfreich, dass wir grundsätzlich in den Fachbereichen schon eine vergleichsweise hohe HomeOffice

Fähigkeit haben und z.B. auch einige CallCenter Agenten im HomeOffice arbeiten. Es war eine gute Idee, auch die Fachbereiche frühzeitig auf das Unwahrscheinliche einzustimmen.

Was könnte passieren?

Die erste Frage, die wir uns beantworten mussten: Welche möglichen Szenarien wollten wir im Rahmen von Corona vorbereiten?

Letztendlich haben wir zwei Szenarien vorbereitet:

- Einzelne Mitarbeiter/innen gehen ins Home Office (z.B. weil sie einer Risikogruppe angehören).
- Ein kompletter Standort wird geschlossen (z.B. aufgrund einer behördlichen Anordnung).

Das zweite Szenario war für uns gedanklich zwar extrem unwahrscheinlich, jedoch wissen wir heute, dass es eine gute Idee war, auch dieses Szenario schnell vorzubereiten.

Was soll im HomeOffice funktionieren?

Uns war wichtig, dass jede/jeder Mitarbeiter/in im Home Office in irgendeiner Art arbeitsfähig ist. Der Anspruch war dabei eine **grundsätzliche Arbeitsfähigkeit** herzustellen, jedoch nicht jede Spezialanwendung, die nur von 2 Personen genutzt wird, lauffähig zu haben.

Auch war es uns wichtig sicherzustellen, dass die Personen im Home Office miteinander **kommunizieren** können. Im Home Office kann ich nicht einfach mal schnell ins Nachbarbüro gehen oder mal schnell fünf Personen an einen Tisch holen. Themen wie z.B. Telefonkonferenz, Chat, Videokonferenzen usw. mussten auf ihren erhöhten Bedarf vorbereitet werden.

Worauf können wir schon aufbauen?

Wir haben geprüft, auf welche **Infrastruktur** wir an den jeweiligen Standorten bereits jetzt zurückgreifen konnten und wo es noch Lücken gab. Wie konnten wir am schnellsten die Lücken schließen? Welche Skalierung/Vorbereitung wollten wir bereits umsetzen (Beispiel: Erweiterung des VPN IP-Netzes) und welche

Themen wollten wir erst bei Bedarf umsetzen (Beispiel: Bereitstellung von kostenpflichtigen Videokonferenzräumen)?

Müssen wir noch shoppen gehen?

Eines der besonders spannenden Themen war u.a. auch, ob wir eigentlich genug mobile **Client Hardware** hatten, um die Mitarbeiter/innen von Thalia ins Home Office zu schicken. Wenn nicht, wäre es eine Option, private Geräte für die Arbeit im Home Office zu erlauben? Wenn ja, welche (auch rechtlichen) Bedingungen müssten dafür erfüllt sein?

Schon als wir gesehen hatten, was die Corona-Krise in China angerichtet hat, mussten wir uns noch einmal die Frage in Erinnerung rufen, wer eigentlich die Client-PCs, Monitore oder Server **herstellt und liefert**. Spätestens jetzt war es ein guter Zeitpunkt zu überlegen, welche Client- & Server Hardware für die kommenden Wochen – wenn nicht gar Monate – benötigt würden. Woher bekomme ich eigentlich Ersatzteile, wenn in China die Produktion steht und alle Menschen im Home Office sind?

Wir machen nicht alles alleine...

Das brachte uns zur nächsten Frage: Sollten wir mal mit unseren **Dienstleistern** reden, ob auch die einen Notfallplan vorbereitet hatten? Ist eigentlich jemand in unserem „managed Datacenter“, wenn ein Stück Hardware getauscht werden müsste? Zu dieser Zeit haben wir mit vielen Dienstleister gesprochen, ob auch sie gut vorbereitet sind.

Was haben wir erreicht?

Als Ergebnis unserer Vorbereitungen hatten wir eine prall gefüllte Confluence Seite mit vielen Themen, die bereits vorbereitet waren bzw. nur darauf gewartet haben, umgesetzt zu werden.

Einige Zeit später kam es doch schneller und heftiger als wir es alle erwartet haben. Dank unserer Vorbereitung konnten wir jedoch mit vier Stunden Vorlaufzeit die Home Office Fähigkeit für alle Mitarbeiter/innen herstellen. Seither hat sich unsere Arbeitsweise geändert. Darüber berichten wir gerne in weiteren Artikeln dieser Serie.

Mehr zum Thema

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

XPdays: DevOps ist keine Rolle, oder?



Am 8.11.2018 haben wir auf den XPdays unsere Erfahrungen mit der Einführung einer Produkt-Organisation aus Sicht von IT-Operations geteilt.

Was passiert eigentlich mit einem klassischen IT-Betrieb, wenn ein Unternehmen sich für eine Produktorganisation entscheidet? Eines ist klar: Alles verändert sich!

In diesem Beitrag wollen wir euch auf unsere spannende Reise vom IT-Betrieb hin zu Platform Engineering und DevOps mitnehmen. Woher kommen wir? Wie haben

wir uns technologisch, kulturell, prozessual und als Team verändert um optimal mit und in agilen Produktteams zusammenzuarbeiten? Wie unterstützen wir unsere Entwicklungsteams als zentrales Plattform Team aber auch als Mitglied eines crossfunktionalen Produktteam? Was passiert mit unserem Mindset „Never change a running system“ in einer Welt mit 1.000+ Deployments pro Monat? Wie haben wir uns organisiert und wie versuchen wir täglich aufs Neue das Beste aus einer agilen Produktentwicklung herauszuholen? Wie verstehen wir DevOps? Ist das, was früher mal der IT-Betrieb war, auch zu einem crossfunktionalen Produktteam geworden?

