APPDYNAMICS

DevOps-Leitfaden:

Wie APM-Lösungen zur effektiven Durchführung von Performance-Tests beitragen

KAPITEL 1:

Einleitung

In diesem kurzen eBook erläutern wir, wie der Einsatz einer Lösung für das Application Performance Management (APM) in Verbindung mit automatisierten Performance-Tests die Zusammenarbeit zwischen Entwicklung und IT-Betrieb im Unternehmen effektiv unterstützt.

Bei Unternehmen, die aufgrund der gestiegenen Anforderungen in der digitalen Welt ihre Anwendungen schneller und in höherer Qualität veröffentlichen wollen, gehören DevOps-Strategien fast schon zum Standard. Kürzere Entwicklungszyklen und höherwertige Releases sind ein entscheidendes Kriterium für den Erhalt der Wettbewerbsfähigkeit. Denn sie sind die Voraussetzung dafür, dass Applikationen kontinuierlich an die sich ändernden geschäftlichen Anforderungen angepasst werden können.

In Kombination mit Performance-Tests ermöglicht APM es, dass wichtige Applikationen auch unter Spitzenlastbedingungen auf Dauer ein erstklassiges Kundenerlebnis bieten.

Wenden wir uns aber zunächst dem Performance-Test zu: Worum geht es und warum wird er immer wichtiger?

Performance-Test

ÜBERBLICK

Ein Performance-Test ist darauf ausgerichtet, zu simulieren und zu messen, wie stabil und reaktionsfähig eine Softwareanwendung bei einer bestimmten Belastung ist. Er kann auch dazu dienen, weitere qualitätsbezogene Eigenschaften des Systems wie Skalierbarkeit, Zuverlässigkeit und Ressourcennutzung zu untersuchen, zu messen, zu validieren oder zu verifizieren.

Die folgenden Performance-Tests dienen jeweils einem eigenen Zweck:

Lasttest

- Simulation der Spitzenlast unter den gegenwärtigen Betriebsbedingungen

Stressstest

- Ermittlung, wie das System auf extreme Belastungen reagiert – zum Beispiel bei 200 % der Spitzenlast unter den gegenwärtigen Betriebsbedingungen

Kapazitätstest

- Ermittlung, ob die Auslegung und Anzahl der Server geeignet sind

Geräte-, Front-End-, Back-End- und End-to-End-Optimierung mit Fokus auf:

- Höchstzahl der Aufrufe
- Latenzreduzierung
- Reduzierung von Bandbreite und Ressourcenumfang (css, JS, Bilder)

Beständigkeitstest

- Prüfung der Ausfallsicherheit über einen längeren Zeitraum

Performance-Tests werden automatisiert durchgeführt. Eine manuelle Ausführung wäre unzweckmäßig. Die Automatisierung bietet in erster Linie die Möglichkeit, die Aktivitäten der Endnutzer aufzuzeichnen und zu scripten, um sie dann in einer Zielumgebung maßstabgetreu zu wiederholen, wobei die Applikations- und Systemleistung gemessen wird.

Heute sind für jedes Budget und jede Anforderung Lösungen für die Performance-Testautomatisierung am Markt erhältlich (lizensiert oder Open Source). Viele lassen sich direkt in eine APM-Lösung wie die von AppDynamics einbinden.

Das Testen der Performance ist eine eigenständige Disziplin, die sich von anderen Arten von Software-Tests unterscheidet. Performance-Tests werden üblicherweise als „nicht funktionelle" Tests bezeichnet, weil sie Eigenschaften von Applikationen in den Blick nehmen, die nicht mit der reinen Funktionalität im Zusammenhang stehen (d. h. Skalierbarkeit und Verfügbarkeit unter Last).

Ein effektives Performance-Testing setzt die Festlegung der wesentlichen nicht funktionellen Anforderungen und eine genaue Aufgabenverteilung voraus:

Geschäftsrelevante Anforderungen

- Bestimmung des Umfangs (strategisch, taktisch)
- Bestätigung der erwarteten Leistungen (zur Vermeidung von Überraschungen)
- Bestätigung der zu testenden Client-Plattformen (Desktop, Mobil, Service-Nutzer, Batch)
- Identifizierung von grundlegenden Anwendungsfällen, Testdatenanforderungen und Testszenarien für jede Client-Plattform
- Identifizierung der zu erfassenden Key Performance Indikatoren (KPIs) für eine genaue Messung der Applikationsleistung
- Erstellung eines genauen Lastmodells (unerlässlich)
- Festlegung, ob Performance-Test im Haus oder von einem Dritten durchgeführt wird

