



WHITEPAPER

DIE PRODUKTIVITÄT IHRES IT-TEAMS MIT **CLOUD DEVELOPMENT ENVIRONMENTS AUF DAS NÄCHSTE LEVEL HEBEN**

Key Facts

- 61 % befragter Entwickler_innen berichten, dass sie zumindest einmal im Monat Probleme in ihrer lokalen Entwicklungsumgebung beheben müssen. 31 % sogar zumindest einmal pro Woche. Das führt zu Produktivitätseinbußen.
- Standardisierungsrichtlinien helfen, es treten trotzdem individuelle Probleme mit lokalen Entwicklungsumgebungen auf. Für rechenintensive Tasks steht nicht immer die benötigte Leistung zur Verfügung. Source Code liegt unsicher am Rechner. Branch-Switching kostet Zeit.
- Eine Lösung dafür sind Cloud Development Environments (CDEs). CDEs sind standardisierte Remote-Umgebungen, in denen Entwickler_innen alle Tools sofort zur Verfügung haben, die sie für ihre Arbeit brauchen.

Einleitung

Softwareentwickler_innen zu finden ist schwierig. Bestehende zu halten nicht einfach.

Diese zwei Faktoren führen dazu, dass

- die Produktivität der eigenen Entwickler_innen optimiert und
- die Developer Experience verbessert werden muss. Entwickler_innen müssen also Werkzeuge und Umgebungen zur Verfügung stehen, die sie gerne verwenden.

Ein großes Hindernis, um diese Ziele zu erreichen, sind lokale Entwicklungsumgebungen. Stunden werden verbracht, diese aufzusetzen und zu warten. Besonders die zu entwickelnde Software lokal zu betreiben und der lokale Build sind eine große Herausforderung.

Cloudomation reduziert diesen Aufwand drastisch. Und zwar mit standardisierten Entwicklungsumgebungen, sogenannten Cloud Development Environments (CDEs). CDEs sind sofort für alle Entwickler_innen verfügbar und individuell konfigurierbar.

Dieses Whitepaper beschreibt, welche Probleme Cloud Development Environments lösen, welche Vorteile CDEs mit sich bringen, wie Sie mit CDEs die Produktivität steigern und die Qualität der Softwareentwicklung in Ihrem Unternehmen verbessern.

Lokale Entwicklungsumgebungen

Softwareentwickler_innen arbeiten normalerweise in ihrer lokalen Entwicklungsumgebung. Da jede lokale Entwicklungsumgebung anders eingerichtet und nicht standardisiert ist, treten sehr individuelle Probleme auf, deren Lösung zeitaufwändig und schwierig ist.

Neben den IDE-Tools und Source Control, die meist schnell aufgesetzt sind, sind viele Abhängigkeiten einzurichten und aktuell zu halten, Tools für lokalen Build zum Laufen zu bringen und meist am herausforderndsten: Die zu entwickelnde Software selbst lokal zu betreiben. Dabei kommen oft selbst entwickelte Tools und Skripte zum Einsatz, die fehleranfällig und schwer zu bedienen sind.

Elijah Bate von theoptimisedweb.com beschreibt das so:

„Das Einrichten neuer Entwicklungsumgebungen oder der Versuch, den lokalen Rechner für verschiedene Projekte zu aktualisieren, wenn die Konfigurationen unterschiedlich sind, ist wirklich ätzend. Die Zeit, die andere Entwickler und ich damit verbracht haben, Probleme mit der Nodejs-Version zu debuggen, verschiedene SDKs zu installieren und zu deinstallieren und Umgebungsvariablen und Secrets zu finden und einzurichten, ist einfach wahnsinnig.“

Der Aufwand für die Wartung der lokalen Entwicklungsumgebung ist tatsächlich enorm. Das bestätigt eine [Umfrage unter Gitlab Usern](#). 61% der befragten Entwickler_innen berichten, dass sie zumindest einmal im Monat Zeit zur Problembehandlung mit ihrer lokalen Entwicklungsumgebung aufbringen – 31% sogar zumindest einmal pro Woche. Die ständige Fehlersuche und die Beschäftigung mit der Entwicklungsumgebung, wenn beispielsweise ein Bug in einer alten Version behoben werden muss, führen zu erheblichen [Produktivitätseinbußen und Frust](#).