<https://www.xpdays.de/2018/sessions/150-devops-ist-keine-rolle-oder.html>

Natürlich ist es nicht möglich alle Erfahrungen in einen kurzen Betrag zu packen. Uns hat es aber viel Spaß gemacht unsere Erfahrungen zu teilen und haben uns sehr über die vielen Rückfragen und Gespräche zum Thema gefreut. Sprecht uns auch gerne weiterhin an, wenn ihr euch für das Thema interessiert.



Christoph Drosten

c.drosten@thalia.de



Matthias Wellmeyer

m.wellmeyer@thalia.de

https://www.xing.com/profile/Christoph_Drosten https://www.xing.com/profile/Matthias_Wellmeyer

Hier die Folien zum Beitrag:

Unseren gesamten Reisebericht findet ihr hier:

[Vom IT-Betrieb zu Platform Engineering. Ein Reisebericht \(1/3\)](#)

[Vom IT-Betrieb zu Platform Engineering. Ein Reisebericht \(2/3\)](#)

[Vom IT-Betrieb zu Platform Engineering. Ein Reisebericht \(3/3\)](#)

Vom IT-Betrieb zu Platform Engineering. Ein Reisebericht (3/3)



Vor einer Woche habe ich berichtet, wie wir Basistechnologien, SelfServices und Automaten etabliert haben. Das hat uns dabei geholfen, wiederkehrende Aufgaben zu automatisieren, die Umsetzungsqualität zu erhöhen, die Entwicklungsteams beim Aufbau und Betrieb von neuen Services zu beschleunigen und die Aufwände im IT-Betrieb zu reduzieren. Aber das alles war nur Werkzeug. Auf unserem Weg zum Platform Engineering Team war es auch wichtig, unser Mindset anzupassen. Wir mussten verstehen, wie unsere wichtigsten Kunden, die Entwicklungsteams, denken und was sie brauchen. Auch unsere Prozesse mussten weiter optimiert werden. Es gab also noch mehr zu tun, als nur ein paar Tools zu etablieren. Auch wenn dieser Reisebericht so geschrieben ist, als ob der Mindset-Change als Letztes stattgefunden hätte, so ist das natürlich nicht richtig. Die technologischen und die kulturellen Änderungen im Team fanden mehr oder weniger zeitgleich statt.

Kapitel 3: Mindset, Methoden, Prozesse und mehr...

PENG! Technik ist nicht alles!



Wenn wir schneller werden wollen, dann müssen wir einiges mit SelfService Schnittstellen machen. Cool, verstanden, fertig? Nope, never, auf keinen Fall! Was ist eigentlich mit Mindset, Kultur, Methoden,...? Auf den Konferenzen reden sie immer von DevOps, Scrum, Agile, Kanban Ah, Kanban, da machen wir doch schon was. Und unsere Devs und Ops mögen sich doch auch schon und gehen zusammen Bier trinken. Dennoch fehlt da noch was.

Wir müssen uns also überlegen, wie die künftigen Produkt-Teams und der IT-Betrieb zusammenarbeiten sollten. OK, noch mehr Ziele für den Umbau. Wir müssen nicht-technische Themen wie z.B. **Kultur, Zusammenarbeit, Mindset, Prozesse, Zuständigkeiten und die Schnittstellen zwischen den Teams definieren**. Da wartet einiges an Arbeit auf uns. Um die Veränderung unseres Teams nach innen und nach außen zu verdeutlichen, wollten wir uns auch einen neuen Namen geben. Nach einigen Diskussionen war uns klar: aus „IT-Betrieb“ sollte „Platform Engineering“ oder kurz „PENG“ werden.

Im Rahmen des Aufbaus der Produkt-Organisation wurden wir als Platform Engineering Team sehr früh mit einbezogen. Gute Idee! Das hat uns die Chance gegeben, neben den ganzen Team-, Technik-, Kultur- und Prozessumbauten auch die notwendigen Operations Umbaumaßnahmen mit einzubringen. Warum ist das so wichtig dieses früh und offiziell zu machen? Ich habe drei technisch Beispiele beschrieben, wo dringend Änderungen notwendig sind und SelfServices etabliert werden müssen, um schneller zu werden. Das macht man nicht mal eben so nebenbei. Notwendig dafür sind größere Investitionen und Anschaffungen in die Infrastruktur. Es muss bewertet werden, ob man SelfServices selber erstellen will oder irgendwo einkauft (Kosten/Nutzen). Zum Selberbauen brauchen wir mehr Personal, welches temporär extern beschafft und bezahlt werden muss. Auch die Themen Kultur, Zusammenarbeit, Mindset, Prozesse, Zuständigkeiten und Schnittstellendefinition zwischen Teams brauchen einiges an Zeit und sollten möglichst zusammen mit den Umbauten hin zu Produkt-Teams erfolgen.

OK, Bestandsaufnahme: Wir wissen wir sind gut, aber nicht schnell genug -> Handlungsbedarf. Wir haben uns technische und nicht technische Ziele gesetzt. Die Transformation von IT-Betrieb zu Platform Engineering kann beginnen!



Um die Transformation zu unterstützen, haben wir im Büro eine „Transformation Wall“ erstellt mit den Zielen, Regeln, Infos und was sonst mit der Transformation zu tun hat. Diese Wand lebt, hat immer die aktuellsten Themen wie z.B. die Maßnahmen einer Retro. Jeder, der ein Thema hat, darf dieses auch auf die Wand pinnen, so dass wir bei der nächsten Gelegenheit darüber reden können. Kurz gesagt: Die Wand **begleitet unsere Transformation und erinnert uns auch jeden Tag daran, was wir tun wollen und warum.**

Setting the Stage: Agile Knowledge (the Basics)



Alle redeten über DevOps, Agile, SCRUM, Kanban, „You build it, you run it“ und so weiter Doch was ist das eigentlich alles? Es gab reichlich Unwissenheit oder - schlimmer - gefährliches Halbwissen. Lasst uns also eine gemeinsame Wissensbasis schaffen, um all die umherschwirrenden Buzzwords zuordnen und verstehen zu können. Auf dieser Basis können wir dann alle anderen Themen aufbauen.

