Universität Hamburg
Fachbereich Informatik

Microservices und die Zukunf von DevOps

am Arbeitsbereich Softwarearchitektur (SWA)

Bennet Brunsen, Mathias Schmialek

27. November 2015

Blubb

Inhaltsverzeichnis

1 Einleitung

Lorem Ipsum...

1.1 Developer

Viele Betriebe wie zum Beispiel Versicherungen sind heutzutage beinahe vollständig auf Software angewiesen, die ihre Arbeit korrekt abbilden und vereinfachen. Diese Software wird von Entwicklern (Developern) mit Hilfe von Abstraktion und Programmiersprachen hergestellt. Damit die Software auch den Ansprüchen der Anwender entspricht, nimmt der Entwickler die Anforderungen vom Anwender beziehungsweise dem Auftraggeber entgegen und in Programmcode umgesetzt. Zusätzlich ist es die Aufgabe des Entwicklers den geschriebenen Programmcode sowohl manuell als auch programmatisch zu testen und damit ein korrektes Verhalten der Anwendung sicherzustellen.
Das Ergebnis der Arbeit eines Entwicklers ist ein Stück Software, das dem Kunden als Installationsdatei oder Ähnlichem übergeben wird. Der Entwickler selbst ist nicht dafür zuständig, dass die Anwendung in den laufenden Betrieb des Kunden integriert wird.

1.2 Operation

Unter Operations werden im herkömmlichen Sinne Mitarbeiter mit IT-Kenntnissen verstanden, deren Ziel es ist, den täglichen IT-Betrieb am laufen zu halten. Zu ihren Aufgaben gehören unter anderem die Bereitstellung von Hardware für Mitarbeiter des Betriebes sowie die Verwaltung von Servern. Sie sind ebenfalls dafür zuständig, Software unter anderem auf die Geräte der Mitarbeiter und auf die Server des Betriebes zu spielen. Dabei ist kann die Software entweder käuflich erworben oder im eigenen Unternehmen entwickelt worden sein. Dies wird in der Regel mit Skripten realisiert, die Operations selber anlegen.
Zu einer Software, die kommerziell erworben wurde gibt es in der Regel so gennante Service Level Agreements, die eine mit dem Softwarehersteller vereinbarte Leistungsqualität der einzusetzenden Software beschreiben. Ein Beispiel hierfür wäre, dass ein Hersteller von einer Cloudbasierten Anwendung, seinem Kunden verspricht, dass die Anwendung mit einer bestimmten Anzahl an Anfragen innerhalb einer definierten Zeitspanne zurecht kommt, sodass keine Ausfälle entstehen. Innerhalb eines Betriebes überwachen Operations die Service Level Agreements melden gegebenenfalls Verstöße ihren Vorgesetzten oder dem Hersteller der Software.
Um die Einhaltung der Service Level Agreements zu überprüfen, wird die Software mit Hilfe von Monitoring und Logging Werkzeugen überwacht. Dabei geben die Werkzeuge des Monitorings den Operations, wichtige Informationen wie zum Beispiel den Speicherverbrauch und den Status der laufenden Anwendungen. Mit Hilfe der Logging Werkzeuge, können die Operations auf Fehler innerhalb der Anwendung schließen. Für den Fall, dass die Umgebung der Anwendung falsch konfiguriert ist, wird dies üblicherweise in Log-Dateien geschrieben, die den Operations zur Verfügung stehen.

Wie bereits erwähnt, ist das Ziel von Operations den IT-Betrieb am laufen zu halten. Dies wird in der Regel als Business Continuity bezeichnet. Allerdings ist dies nicht nur mit einer für die Anwendung passend konfigurierten Umgebung möglich. Die Rechner und Server eines Betriebes müssen nach Außen abgesichert werden, um Ausfälle durch Schadangriffe so gut wie möglich zu unterbinden. Dies ist ebenfalls ein Aufgabenbereich von Operations.

1.3 Agile Softwareentwicklung

Neben der klassischen Softwareentwicklung, die nach starren Modellen wie zum Beispiel dem Wasserfallmodell arbeitet, hat sich in den letzten Jahren die agile Softwareentwicklung etabliert. Die agile Softwareentwicklung mit agilen Vorgehensweisen wie zum Beispiel Scrum oder Xtreme Programming, bindet den Kunden stärker in den Entwicklungsprozess ein und bietet die Möglichkeit schon während der Entwicklung auf das die Software Einfluß zu nehemen.
Len Bass hat bei der agilen Softwareentwicklung drei Phasen identifiziert, auf die im folgenden eingegangen wird. Die erste Phase ist die sogenannte Inception Phase. In dieser Phase werden unter anderem die Anforderungen an die Software aufgenommen und erste Modellierungsarbeiten betrieben. Zustäzlich wird in dieser Phase ein Release Plan erstellt, der mit dem Kunden abgestimmt wird.
Die zweite Phase ist die so genannte Construction Phase. In dieser Phase wird die Software nach agilen Methoden wie zum Beispiel Scrum entwickelt. Das Ergebnis dieser Phase ist ein Software Artifakt, das dem Kunden übegeben werden kann.
Die finale Phase ist die Transition Phase. In dieser Phase wird das zuvor hergestellte Artifakt beim Kunden auf die Rechner beziehungsweise Server gespielt und somit in den laufenden Betrieb integriert.

1.4 Monolith

2 DevOps

2.1 Motivation

2.2 Practices

2.3 Ziele

3 Microservices

blubbb

3.1 Motivation

3.2 Architektur

3.3 Design Entscheidung

3.4 Herausforderungen

3.5 Antipattern

3.6 Gefahren

3.7 Unterschiede zum Monolith

3.8 Beispiel

4 Zukunft von DevOps

4.1 Vorraussichtliche Marktakzeptanz

4.2 Organisatorische Probleme

4.3 Prozess Probleme

4.4 Technologische Probleme

4.5 Bugfixing

5 Zusammenfassung und Fazit