Technische Anforderungen

- Verwaltung und Bereitstellung geeigneter Umgebungen für Performance-Tests
- Auswahl der Tools für automatisierte Performance-Tests, die die technischen und funktionellen Anforderungen sowie die Budgetvorgaben erfüllen (idealerweise mittels PoC)
- Monitoring-Konzept (idealerweise auf APM basierend)
- Daten erheben, auswerten, korrelieren und berichten

Auf der Grundlage dieser nicht funktionellen Anforderungen, die festzulegen sind, lassen sich realitätsgetreue Testszenarien für das Anwendungsverhalten erstellen wie unter anderem:

- Skriptgesteuerte Anwendungsfälle, die realistische Endnutzeraktivitäten über Browser oder Mobilgeräte darstellen. (Alternativ können sie API-Aufrufe von Service-Nutzern darstellen.)
- Ein genaues Lastmodell, das Folgendes festlegt:
- Die Art der erforderlichen Performance-Tests
- Die Verteilung der skriptgesteuerten Anwendungsfälle für Performance-Tests und der Umfang der zu generierenden Last
- Eine Testumgebung, die für den erforderlichen Umfang geeignet ist. Für End-to-End-Tests sollte die Produktivumgebung vollständig oder nahezu vollständig abgebildet werden. Wenn in der Entwicklungsphase einzelne Teilbereiche einer Applikation daraufhin getestet werden sollen, ob die vereinbarten Service Levels eingehalten werden, kann es ausreichen, die Testumgebung auf diese Anforderungen abzustimmen.

MONITORING UND ANALYSE

Die Simulation der Endnutzer-Aktivitäten ergibt selbstverständlich noch kein vollständiges Bild. Denn es ist genauso wichtig, die Applikationsleistung unter Last zu erfassen und auszuwerten. Konventionelle Performance-Management-Lösungen messen üblicherweise folgende Größen:

- Reaktionszeit in vollständigen oder festgelegten Teilbereichen von skriptgesteuerten Anwendungsfällen. Wenn ein Nutzer sich anmeldet, eine Suche durchführt und dann einen Artikel im Warenkorb ablegt, lässt sich normalerweise messen, wie lange die einzelnen Aktivitäten dauern.

- Die Anzahl der gleichzeitigen Nutzer zu einem beliebigem Zeitpunkt während der Testdurchführung. Diese Messgröße wird üblicherweise mit der Reaktionszeit korreliert, um zu veranschaulichen, wie sich die steigende Last auf die Reaktionszeit auswirkt.

- Die Anzahl und Art von Fehlern, die während eines Testlaufs auftreten. Wenn Fehler bei einer bestimmten Last auftreten, ist das ein wichtiger Hinweis auf Probleme mit der Skalierbarkeit der Anwendung.

Diese Messgrößen eignen sich zwar bestens für das Aufspüren von auftretenden Problemen, geben aber nicht notwendigerweise Aufschluss über die Problemursachen. Wenn man den Problemursachen auf den Grund gehen will, muss man die Leistungsdaten aus der gesamten Infrastruktur einschließlich Netzwerk, Host-Servern und Datenbanken erfassen.

AKTUELLE TRENDS IM PERFORMANCE TESTING

Performance-Management-Lösungen lassen sich seit langem in Lösungen für punktuelles Monitoring wie Windows Performance Monitor oder SNMP einbinden. Dadurch erhält man zwar zusätzliche Einblicke in die Applikationsleistung, nicht aber in die gesamte Infrastruktur. Es werden nur die ausgewählten Messdaten der überwachten Server oder Softwarekomponenten geliefert.

Aufgrund der gestiegenen Anforderungen in der digitalen Welt kommen zusehends Lösungen für die Performance-Testautomatisierung auf den Markt, die in erster Linie auf folgende Anforderungen ausgerichtet sind:

- Kürzere Entwicklungszyklen als Katalysator für den Wandel
- Kleinere, häufigere Releases
- Schnelleres Time-to-Market für neue Funktionen und Fehlerbehebungen
- Abkehr von schwerfälligen „monolithischen Architekturen"
- Kundenanforderungen: schneller, besser, weniger Geduld

Die Anbieter von Tools bieten heute üblicherweise Folgendes:

- Fokussierter Support für moderne Webentwicklungstechnologien wie Node.js, AngularJS, SPF und WebSockets
- Integration in etablierte Funktionstestlösungen wie Selenium
- Integration in Testautomatisierungsserver wie Jenkins als Bestandteil von CI und CD
- APM-Integration

Dieser letzte Punkt ist entscheidend, da die Einbindung in APM-Lösungen wie AppDynamics beispiellose Einblicke in die Ursachen von lastbedingten Problemen mit der Skalierbarkeit oder Reaktionszeit von Anwendungen ermöglicht und – was noch wichtiger ist – Klarheit darüber schafft, wie sie sich auf das Nutzererlebnis auswirken.