Vor einigen Monaten hatte ich das Glück, einen Zwei-Tages „Agile Mindset“-Workshop zu besuchen, der Grundlagen zu SCRUM, Kanban, Agilität, agiles Manifest, agile Werte usw. vermittelt hat. Während des Workshops, der angereichert war mit kleinen Übungen, habe ich mir die ganze Zeit überlegt, wie ich das ins Team transportiert bekomme. Am Ende des ersten Tages war die Antwort klar: Gar nicht. Zusammen mit dem Agile Coach, der u.a. einen Operations Background hatte, haben wir uns dann überlegt, wie wir diesen Workshop für unser IT-Betrieb-Team abhalten können. Nach kurzer Zeit stand die Planung, und wir haben zwei echt tolle und spannende Workshop-Tage gehabt. Im Anschluss konnten wir viele „Aha Momente“ verzeichnen und hatten als IT-Betrieb ein recht gutes Basis-Verständnis zu den agilen Methoden, der Idee dahinter, den Unterschieden und auch den Vor- und Nachteilen. Klar waren wir weit ab, agile Spezialisten zu sein, aber wir hatten einen super Werkzeugkasten erhalten, der uns auf den weiteren Wegen sehr geholfen hat.

Ausprobieren: Funktioniert SCRUM für Operations?



So, Workshop fertig, jetzt sind wir agile! Nicht wirklich. Uns war klar, wir durften gerade einmal am Inhaltsverzeichnis schnuppern. Die wirkliche Arbeit wartete nun noch auf uns. Wir als Thalia verkaufen ja jede Menge Bücher, aber ein Buch „Agile Operations @ Thalia“ hatten selbst wir als Buch-Spezialist nicht. Vielleicht kommt das später noch ☐ Mit anderen Worten: Wir haben eine Idee bekommen, welche Werkzeuge es gibt. Welche Werkzeuge uns bei unserer Arbeit wirklich helfen (und nicht einfach nur Hip sind), mussten wir in der Praxis selber ausprobieren.

Wir hatten eine Menge über SCRUM, Kanban und Co. erfahren. Kanban hatten wir bereits einige Zeit (zumindest im Ansatz) im Tagesgeschäft praktiziert. Nun wollten wir SCRUM etwas näher kennenlernen. Zusammen mit einem SCRUM Master haben wir unser Projekt zur Einführung der Automatischen Service Bereitstellung (ASB) kurzerhand von einem klassischen Wasserfall-Projekt nach SCRUM umgestellt. Dazu haben wir die notwendigen Meetings aufgesetzt, ein Backlog angelegt und gepflegt, Rollen besetzt, Sprints geplant, durchgeführt, reviewed, verbessert usw... Gerade am Anfang haben wir uns sehr schwer getan. Ganz besonders das Review als auch das Schneiden, Schätzen und Planen von Tickets brauchten einige Übung. Wir haben viel ausprobiert, was gehen könnte. Schätzen wir z.B. Aufwände in Personentagen, Story Points, Anzahl Tickets, ...? Wir machten unsere Erfahrungen und fanden raus, was gut funktioniert, aber auch, was wir besser nicht machen sollten. Und so wurde es von Sprint zu Sprint leichter und nutzbringender. Das kleine Test-SCRUM-Team berichtete von erhöhter Transparenz, klarerer Struktur und ruhigerem Arbeiten im Sprint. Der Produkt Owner hatte einen klareren Blick auf was gemacht wurde, was wann kommen kann und hatte die Möglichkeit zu entscheiden, welches Feature er wann haben wollte. Unterm Strich war es anfangs sehr ungewohnt, jedoch sehr spannend und hilfreich. Wir haben es also geschafft, ein Operations-Projekt nach SCRUM zu führen. Die Erkenntnis stimmte uns positiv ☐

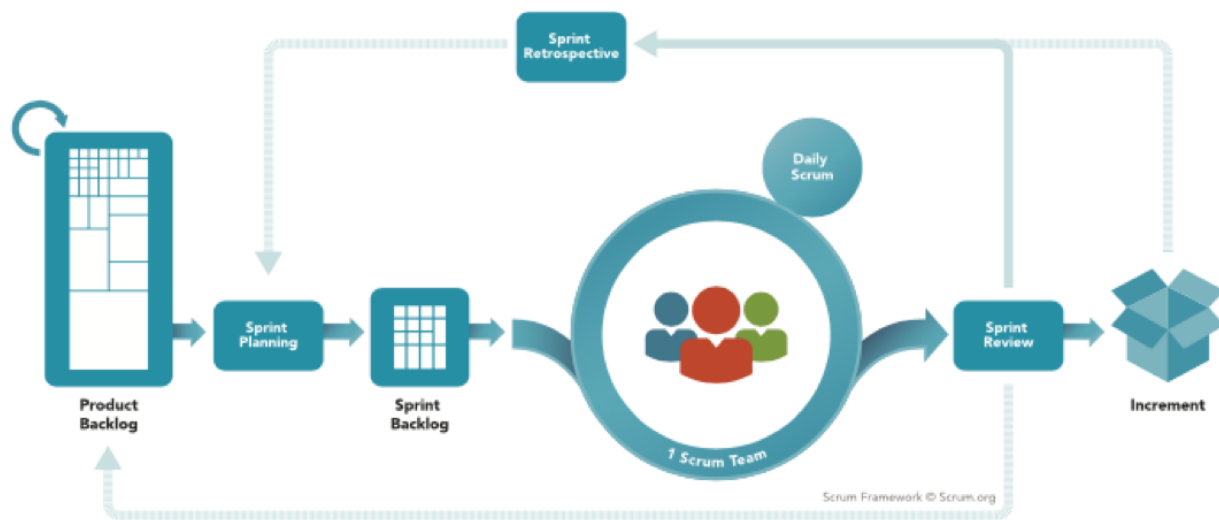
Ein Problem gab es jedoch noch: Unser SCRUM-Testballon wurde in einer reinen Projektumgebung ohne Operations-Tagesgeschäft durchgeführt. Hier gab es keine größeren Störungen, keine spontanen und dringende Anforderungen. Im Operations Tagesgeschäft wimmelt es nur so von unerwarteten Änderungen, worauf wir teilweise hochflexibel reagieren müssen. Leider bekomme ich kein Verständnis, wenn wir die Produktions-Störung des Webshops erst im nächsten Sprint in einer Woche bearbeiten. OK, kann ich verstehen, passt aber nicht so super zu SCRUM. Für das Operations-Tagesgeschäft funktioniert 100% SCRUM für uns also nicht so gut.

