Qualysoft

WHITEPAPER

ESSENTIELLE DEVOPS METRIKEN

www.qualysoft.com

Essentielle DevOps Metriken
um Performance zu messen

Qualysoft

DevOps wird mittlerweile in immer mehr Unternehmen praktiziert. Ein wichtiges Thema dabei ist die kontinuierliche Verbesserung. Dabei ist aber nicht nur die Verbesserung des Produkts gemeint, sondern auch des Teams und der internen Prozesse. Ohne geeignetes Monitoring und vor allem ohne aussagekräftigen Metriken kann aber nur geraten werden, wie es um die Performance steht. Auch Trendanalysen und Vorhersagen sind ohne solch einer Lösung nur schwer möglich. In diesem Whitepaper wird deshalb auf die Rolle des Monitorings eingegangen und die aus unserer Sicht sechs wichtigsten Metriken präsentiert.

DevOps Lifecycle und die Rolle des Monitorings

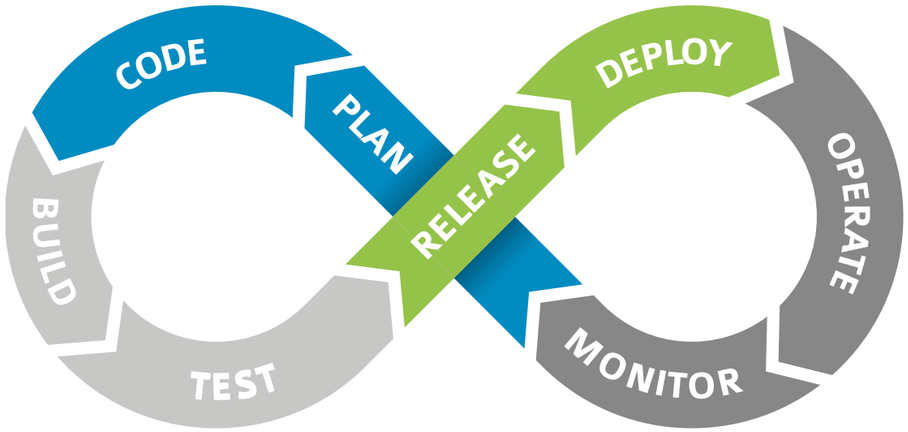

Abbildung 1: DevOps Lifecycle [1]

Anhand der Grafik vom DevOps Lifecycle ist schön zu sehen, dass nach der Entwicklung, dem Testen und dem Deployment in Produktion eine Monitoring Phase folgt. Diese hat den einerseits den Zweck die Applikation zu überwachen (Server-Überwachung) andererseits aber auch Kundenfeedback zu erhalten. Bug-Meldungen und Support Tickets sind dabei wohl die häufigsten Mittel.

Doch das Monitoring muss sich nicht nur auf die Kundensicht beschränken, auch die Teamperformance kann und soll gemessen werden. Denn nur so können Probleme in kürzester Zeit identifiziert und behoben werden, was ganz im Sinne von agiler Entwicklung und kontinuierlicher Verbesserung ist. Um das allerdings sinnvoll machen zu können, müssen Metriken definiert werden.

Im nächsten Kapitel präsentieren wir daher die aus unserer Sicht sechs wichtigsten Metriken, um die Teamperformance messen zu können.

Metriken

Deployment Frequency

Diese Metrik wird in der Literatur auch oft als Throughput bezeichnet. Sie besagt, welcher Prozentsatz der Builds es in Produktion oder eine andere wichtige Umgebung, wie z.B. Staging, schaffen.

