

## „Cloud-native Shift Left“



### Wie erreichen wir das?

- Linux: Entwickeln und Testen auf dem Betriebssystem der Produktion
- Docker: Ausführen der Software-Bausteine der Produktion
- Kubernetes: vollständige Cloud-native Cluster-Umgebungen auch lokal
- Übertrage die Software-Stände der Produktion zurück zu den Entwicklern

### Was tun wir damit?

- Fehler lokal unter voller Kontrolle und mit allen Mitteln analysieren
- Alternative Cluster-Konfigurationen ausprobieren und messen
- Übertragbare Umgebungen verkürzen die Einarbeitung neuer Entwickler

### .. und was gewinnen wir dazu?

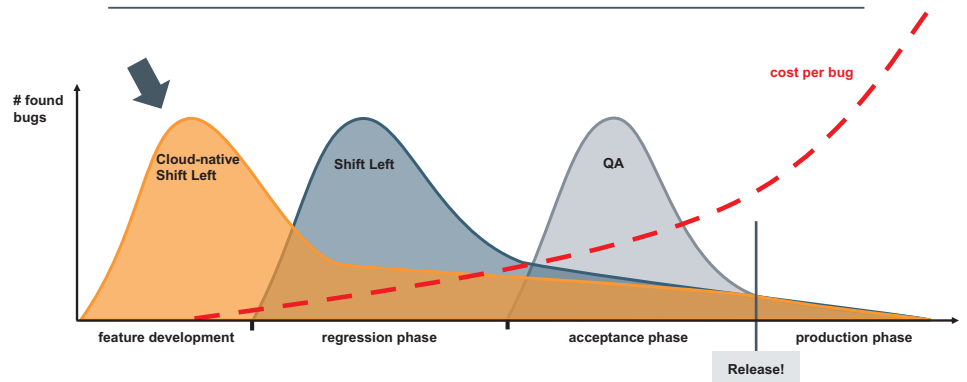
- Die Fähigkeit völlig neue Architekturen auszuprobieren
- Die Unabhängigkeit von Bestellprozessen
- Kosteneffizienz bei Software Development & Engineering
- Zufriedenheit in einem befähigten Team vermindert die Mitarbeiterfluktuation

### Shift Left wird um Cloud-native Techniken erweitert!

Unter „Shift Left“ verstehen wir das Verschieben von Qualitätsrisiken auf einen möglichst frühen Zeitpunkt im Entwicklungs-Lebenszyklus. Moderne Cloud-Techniken wie Container-Technologien und das Everything-as-code-Mantra erlauben uns, diese auch schon im Entwickler-Test einzusetzen.

*Shift Left will den Bug so früh wie möglich aufspüren und argumentiert Testaufwände mit dem dann noch geringeren Preis der Behebung.*

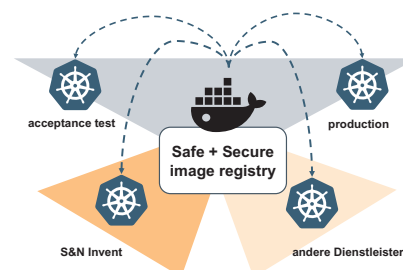
*Cloud-native Shift Left geht bis ganz nach links, dorthin, wo der Code entsteht: auf dem Entwicklerarbeitsplatz sind alle Integrationen testbar zu machen, das Entwickeln in Integrationstests wird zur neuen Norm.*



In der Cloud-nativen Arbeitsweise schieben wir den Testaufwand nach links: Fehler werden also von den Entwicklern direkt bearbeitet. Produktionsfehler werden wieder in die Entwicklungssituation geworfen, wo alle Werkzeuge zur Analyse bereit stehen und die Fähigkeiten, diese einzusetzen, bestens entwickelt sind.

### Eine neue Ökologie in der Software-Entwicklung

Verteilte Software-Architekturen wie Microservices zielen darauf ab, den Entwicklungsprozess auf kleine, besser organisierbare Einheiten aufzuteilen. Jede Einheit wird dazu an ein autonomes Team übereignet. Für große Vorhaben sind mehr als nur ein Dienstleister wünschenswert. Dennoch, das Aufspalten einer singulären Aufgabenstellung in mehrere Einheiten und das Verstreuen auf Teams generieren das Problem der Teilung für jedes einzelne Team.



Open Source-Teams zeigen, wie räumlich getrenntes und zeitlich versetztes Arbeiten auf festen Grundsätzen besteht: es muss einfach sein, dem Projekt beizutreten; Features und Bugs müssen für jeden reproduzierbar sein; Mitarbeit zu unterbrechen ist praktikabel; zentrale Builds führen zum Release, blockieren aber nicht das Entwickler-Team.

Die Image Registry sollte mit allen Komponenten gefüllt sein, die zur vollständigen Lösung dazu gehören, inklusive der Konfiguration und einigen Testdaten. Sie enthält die Grundlagen für autonome d.h. selbstwirksame Entwickler, die ihre Stärken auf den Code anwenden können. Die eingebauten Sicherheitsprüfungen einer Image Registry machen sie zur Drehscheibe einer Secure Supply Chain. CI/CD Pipelines können überall dort entstehen, wo die Teams sie brauchen.

Wir zeigen interessierten Kunden, worauf sie ihre Projektorganisation einstellen und was sie von ihren Dienstleistern erwarten dürfen.