Nicht Hip, aber hilfreich: An SCRUM orientieren ohne SCRUM zu machen

Warum sind wir nicht einfach bei Kanban geblieben? Das passt doch viel besser zu so unerwarteten Themen. Nun ja, wir fanden einige Elemente von SCRUM sehr spannend und hilfreich. So ist es hilfreich, dass wir uns alle zwei Wochen verbindlich zusammensetzen, um Aufgaben zu planen. Die Backlogfunktion im JIRA SCRUM-Board finden wir super, eine Retro sowie ein Review hilft uns besser zu werden, auch das Messen des Erreichten ist für uns und unsere Planung hilfreich. Das ist ein Auszug, warum wir uns aktuell an SCRUM orientieren. Klar kann es sein, dass es irgendwann Gründe gibt, wieder Kanban zu machen. Im Moment sind wir jedoch glücklich und vor allem sehr transparent. Wir haben auch gelernt, dass wir nun die gleiche Sprache sprechen wie die Produkt-Teams. Folgende Unterhaltung soll es verdeutlichen: [Dev] „Kannst du bitte folgende Ops Aufgabe diese Woche noch für mich machen? Ich erreiche sonst unser Sprintziel nicht“ - [Ops] „Das kommt etwas überraschend, um dir zu helfen müsste ich unseren Ops Sprint verändern und eine geplante Aufgabe entfernen um deine Aufgabe zu erledigen“ - [Dev] „Oh, das ist ja nicht so gut. Nein, dann plane meine Ops Aufgabe bitte in den nächsten Sprint ein. Ich kann im Zweifel warten“. Hurra, ohne das Wort „Nein“ haben sich Ops und Dev auf eine Verschiebung einer Aufgabe geeinigt und sind dabei auch noch zufrieden.

Wir wollten also SCRUM nicht nur für Projekte nutzen, sondern auch für unser Tagesgeschäft. Wie haben wir das gemacht? Erneut mit viel Ausprobieren, Überprüfen, Lernen, Bessermachen und wieder Ausprobieren. Nach einiger Zeit haben wir uns auf folgendes Setup geeinigt:

SCRUM FRAMEWORK



This image is the copyrighted material of Scrum.org. The original can be found here: <https://www.scrum.org/resources/scrum-framework-poster>

Wir haben ein **Product Backlog**, dieses ist jedoch nur rudimentär priorisiert. Bislang kommen wir mit den Themen im Sprint gut aus :-). Alle zwei Wochen führen wir ein **Sprint Planning** durch. Wir planen dabei etwas weniger als 50% der Tickets ein, die wir schaffen können. Wir orientieren uns dabei an der Anzahl der Tickets. Wir orientieren uns nicht an genaueren Aufwänden oder Story Points (siehe auch „#noEstimation-Bewegung“ unter „Messen“). Zum einen messen wir, dass die Anzahl an Tickets im Schnitt erstaunlich konstant ist. Zum anderen würden wir zwar mit Story Points o.ä. genauer werden in der Planung, jedoch würde sich unser Planungsaufwand deutlich erhöhen. Für uns ein ungünstiges Aufwands/Nutzen-Verhältnis. Nach dem Sprint Planning starten wir den Sprint und haben ein **Sprint Backlog**. Dieses ist aufgrund der vielen Änderungen (ungeplante dringende Anforderungen oder Störungen) jedoch hoch dynamisch. Unser **Scrum Team** besteht aus Linux-, Microsoft-, Datenbank- und Security-Spezialisten. Diese arbeiten sehr eng zusammen. Das Erweitern der Wohlfühzone ist ausdrücklich erwünscht ohne den Anspruch, Spezialist in jedem Bereich zu werden. Ein richtiges **Daily Scrum** nach Lehrbuch machen wir nicht, jedoch haben wir ein Daily Standup, um den Tag zu planen und um jedem den Raum für Fragen oder für Unterstützung zu geben. Alle zwei Wochen vor dem Planning führen wir ein nicht öffentliches **Sprint Review** durch. Warum nicht öffentlich? Wir bieten allen interessierten den „Blue-Board“ Termin (Details folgen □) an, um

sich über unsere Arbeit zu informieren. In unserem Sprint Review bringen wir noch einmal auf den Punkt, was wir erreicht haben und was wir bewusst im letzten Planning raus gelassen haben oder was nicht während des Sprints aufgenommen wurde. Dann prüfen wir zwei Wochen nach der Entscheidung, ob unsere damalige Entscheidung heute immer noch richtig ist. Haben wir das Richtige getan? Es ist kein Blame-Game! Es geht darum zu lernen, um künftig bessere Entscheidungen treffen zu können. Direkt nach dem Sprint Review führen wir eine **Sprint Retro** durch. Was lief gut, was lief schlecht? Welche Maßnahmen leiten wir ab? Diese Maßnahmen landen dann wieder auf unserer Transformation Wall.