Wo bleiben die Werkzeuge, die es Software-Entwickler_innen erlauben, jederzeit produktiv zu arbeiten?

Cloud Development Environments

CDEs sind Remote-Umgebungen, in denen Entwickler_innen alle Tools zur Verfügung haben, die sie für ihre Arbeit brauchen. Eine CDE ist also ein Werkzeug, das Remote Development ermöglicht und einfacher macht. CDEs haben den Anspruch, lokale Entwicklungsumgebungen obsolet zu machen.

Remote bedeutet mal nur, dass die Entwickler_innen nicht lokal auf ihrem Laptop oder Rechner arbeiten, sondern sich zu einer anderen Umgebung verbinden und dort arbeiten. Diese kann sich im Rechenzentrum des Arbeitgebers befinden, oder in der Private Cloud des Arbeitgebers, oder ein SaaS Angebot darstellen. Verschiedene CDEs bieten hier verschiedene Optionen an.

Mehr dazu:

[Cloud Development Environments: Der ultimative Guide](#)

Vorteile

In diesem Abschnitt erhalten Sie einen schnellen Überblick, wie CDEs Unternehmen dabei helfen können, die Produktivität und Developer Experience auf das nächste Level zu heben.

Standardisierung

Ein Problem lokaler Entwicklungsumgebungen ist, dass jede anders ist.

Selbst mit strikten Geräte- und Software-Richtlinien können die möglichen Störfaktoren zwar eingeschränkt, aber nie ganz aus der Welt geschafft werden. Es treten trotzdem individuelle Probleme auf.

Besonders frustrierend ist das, wenn man bedenkt, wie aufwändig es ist, Standardisierungs-Richtlinien zu entwickeln und diese kontinuierlich durchzusetzen. Häufig wird damit viel Widerstand und Frust bei Entwickler_innen erzeugt, die auf lieb gewonnenen Tools verzichten und stattdessen mit Software arbeiten müssen, die sie nicht mögen.

Wie wäre es, wenn Entwickler_innen auf Knopfdruck vollständig standardisierte Entwicklungsumgebungen zur Verfügung hätten, die jedes mal neu erstellt werden und zu 100% exakt den Richtlinien entsprechen? Und daneben Entwickler_innen ganz freie Hand gegeben werden kann, was sie lokal für Betriebssystem, Browser oder sonstige Tools verwenden?

Das ist mit CDEs möglich. Die Umgebungen können außerdem so definiert werden, dass sie der Produktionsumgebung der zu entwickelnden Software so nahe wie möglich kommen (z. B. wird auch der Betrieb eines Kubernetes-Clusters auf einer CDE plötzlich machbar, was lokal zu sehr viel Kopfschmerz führen kann). Damit wird das „works on my machine“ Problem reduziert.

Und das bei gleichzeitig deutlich höherer Developer Experience - also angenehmer und erfreulicherer Arbeitsweise für Entwickler_innen.

Zeitersparnis

Joel Spolsky schreibt: "Erstklassige Entwicklungsteams quälen ihre Programmierer nicht. Selbst kleine Frustrationen, die durch die Verwendung unzureichender Werkzeuge verursacht werden, summieren sich und machen die Programmierer mürrisch und unglücklich. Und ein mürrischer Programmierer ist ein unproduktiver Programmierer."

Entwicklung von Unternehmenssoftware bedeutet, dass rechenintensive Arbeiten durchzuführen sind. Das führt oft zu Frustration bei Entwickler_innen. Beispielsweise, wenn der RAM aufgrund eines Tools fast in Flammen aufgeht und auf die Fertigstellung von Tests etc. gewartet werden muss. Und: Frustration beim Management aufgrund der Kosten für die Bereitstellung von entsprechendem Equipment.