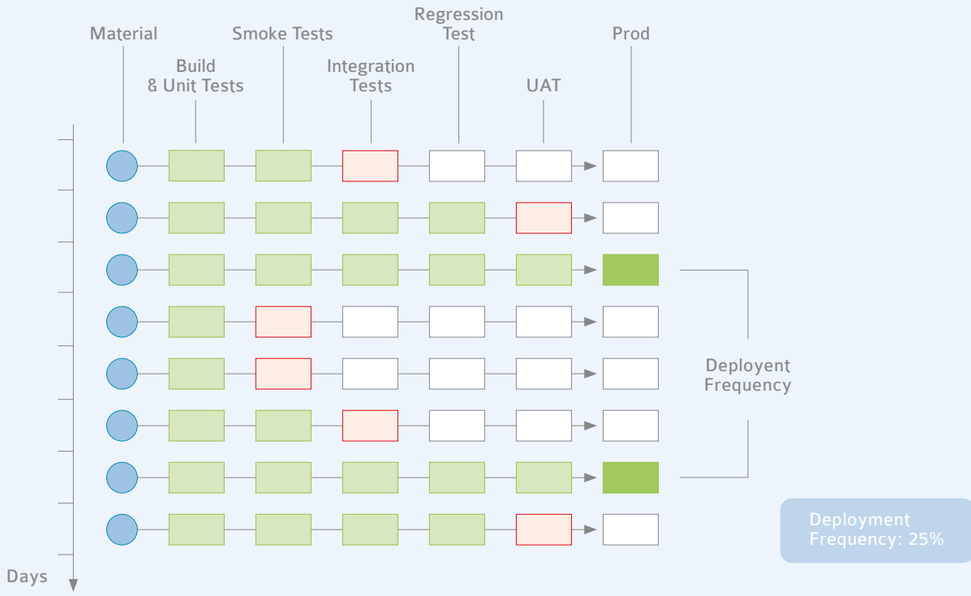

Abbildung 2: Deployent Frequency [2]

Für die Berechnung werden lediglich die Anzahl der Deployments in Production und die Anzahl der der Deployments, die es nicht in Production schaffen gezählt. Im obigen Beispiel sind es acht Möglichkeiten in Production zu gehen, aber nur zwei dieser Deployments werden durchgeführt, daher ist die Deployment Frequency 25%.

Das Wort “Continuous” in Continuous Delivery impliziert eine hohe Deployment Frequency. Eine hohe Deployment Frequency bedeutet öfter Deployments zu machen und schafft damit mehr Möglichkeiten, um Feedback zu erhalten. Außerdem wird das Produkt durch eine höhere Deployment Frequency auch öfters an Kunden und Stakeholder ausgeliefert.

Im Allgemeinen sollte versucht werden die Deployment Frequency so weit wie möglich zu erhöhen. Wichtig dabei ist allerdings, sie in Balance mit der Qualität zu halten. Es sollten keine Tests entfernt werden, um die Frequenz zu erhöhen, denn dadurch könnte die Qualität leiden. Ziel von Continuous Delivery sind häufige Deployments in Production, während die Qualität steigt.

Lead Time

Während die Deployment Frequency die Anzahl der Deployments überblickt, gibt die Lead Time die Dauer für Änderungen vom Beginn der Entwicklung bis zur Fertigstellung an. Diese Metrik gibt also eine Zeit in Stunden oder Tagen an und beantwortet die Frage „Wie lange dauert es, bis ein Feature in Production kommt?“.