Blue Board: Die Fäden zusammenhalten

Unser SCRUM-Board hilft uns in der täglichen Arbeit mit den vielen kleinen Aufgaben. Jedoch ergibt es auch Sinn, einen Schritt zurückzugehen, um auch die größeren Themen zu betrachten, die viele dieser kleineren Aufgaben zusammenhalten. Auf einer höheren Flugebene machen wir größere Themen sichtbar. Größere Themen können dabei unternehmensweite Projekte, Operations Projekte aber auch Penetration Test sein. Es sind also Themen, die uns längere Zeit begleiten und Kapazitäten binden. Für jedes Thema haben wir eine gelbe oder eine grüne Karte. Gelbe Karten sind für größere Themen aus dem Tagesgeschäft (Replacements, Updates, Security Scans ...) - also Themen, die wir machen müssen, um eine stabile Plattform zu gewährleisten. Grüne Karten sind für größere Themen, die unsere Produkte weiterentwickeln und Mehrwert generieren. Die Themen landen dann an einem physikalischen Board. Lustigerweise ist das ein Kanban-Board, welches wir aufgrund der Board-Farbe ganz kreativ „Blue-Board“ getauft haben. Alle zwei Wochen gibt es zusätzlich zu den Daily Standups eine größere Runde für 60 Minuten, wo wir über den Status der Themen informieren. Diese Runde ist öffentlich, so dass Vertreter aller Produkt-Teams sehen, was gerade bearbeitet wird und welches Thema wann kommt. Sie können sich informieren, ihre Fragen loswerden und Feedback geben.



„Blue-Board“ für die Transparenz und Planung von größeren Operations Themen

Definieren unserer Produkte

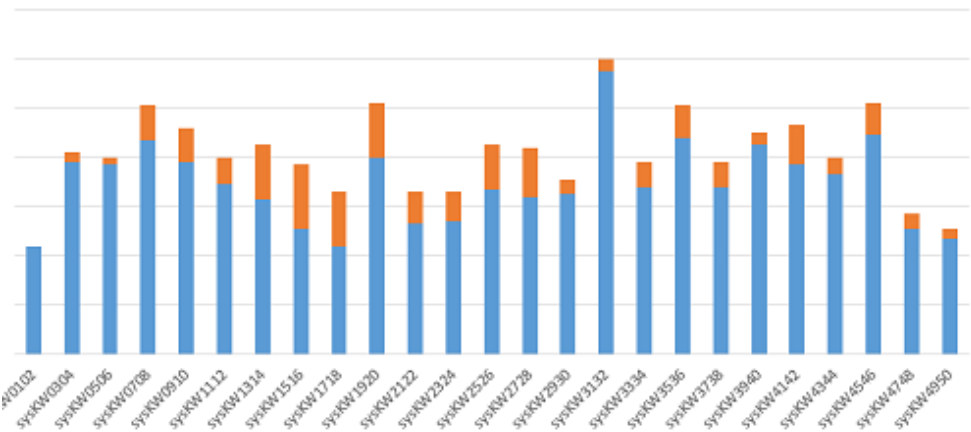
Wir wollen ein Produkt-Team sein. Aber was ist eigentlich unser Produkt? Mit dieser Frage haben wir uns lange auseinandergesetzt. Ist ein Loadbalancer schon ein Produkt? Dann haben wir ja hunderte Produkte. Wenn wir es zusammenfassen, ist dann „die Plattform“ unser Produkt? Hmmm, die Definition hilft nicht so richtig weiter bei der Planung, Strukturierung und Messung. Mit Hilfe unseres Agile Coach haben wir definiert: Ein Produkt ist all das, was einem anderen Team ein Mehrwert bietet. Etwas, wofür jemand im Zweifel auch Geld bezahlen würde.

Und so sind wir losgezogen und haben nach und nach Technologien zu Produkten zusammengefasst, die es uns erlauben, zu strukturieren und sowohl uns als auch allen anderen zu verdeutlichen, was wir eigentlich alles machen. Dabei herausgekommen ist folgende Übersicht:



So haben wir z.B. den Loadbalancer, Firewall, Storage, Netzwerk, Rechenzentrum ... zum Produkt **Basis Infrastruktur** zusammengefasst. Wir bieten den anderen Teams unser Produkt **Notdienst** an, welches die Teams nutzen können. Wir bieten den Teams als Produkt eine **Alarmierungs & Monitoring**-Infrastruktur an, die sie nutzen können. Und wie in jedem guten Haushalt so gibt es bei uns auch einen Bereich „Sonstiges“ mit dem klangvollen Produktnamen **3rd Party Service**. Da wir leider noch nicht alle Teams mit einem eigenen Operations-Spezialisten versehen konnten, machen wir für einige **Touchpoint Teams** noch den Operations-Teil.