# Unser Angebot für Ihr Team!



## Cloud Team Enabler (CTE)

Cloud-native sollte unsere Projekt-Teams nach vorne bringen.

*Unsere neue Rolle hilft auch in Ihrem Team: der Cloud-Team-Enabler.*

Cloud-native Arbeitstechniken unterstützen Vorhaben mit verteilten Teams, funktionieren aber genauso auch für das einzelne Team. Lernen und Perfektionieren braucht Zeit. Organisches Lernen mit dem ganzen Team schafft nachhaltige Werte. Dazu beraten und unterstützen unsere Cloud-Team-Enabler ihre Entwickler und üben mit ihnen - mit spürbarem Unterschied.

Team-Mitglieder praktizieren und verfeinern Cloud-native Techniken mittels Schulungen und im Pair-Programming. All das im Rahmen des bekannten Projektgeschehens, in Sprints, in denen dies in Arbeitspaketen eingeplant wird.

## Der Arbeitsauftrag für den CTE:

- wird vom Projekt bestellt, üblicherweise für Build-Automatisierung oder Deployment-Tätigkeiten.
- bewertet die bisherige Arbeitsweise und das vorherrschende Skill-Set, beobachtet Störungen und rät zu konkreten Maßnahmen.
- erkennt Neigungen und weist auf Chancen für individuelles Wachstum hin, um aus dem ganzen Team ein Cloud-natives Team zu formen.
- kennt die beliebtesten IDEs und stellt sie auf maximale Wirkung ein.
- weiß vom Wunsch der agilen Softwareentwicklung und sorgt für autonome, cross-funktionale Teams.

## Linux Docker Kubernetes – die Schulung für das Team

Seit 2022 bietet S&N Invent diese Schulung an, darin enthalten sind Einführungen aller Cloud-nativen Techniken. Die Grundlagen von Linux, Docker und Kubernetes werden eingeübt. Jeder Teilnehmer lernt das Übertragen von Umgebungen zu beherrschen. Diese Inhalte richten sich an alle Entwickler, egal ob Frontend, Backend, Middleware oder Full-stack. Am Beispiel gehen wir auf die klassischen Technologien ein, insbesondere Networking. Auf dem eigenen Arbeitsplatzrechner kommt es zum (Erst-)kontakt mit Cloud-Technologie.

Der zweite Tag gehört ganz dem mitgebrachten Beispielprojekt, das in seine Cloud-native Form gewandelt und lokal zum Laufen gebracht werden muss. Es besteht aus Angular Webcomponents, einem Java (Quarkus) REST Service mit Kafka Consumer und Producer, der erforderlichen Middleware Apache Kafka und PostgreSQL.

Zusätzliche Inhalte sind Monitoring, Performance- und Last-Tests, Service Mesh, Verteilte Architektur-Muster und event-driven Modularisierung in weiteren Service-Technologien (Springboot+Java, NestJS+Typescript).

Am Ende besitzt jeder Teilnehmer eine Cloud-native Toolbox auf seinem vertrauten Arbeitsplatz, die sofort im nächsten Projekt eingesetzt werden kann, und dazu das Vertrauen, damit nicht zu zögern.



## Testsysteme in Zeiten von Containerisierung

Zentrale Testumgebungen werden in Form sogenannter Stages für ein Projekt oder sogar für das gesamte Unternehmen bereitgestellt und gemeinsam von mehreren Projektteams genutzt. Auf diesen Umgebungen sind einige Services als produktionsnah eingestuft, beispielsweise das Authentisierungssystem. Damit fehlt hier schon gleich ein Freiraum für den Test dieser Systeme.

Dank der Containerisierung mit der Übertragbarkeit lauffähiger Systeme samt ihrer Konfiguration ist die Zahl möglicher Testumgebungen nur noch durch den verfügbaren Ressourcenrahmen begrenzt. Entwickler-Arbeitsplätze haben nicht selten mehr Leistung als ein einzelner Server. Damit kann ein Entwickler-Arbeitsplatz problemlos für den Betrieb einer vollwertigen, lokalen Testumgebung genutzt werden. Hier hilft auch die Fähigkeit von Containern, nicht nur nach oben sondern auch in eine reduzierte Kapazität zu skalieren.

So können Testumgebungen, die identisch zur aktuellen oder geplanten Produktionsumgebung sind, von jedem Team selber und in beliebiger Anzahl erstellt werden.

Dabei werden zentral genutzte Systeme frei für Abnahme- und Produktionsvorbereitung. Integrationstests werden lokal möglich und die Entwicklung wird unabhängig von Bereitstellungsprozessen.

S&N Invent GmbH  
Klingenderstr. 5  
33100 Paderborn

65760 Eschborn  
Frankfurter Str. 71-75  
+49 (6196) 80269-0

93049 Regensburg  
Heinkelstr. 3  
+49 (941) 30746-0

48155 Münster  
Hafenweg 46-48  
+49 (5251) 1581-0

Zukunftsmühle 2  
33102 Paderborn  
+49 (5251) 1581-0

85609 Aschheim-Dornach  
Otto-Hahn-Straße 20  
+49 (89) 8898103-0

20457 Hamburg  
Willy-Brandt-Strasse 23  
+49 (40) 21045-291

info@sn-invent.de  
www.sn-invent.de