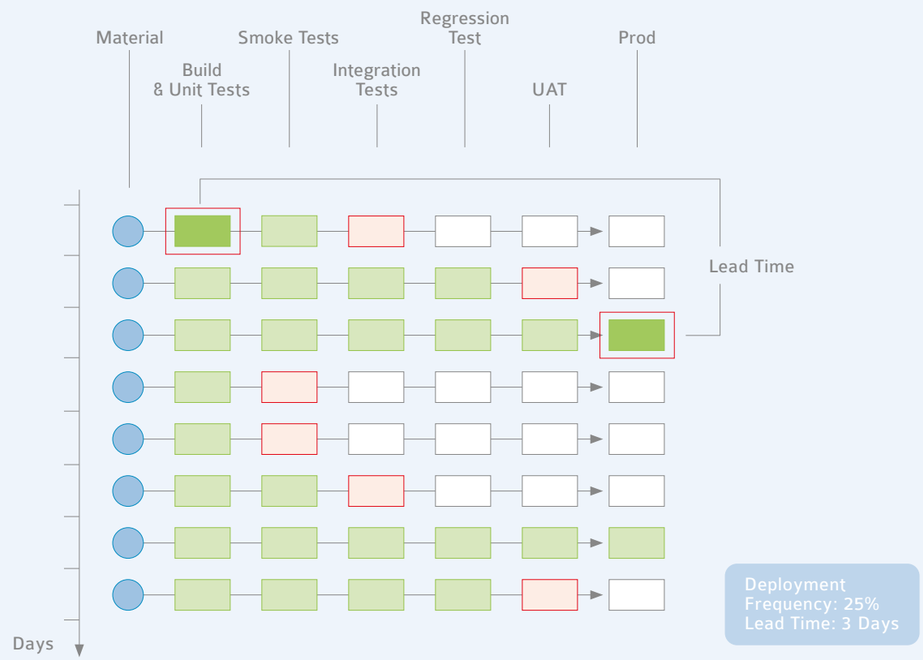

Abbildung 3: Lead Time [3]

Im oben gezeigten Diagramm wird für den Beginn der Lead Time der Zeitpunkt des ersten Commits (Beginn der Continuous Delivery Pipeline) herangezogen. Das Ende der Lead Time wird durch das Deployment in Production (Ende der Continuous Delivery Pipeline) definiert. Angenommen die Pipelineausführung geschieht einmal täglich, ergibt das eine Lead Time für das Feature von drei Tagen.

Die Lead Time ist eine wichtige Metrik, die helfen kann Fragen wie „Wann wird das Feature fertig sein?“, „Wenn wir heute anfangen, wann können wir es in Production bringen?“, „Können wir das
bis nächste Woche ausliefern?“ zu beantworten.

Change Failure Rate

Diese Metrik gibt an, welcher Prozentsatz der Änderungen ein fehlgeschlagenes Deployment in Production verursachen. Für die Berechnung werden - ähnlich der Deployment Frequency – die Anzahl aller Deployments in Production und die Anzahl der fehlgeschlagenen Deployments in Relation zueinander gesetzt.

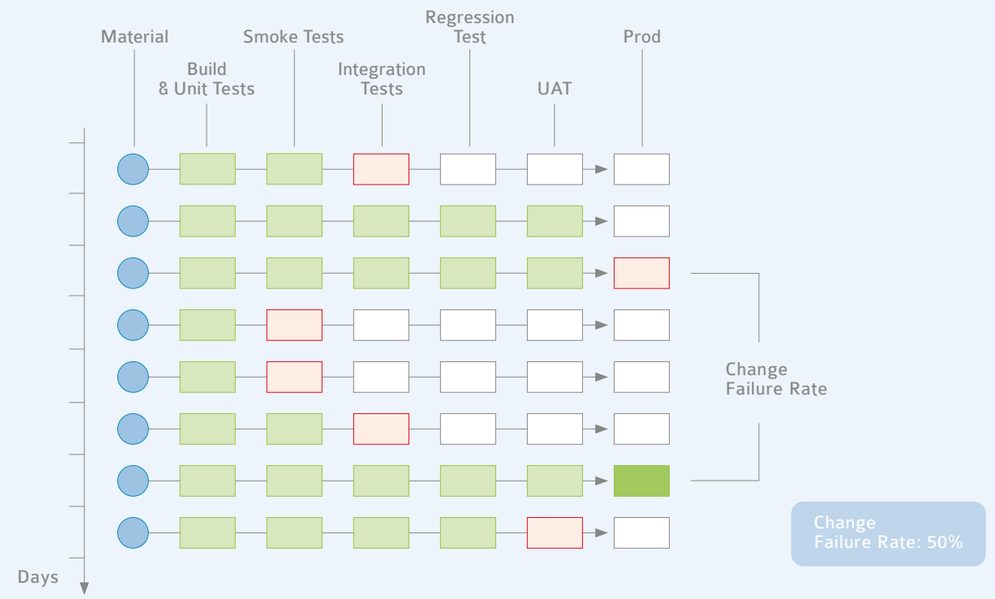

Abbildung 4: Change Failure Rate (angepasst von [2])

Ziel sollte es sein den Wert dieser Metrik so niedrig wie möglich zu halten. Laut dem 2018 DORA Bericht haben Elite DevOps Performer eine Change Failure Rate zwischen 0% und 15%.
Diese niedrigen Werte sind nur erreichbar, wenn der Deployment Prozess automatisch, konsistent und zuverlässig funktioniert.