Messen: Die Basis für eine gute Planung

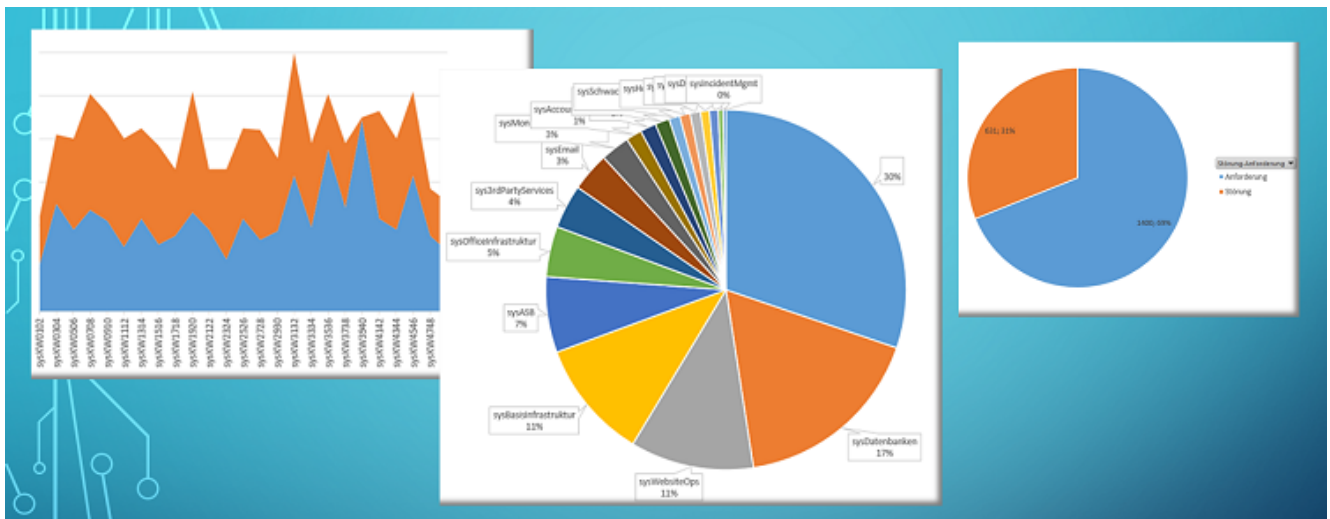


Anzahl der geschlossenen Tickets je Sprint in 2017

Anfangs haben wir überlegt, wie wir unsere Arbeit messen wollen und wie wir Aufwände planen können. Auch brauchten wir eine Möglichkeit zu prüfen, ob unsere Maßnahmen einen positiven Effekt haben. Weiter wollten wir wissen, wie viele Tickets wir in einen Sprint nehmen können. Hier haben wir einiges ausprobiert, verworfen und neu probiert. Um es möglichst einfach zu halten, um möglichst wenig Aufwand zu investieren, haben wir uns darauf geeinigt, die Anzahl der geschlossenen Tickets je Sprint zu messen. Natürlich ist uns klar, dass ein Ticket 5 Minuten oder 5 Stunden dauern kann. Jedoch haben wir festgestellt, dass bei uns die Anzahl der Tickets, die wir schließen, relativ konstant ist und somit eine Größe ist, mit der wir arbeiten können. Klar können wir die Genauigkeit weiter erhöhen, indem wir z.B. auf Storypoints gehen würden, jedoch scheuen wir den Mehraufwand für die Bewertung. Auch erscheint uns der Mehrwert, den wir damit erzeugen, zu gering. Auf der Basis der Anzahl geschlossener Tickets haben wir dann eine Anzahl Tickets abgeleitet, die wir fest in den Sprint einplanen. Leider liegt dieser Wert noch bei unter 50% der Tickets je Sprint. Das Problem ist leider immer noch, dass uns immer wieder ungeplante Aufgaben erreichen, die ihre Berechtigung haben. Bislang funktioniert die Planung basierend auf der Anzahl der Tickets recht gut und stabil. Es gibt auch noch keine Bemühungen, den Aufwand für die Planung (z.B. mit Storypoint) zu erhöhen. So sind wir mehr oder weniger geplant zu Anhängern der #noEstimation-Bewegung geworden.

Neben der Anzahl der Tickets, die uns erreichen, messen wir aber noch mehr – z.B. wie viel Tagesgeschäft und wie viel Weiterentwicklung im Sprint steckt. Wir messen, woher die Tickets kommen, für welches unserer Produkte wir die Arbeit leisten, wie viele der ursprünglich für den Sprint geplanten Tickets tatsächlich

auch fertig werden, das Verhältnis Störung zu Anforderung ...



Eine Auswahl an Messungen

Zur Messung exportieren wir die geschlossenen Tickets aus JIRA nach Excel, wo wir sie dann via Pivot mit wenig Aufwand in ein paar Graphen visualisieren. Um die Tickets auswerten zu können, nutzen wir u.a. drei JIRA Stichwörter: (1) Ein Stichwort für die Kalenderwochen, in denen das Ticket bearbeitet wurde, (2) Ein Stichwort, um das Ticket einem unserer Produkte zuzuordnen und (3) haben wir ein Stichwort, um ein Ticket als Entwicklungsticket zu kennzeichnen (ohne dieses Stichwort ist es ein Tagesgeschäftsticket).

XFT - oder der Microsoft Spezialist der MySQL DBs auf Linux installiert

Super, jetzt haben wir ein wenig „agiles Mindset“, einen Agile Coach und ein paar Boards. Wir können nun priorisieren, um das Richtige zur richtigen Zeit tun. Aber was, wenn der Spezialist für das Richtige gerade im Urlaub ist? Im Team waren wir so organisiert, dass wir für Linux-, Datenbank-, Security- und Windows-Themen Spezialisten haben, die ihr Handwerk wirklich gut verstehen. Dadurch hatten wir jedoch, wenn man so will, vier Teams in einem Team. Diese künstlichen Hürden wollten wir entfernen und dazu die Wohlfühlzonen, die wir hatten, erweitern. Warum sollte nicht der Windows-Spezialist eine MySQL-Datenbank auf einer Linux-Maschine installieren, wenn das gerade die wichtigste

Arbeit ist? Als kleiner Nebeneffekt hat er seine Wohlfühlzone und sein KnowHow erweitert. Zudem macht es auch noch Spaß, neue Sachen zu versuchen. Aber funktioniert das auch im Alltag? Um das herauszufinden, haben wir jeden Ausflug raus aus der Wohlfühlzone auf einer einfachen Matrix gemessen und in der Retro besprochen. Die Reaktionen im Team waren ziemlich positiv, und die Fokussierung auf die umzusetzenden Themen im Sprint hatte positive Auswirkungen auf die Bearbeitungsgeschwindigkeit. Grenzt das nicht an ein Cross-Functional-Team (kurz XFT)? ☐



Erweitere deine Wohlfühlzone. Wer hat mit wem zusammengearbeitet?