Da CDEs auf Servern laufen und mit mehr Ressourcen (RAM, Speicherplatz, etc.) ausgestattet werden können als lokale Laptops, reduziert sich die Zeit für Builds, Tests und andere leistungsintensive Prozesse und damit die Wartezeit von Entwickler_innen. Angenehmer Nebeneffekt: Die Auswahl des Arbeitsgeräts kann dadurch günstiger ausfallen und die Kosten für Unternehmen senken. Entwickler_innen haben in den CDEs dennoch alle – und noch mehr – Ressourcen zur Verfügung, die sie für komfortable Entwicklungsarbeit brauchen.

Die Zeiten sind vorbei, in denen Entwickler_innen ewig auf das Fertigwerden des lokalen Builds warten (weil der Laptop daneben nichts schafft und der Build trotzdem lange braucht).

Sicherheit

In Zeiten der Remote-Arbeit wissen Vorgesetzte manchmal gar nicht so genau, wo sich ihre Entwickler_innen gerade befinden.

Zusammenarbeit mit Freelancern in der ganzen Welt wird immer normaler. Und damit wird das Thema der Sicherheit von Source Code und Daten immer dringender. Denn was passiert, wenn der Laptop im Taxi in Delhi liegen bleibt? Oder die Kolleg_innen in den USA im Co-Working-Space kein absperbares, eigenes Büro haben, sondern ihr Arbeitsgerät über Nacht einfach offen liegen lassen? Ist das Endgerät einmal verloren, kann zwar der Zugriff auf Remote-Sourcen gesperrt werden – was sich am Laptop befindet, ist jedoch für immer verloren.

Eine ganz eigene Kopfweh-Kategorie: Zugriff auf personenbezogene Daten, die zum Testen der Software notwendig sind. Wie ist das am Besten zu regeln, rechtlich und auch praktisch, wenn das eigene Team auf der ganzen Welt verstreut ist?

Maßnahmen wie Verschlüsselung von Festplatten, Richtlinien zur Nutzung von vertrauenswürdigen Netzwerken, Deaktivierung von USB Ports, Arbeiten über VPNs – viele mögliche Lösungsansätze existieren. Alle sind problematisch, denn jeder einzelne involviert signifikanten Aufwand, ist schwer zu durchzusetzen und löst das Problem dennoch nur zum Teil. Was tun?

Sie ahnen die Antwort: CDEs können vollständig Air-Gapped eingerichtet werden, sodass Source Code und Testdaten nie lokal am Rechner der Entwickler_innen landen. Ein Air-Gap wird eingesetzt, wo höchste Sicherheitsanforderungen gefragt sind. Es handelt sich um eine Umgebung, die isoliert von anderen Geräten und vom Internet ist. Informationen können innerhalb des Netzwerkes ausgetauscht werden, sie verlassen es aber niemals. Testdaten können in CDEs deshalb genutzt, aber so abgeschirmt werden, dass Entwickler_innen keinen direkten Zugriff haben.

Für Entwickler_innen ist die Arbeit in der CDE komfortabel, damit existieren auch keine Anreize für Missbrauch, der häufig im Dienste der Produktivität und mit besten Intentionen stattfindet. Und da Source Code und Daten gar nie lokal abgelegt werden, ist das Risiko bei Verlust oder Diebstahl des Endgerätes gleich null.

Produktivität

Alle Entwickler_innen kennen es: In der lokalen Entwicklungsumgebung Branch zu wechseln, z. B. um auf eine alte Version der Software zurücksteigen, um einen Bug zu fixen oder zu überprüfen, ist mühsam.

Alles muss zusammenpassen: Alle Abhängigkeiten müssen in der richtigen Version vorhanden sein, alle Komponenten der Software müssen miteinander funktionieren – und wer weiß nochmal genau, welche Versionen kompatibel waren?

Nach dem Bug Fix geht das ganze dann wieder in Revers: den eigenen aktuellsten Feature Branch wieder pullen, Abhängigkeiten updaten, und hoffen, dass der Kunde, der noch die alte Version nutzt, nicht so bald wieder einen Bug findet. Die in der neuen Version ohnehin alle bereits behoben sind.