Mean Time to Repair

Die Mean Time to Repair oder auch Mean Time to Restore (MTTR) gibt an, wie lange es dauert, um ein fehlgeschlagenes Deployment in Production zu beheben. Abhängig von der Komplexität der Applikation und dem Problem, kann die MTTR Minuten, Stunden oder sogar Tage betragen. Die Berechnung der Metrik erfolgt, indem die summierte Downtime, die durch Fehler entstanden ist, durch die Anzahl der Fehler geteilt wird.

Beispiel:
Wenn ein System drei Mal im Monat fehlschlägt und eine Downtime von insgesamt sechs Stunden beträgt, dann beträgt die MTTR im Durchschnitt zwei Stunden.

$$\text{MTTR} = \frac{6 \text{ Stunden}}{3 \text{ Failures}} = 2 \text{ Stunden}$$

Change Volume

Diese Metrik gibt an wie viele User Stories bzw. Features und New Lines of Code pro Deployment ausgeliefert werden. Sie gibt also an, wie wertvoll die Deployments im Durchschnitt sind. Um häufige Deployments durchführen zu können, muss die Change Volume eher klein bleiben. Wie klein ist abhängig vom Produkt und kann pauschal nicht gesagt werden.

Ziel sollte es allerdings sein, keine hohen Werte in dieser Metrik zu erzielen, da mit hohen Werten eindeutig keine häufigen Deployments durchgeführt werden. Die Deployment Frequency in Kombination mit dem Change Volume zeigt sehr schnell, ob wirklich nach Continuous Delivery/Continuous Deployment gearbeitet wird.

Defect Escape Rate

Diese Metrik gibt an, wie viele Bugs in Produktion gefunden wurden. Durch diese Metrik kann also der Wirkungsgrad der Tests und der QA gemessen werden.

Um die Berechnung der Metrik zu erleichtern, sollten gefundene Bugs mit der Info, wo sie gefunden wurden, hinterlegt werden.

Ziel dieser Metrik ist es so niedrig wie möglich zu sein. Bugs in Produktion zu finden ist kein Weltuntergang, zeigt aber, dass es in der Qualitätssicherung Verbesserungspotential gibt.

FAZIT

In der DevOps Welt geht es nicht nur um die Applikationsperformance (Ops) sondern eben auch um die Performance der Entwickler und Tester (Dev). Um den Überblick über die Performance beider Welten zu haben, sollten aussagekräftige Metriken verwendet werden. Die von uns beschriebenen Teamperformance Metriken sind allerdings nicht die einzigen Metriken die im DevOps Bereich gemessen werden können, sind aber ein solider Startpunkt.

[1] A. Yildirim, „DevOps Lifecycle: Continuous Integration and Development," Medium, 7 August 2019. [Online].
Available: https://medium.com/@yildirimabdrhm/devops-lifecyclecontinuous-integration-and-development-e7851a9c059d. [Accessed 12 March 2021].

[2] S. Aravind and S. Prince, „Continuous Delivery Metrics Part 2: How often do you deploy to production?," GoCD, 30 November 2018. [Online].
Available: https://www.gocd.org/2018/11/30/deployment-frequency/. [Accessed 12 March 2021].

[3] S. Aravind and S. Price, „Continuous Delivery Metrics Part 3: How long does it take to get from committing code to production?," GoCD, 14 January 2019. [Online].
Available: https://www.gocd.org/2019/01/14/cd-metrics-deployment-lead-time/. [Accessed 12 March 2021].