DevOps ist keine Rolle, oder? Operations KnowHow in die Produkt-Teams

Dank der guten und nicht ganz einfachen Arbeit der Kollegen wissen wir bereits, welche Produkt-Teams es gibt und welche Software-Services sie entwickeln und

betreiben sollen. Auf dieser Basis konnten wir Team-Mitglieder suchen, die Lust auf ein neues Abenteuer haben. So haben wir Linux Spezialisten gefunden, die bereit waren, zwei oder auch drei (je nach Teamgröße) Produkt-Teams zu betreuen. Aber was bedeutet das? Um die Teams zu betreuen, sind die Linux Spezialisten sowohl fachlich als auch physisch in die Produkt-Teams gegangen und fester Bestandteil der Teams geworden. Hurra! Endlich haben wir unseren eigenen Linux Spezialisten, der alle Operations Aufgaben erledigen kann! Nope! Die Idee ist, dass er zwar der Operations Spezialist im Team ist. Jedoch ist seine Aufgabe, sein Operations-KnowHow so weit wie möglich zu verbreiten und das Team in die Lage zu versetzen, eigenständig alle für ihr Produkt notwendigen Operations Aufgaben umzusetzen. Wir erinnern uns: Wenig Reibungsverlust an Schnittstellen. Möglichst viel selber machen. Gilt auch im Kleinen. Wollen wir nun z.B. die Entwickler zu Operations Spezialisten machen? Nein! Wir wollen, dass die Teams in der Lage sind, alles, was nötig ist, selber zu machen. Im Optimalfall gibt es einen einfachen SelfService mit einer Entwickler Schnittstelle, die man mit geringem KnowHow bedienen kann. Die ganze Operations Magie passiert dann automatisch im Hintergrund. Um zu verstehen, wie die optimale Schnittstelle zwischen Platform Engineering und dem Produkt-Team aussieht, haben wir zusammen mit den Linux Spezialisten in den Teams definiert, was in den Teams in welcher Tiefe passieren soll und wo Platform Engineering eine einfache Schnittstelle bereitstellen muss. Auch wenn jede Schnittstelle im Optimalfall perfekt mit einer großartigen GUI oder API automatisiert ist, so ist das besonders zum Start nicht realistisch. Im Zweifel kann eine Schnittstelle auch ein Ticket oder eine gute Dokumentation sein. Automatisiert aber so schnell und so früh wie möglich die Schnittstellen. So, nun sind die Linux Spezialisten in den Produkt-Teams, weg, raus aus dem Platform Team. Und wie bleiben die Ops Spezialisten in den Teams nun am Operations Puls der Zeit? Das ist recht einfach. Zum Einen nehmen sie weiter an den bereits etablierten Meetings teil.



Unser Daily Standup mit Teilnehmern vor Ort, in Hagen, Berlin und im Homeoffice.

So können Sie an den Daily Standups oder an den alle zwei Wochen stattfindenden „Blue Board“ Meetings teilnehmen (da wo wir die größeren Themen und deren Priorität und Fortschritt besprechen). Das ist alles nice und hilft. Besonders wichtig ist jedoch ein neu geschaffenes Meeting. Noch ein Meeting? Och nöö! Doch, ergibt Sinn. Tut nicht wirklich weh, bringt dafür aber viel. Neu etabliert haben wir die „Operations Community of Practice“ – kurz: OpsCoP. Hier treffen sich alle zwei Wochen die Ops Spezialisten aus allen Produkt Teams sowie ein Vertreter aus dem Platform Engineering Team. Dort tauschen wir unsere Erfahrungen zwischen den Teams aus. Was kann ich aus Team A lernen und für Team B übernehmen? Welche Entwicklung passiert gerade im Platform Team oder welche Known Issues gibt es? Und so weiter

Irgendwie ist es dann doch noch passiert, das unsere Linux-Spezialisten in den Produkt-Teams nur noch „die DevOps“ genant werden. Im Grunde total falsch! DevOps ist keine Rolle. Wir haben sogar einen Versuch gestartet das wieder raus zu bekommen. Erfolglos. Die falsche Bezeichnung DevOps erfreut sich großer Beliebtheit bei uns. Nun lassen wir es einfach so, jedoch immer mit der Ergänzung dass es eigentlich falsch ist ☐