Die Lösung? Klar, alle Kunden dazu zwingen, immer die letzte Version zu nutzen. Leider ist das in der Realität oft nicht möglich. Und auch ohne Kunden-Bugfixes in alten Versionen ist das Wechseln von Branches oft notwendig, es ist ein Teil normaler Entwicklungsarbeit. Die Lösung kann daher nicht sein, Branch-Switching obsolet zu machen. Eine schönere Lösung: CDEs.

CDEs stehen in allen benötigten Versionen fix und fertig zur Verfügung. CDEs können bei Bedarf nicht nur für Major-Versionen, sondern sogar für spezifische Commit-Hashes einzelner Branches erstellt werden. Jede_r Entwickler_in kann mehrere davon parallel betreiben. Damit wird das Arbeiten in unterschiedlichen Branches plötzlich zum Kinderspiel.

Das erhöht nicht nur die Produktivität und senkt den Frust, sondern führt auch zu deutlich besserer Unterstützung von Kunden. Denn wenn es einfach geht, hilft man doch gleich viel lieber.

Zusammenfassung

- Die lokale Entwicklungsumgebung einzurichten kostet Zeit. Die Wartung erfordert kontinuierlichen Aufwand. 61 % befragter Entwickler_innen berichten, dass sie zumindest einmal im Monat Probleme beheben müssen. 31 % sogar zumindest einmal pro Woche. Das führt zu Produktivitätseinbußen.
- Standardisierungsrichtlinien helfen, es treten trotzdem individuelle Probleme mit lokalen Entwicklungsumgebungen auf. Bei Entwickler_innen können Standardisierungsvorhaben zu Frust führen, weil liebgewonnene Tools nicht mehr verwendet werden können.
- Rechenintensive Tasks erfordern Rechner, welche die nötige Leistung erbringen. Das geeignete Equipment steht aber nicht immer zur Verfügung (Oft auch aus Kostengründen). Das führt zu Frust bei Entwickler_innen.
- Source Code liegt am Rechner der Entwickler_in. Geht das Endgerät verloren oder wird gestohlen, steht die Unternehmensexistenz auf dem Spiel.
- Branch-Switching kostet Zeit. Abhängigkeiten müssen in der richtigen Version vorhanden sein, alle Komponenten müssen zusammenpassen. Der ganze Prozess dauert.
- Eine Lösung dafür sind Cloud Development Environments (CDEs). CDEs sind Remote-Umgebungen, in denen Entwickler_innen alle Tools sofort zur Verfügung haben, die sie für ihre Arbeit brauchen. CDEs haben den Anspruch, lokale Entwicklungsumgebungen obsolet zu machen, standardisieren diese unternehmensweit, kommen der Produktionsumgebung nahe, sind auf Knopfdruck verfügbar und geben Entwickler_innen freie Hand bei der Wahl ihrer lokalen Tools. CDEs übernehmen leistungsintensive Prozesse - Entwickler_innen können auf jedem Gerät arbeiten, Wartezeiten auf Builds und Tests verkürzen sich. Kosten für teure Arbeitsgeräte werden reduziert. Die Entwicklungsumgebungen können Air-Gapped eingerichtet werden, sodass der Source Code und sensitive Testdaten nie lokal am Rechner landen. Mit CDEs stehen alle Versionen der Software sofort abrufbar zur Verfügung. Es muss einfach nur die dafür konfigurierte CDE aufgerufen werden.

**STELLEN SIE DIE EFFIZIENZ IHRER
AKTUELLEN ENTWICKLUNGSUMGEBUNG
AUF DIE PROBE UND STARTEN SIE MIT
CLOUD DEVELOPMENT ENVIRONMENTS
VON CLOUDOMATION**

Weiter informieren

Jetzt anfragen



Ein Produkt der Starflows OG
Darnautgasse 6/6, 1120 Wien, Österreich

+43 699 10 69 5915
www.cloudomation.com
info@cloudomation.com

wirtschafts
agentur
wien



Für die
Stadt Wien

Gefördert durch die Wirtschaftsagentur Wien.
Ein Fonds der Stadt Wien.