Die Schnittstelle zwischen Platform Engineering und den Produkt-Teams



Cool, jetzt haben wir die Produkt Teams mit Operations KnowHow ausgerüstet, verteilen dieses, machen die Teams dadurch schneller. Auch die Schnittstellen sowie der Informationsfluss zwischen Platform Engineering und Produkt Teams ist geklärt. Aber was machen eigentlich die Operations-Kollegen, wenn die Produkt Teams ihre Sachen selber betreiben? Wir brauchen ein neues Ziel. Früher war unser Ziel, alle Systeme und Anwendungen 7x24h zu betreiben. Nun machen das (zumindest in großen Teilen) die Produkt Teams. Die Antwort ist einfach, die Angst völlig unberechtigt. Wir sind inzwischen kein echter IT-Betrieb mehr. Da war doch was. Hatten wir uns nicht umbenannt? Platform Engineering.... Was ist das eigentlich? Wir engineeren (gibt es das Wort im deutschen überhaupt? Sorry) eine Plattform für alle Services. Aber gehört alles zur Plattform? In großen Teilen helfen uns da die Schnittstellen, die wir zusammen mit den Produkt Teams definiert haben. Alles was ein Produkt Team braucht, um einen Service zu bauen und zu betreiben, machen die Produkt Teams. Alles andere macht das Platform Engineering. Beispiel: Das Produkt-Team braucht einen Knopf, aus dem ein Server mit einem Service rauskommt. Den Knopf mit all seiner Logik und seinen Automatismen baut das Platform Engineering Team. Ein neuer Blickwinkel. Wir bauen und betreiben nun Services, die es erlauben, Server zu erstellen. Den erstellten Server betreibt nun jedoch jemand anderes. Oder ein anderes Beispiel aus dem Bereich Datenbanken: Wir definieren nun, welche MySQL Versionen automatisiert bereitgestellt werden. Wir definieren verpflichtende Sicherheitskonfiguration. Aus unserer Erfahrung schlagen wir Best Practice Konfigurationen vor, die von Produkt Teams genutzt werden können, nicht müssen. Wir stellen getestete Patches und zugehörige Dokumentation bereit. Die Produkt Teams müssen jedoch all diese Änderungen selber implementieren und betreiben. Dadurch ergeben sich für uns neue Möglichkeiten. Endlich haben wir eine Chance, uns auf neue Innovationsthemen zu stürzen. Wir haben die Chance, die Automation weiter auszubauen. Wir als

Platform Engineering verstehen uns als Anbieter von Technologie für die Produkt Teams, ohne diese auf unsere Technologie limitieren zu wollen. Warum sollte ein echt schnelles Produkt-Team Monate darauf warten, von uns eine Technologie zu bekommen? Warum sollte nicht auch ein Produkt-Team zusammen mit dem Linux Spezialisten in Abstimmung mit dem Platform Engineering Team eine neue Technologie evaluieren und diese Technologie dann über das Platform Engineering Team allen anderen Teams zur Verfügung stellen? Wichtig ist hier wieder die enge Abstimmung der Bereiche. Sobald mehr als ein Team eine neue Technologie nutzen möchte, so übernehmen wir aus dem Platform Engineering den aktuellen Stand und stellen es allen Teams zur Verfügung. Müssen wir deswegen jede Technologie selber entwickeln? Nein, natürlich nicht. Was spricht dagegen, eine Monitoring Lösung, ein CDN, einen DDoS-Schutz und so weiter einzukaufen und den Produkt-Teams zur Verfügung zu stellen? Warum nicht Technologie aus der Public Cloud einkaufen und allen anbieten. Das macht uns wieder schneller. Es ist immer eine Frage des Preis-/Leistungsverhältnisses. Hier darf man jedoch nicht vergessen, dass auch der billiger aufgebaute eigene Service am Ende sehr viel teurer sein kann als der teuer eingekaufte Service. Ausschlaggebend ist natürlich auch der Faktor „time to market“. Wenn der teuer eingekaufte Service es uns ermöglicht, zwei Monate schneller am Markt zu sein, und somit zwei Monate mehr Umsatz generiert, ist das ein wichtiges Entscheidungskriterium.

Was haben wir gelernt?

- Mache kein SCRUM, nur weil alle es machen. Mache es, weil es dir hilft.
- Sei mutig. Mache Fehler. Lerne.
- Lass dir helfen. Hilfe anderen. Vertraue deinem Team.
- Probiere aus. Laufe vor die Wand, um zu verstehen, was man besser machen kann.
- Reduziere Schnittstellen auf ein Minimum und automatisiere den Rest und noch mehr.
- Sei kein Blocker. Wenn du ein Blocker bist, verändere es.

- Messe, was du tust, und mach die Ergebnisse sichtbar.
- Tue das Richtige zur richtigen Zeit. Mache dazu deine gesamte Arbeit sichtbar und priorisiere sie nach dem Nutzen, den die Arbeit erzeugt.
- Mache sichtbar, was du nicht machst.
- Stelle die Aufgabe in den Mittelpunkt und bearbeite sie, auch wenn du länger brauchst als der Spezialist.
- Der Klassiker: Stop starting, start finishing. Es macht dich langsam, wenn du alle Themen gleichzeitig machst.
- Gib Raum für Eskalationen. Eskalationen sind hilfreich und nicht böse.
- Spreche die gleiche Sprache wie dein Dev Kollege.
- Erweitere deine Wohlfühlzone und habe Spaß dabei.
- Lass dich inspirieren. Gehe auf Konferenzen, rede mit Kollegen aus anderen Unternehmen oder ganz einfach: Rede mal mit deinen Entwicklern ☐

Sicherlich habe ich einige Punkte vergessen. Ich möchte mit diesem Reisebericht jedoch unsere wichtigsten Eckpunkte, Erfahrungen und Gedanken vom IT-Betrieb hin zum Platform Engineering teilen. **Ich würde mich total über Feedback in den Kommentaren freuen.** Was sind eure Gedanken zu unserer Reise? Was sind eure Erfahrungen? Was können wir noch besser machen? Wir sind auch offen für einen persönlichen Erfahrungsaustausch. An dieser Stelle ein großes Dankeschön an Kollegen von Otto.de oder dem LVM für tolle inspirierende Gespräche. Ich hoffe, wir konnten auch etwas zurückgeben.

Alle drei Kapitel im Überblick



[Kapitel 1: Woher kommen wir?
6 Jahre im Schnelldurchlauf](#)



[Kapitel 2: Basistechnologien,
SelfServices & Automation](#)



[Kapitel 3: Mindset, Methoden,
Prozesse und mehr...](#)