KAPITEL 2:

Performance-Probleme

In jeder Phase des Systems Development Life Cycle (SDLC) können sich unerkannte Schwachstellen einschleichen, die die Leistungsfähigkeit und Verfügbarkeit einer Applikation so sehr beeinträchtigen, dass die Erwartungen, die das Unternehmen und seine Kunden an sie stellen, nicht erfüllt werden.

Mit der viel zitierten „Linksverschiebung" ändert sich der konventionelle Ansatz des Software-Testens. Vor dem Hintergrund der immer engeren Zusammenarbeit von Entwicklung und IT-Betrieb wird Software häufiger und bereits in einer möglichst frühen Phase auf Fehler und Schwachstellen hin getestet, d. h. nicht erst in der Qualitätssicherung, sondern auch schon in der Entwicklung. Dieser Ansatz bietet gegenüber der konventionellen Qualitätssicherung einige wesentliche Vorteile:

- Je früher Fehler im SDLC erkannt werden, desto leichter und kostengünstiger lassen sie sich beheben.
- Durch das frühere Erkennen und Beheben von Schwachstellen reduziert sich der Aufwand für weitere Tests, und Applikationen können schneller und in besserer Qualität bereitgestellt werden.

Daraus ergibt sich die nächste Frage: „Wie verhindert man von vornherein, dass Schwachstellen sich im SDLC ausbreiten?" Betrachten wir die üblichen Gründe für das Auftreten von Schwachstellen in den einzelnen Phasen des Lebenszyklus:

·Konzeption
- Entwicklung
- Test
- Bereitstellung
- Skalieren

KONZEPTION

Hier liegt die Verantwortung eindeutig bei allen, die von Business- und IT-Seite sowie aus anderen Bereichen an der Planung und Konzeption der Applikation beteiligt sind.

Wenn die Applikation nicht von Grund auf in der richtigen Größenordnung geplant wird, ist die Wahrscheinlichkeit groß, dass sie den geschäftlichen Anforderungen oder denen der Endnutzer nicht gerecht wird. Es besteht die Gefahr, dass diese Probleme erst spät im SDLC erkannt werden, wenn sie normalerweise nur noch mit zeit- und kostenaufwändigen Maßnahmen behoben werden können.

ENTWICKLUNG

Klassischerweise werden die zu erbringenden Arbeiten und Aufgaben auf Programmiererteams aufgeteilt. Performance-Probleme können sich aus den folgenden Umständen ergeben:

- Fehlende oder missachtete Programmierstandards
- Den Entwicklern fehlt der Überblick über die gesamte Applikation
- SLAs werden bei der Programmierung nicht berücksichtigt
- Entwickler führen keine Performance-Tests durch
- Entwickler tauschen sich nicht mit dem Operations Team aus

In der Tat ändern sich viele dieser Dinge inzwischen, was im Wesentlichen auf die zunehmende Verbreitung eines flexiblen Entwicklungsstils zurückzuführen ist. Doch in vielen Unternehmen hat sich die flexible Herangehensweise noch nicht vollständig durchgesetzt.

Die Frage der Performance wird selbst in agilen Sprint-Projekten oftmals zu einem späten Zeitpunkt in Betracht gezogen und wegen des Zeitdrucks häufig zurückgestellt.

TEST

Die Performance-Testautomatisierung hat sich von einfachen Wiederholungsszenarien zu anspruchsvollen Test- und Monitoring-Strukturen entwickelt. Am Markt ist eine große Auswahl an Lösungen erhältlich, unter anderem auch etablierte Open-Source-Angebote wie JMeter.

Die gute Nachricht ist, dass die meisten Unternehmen – unabhängig von der Unternehmensgröße – inzwischen in der Qualitätssicherung ein Tool für automatisierte Performance-Tests einsetzen.

Die schlechte Nachricht ist hingegen, dass Performance-Tests oftmals eine rein taktische Funktion ohne formellen Prozess und ohne Management haben, obwohl sie doch ein wesentlicher Bestandteil der strategischen IT-Planung sein sollten. Dies kann dazu führen, dass nicht funktionelle Anforderungen unvollständig erfasst werden und Release-Entscheidungen auf der Grundlage unvollständiger oder irreführender Testergebnisse getroffen werden.

BEREITSTELLUNG

Die Entscheidung darüber, ob ein Release Candidate veröffentlicht werden kann, ist mit erheblichen Risiken verbunden.

Wenn diejenigen, die für die Bereitstellung verantwortlich sind, keinen hinreichenden Einblick in die Performance-Daten der betreffenden Applikation haben, dann fehlt ihnen eine wichtige Entscheidungsgrundlage.

Es müssen also klare Informationen darüber vorliegen, ob die Applikation die Anforderungen hinsichtlich Antwortzeit, Skalierbarkeit und Verfügbarkeit bei der erwarteten Spitzenlast erfüllt.

SKALIEREN

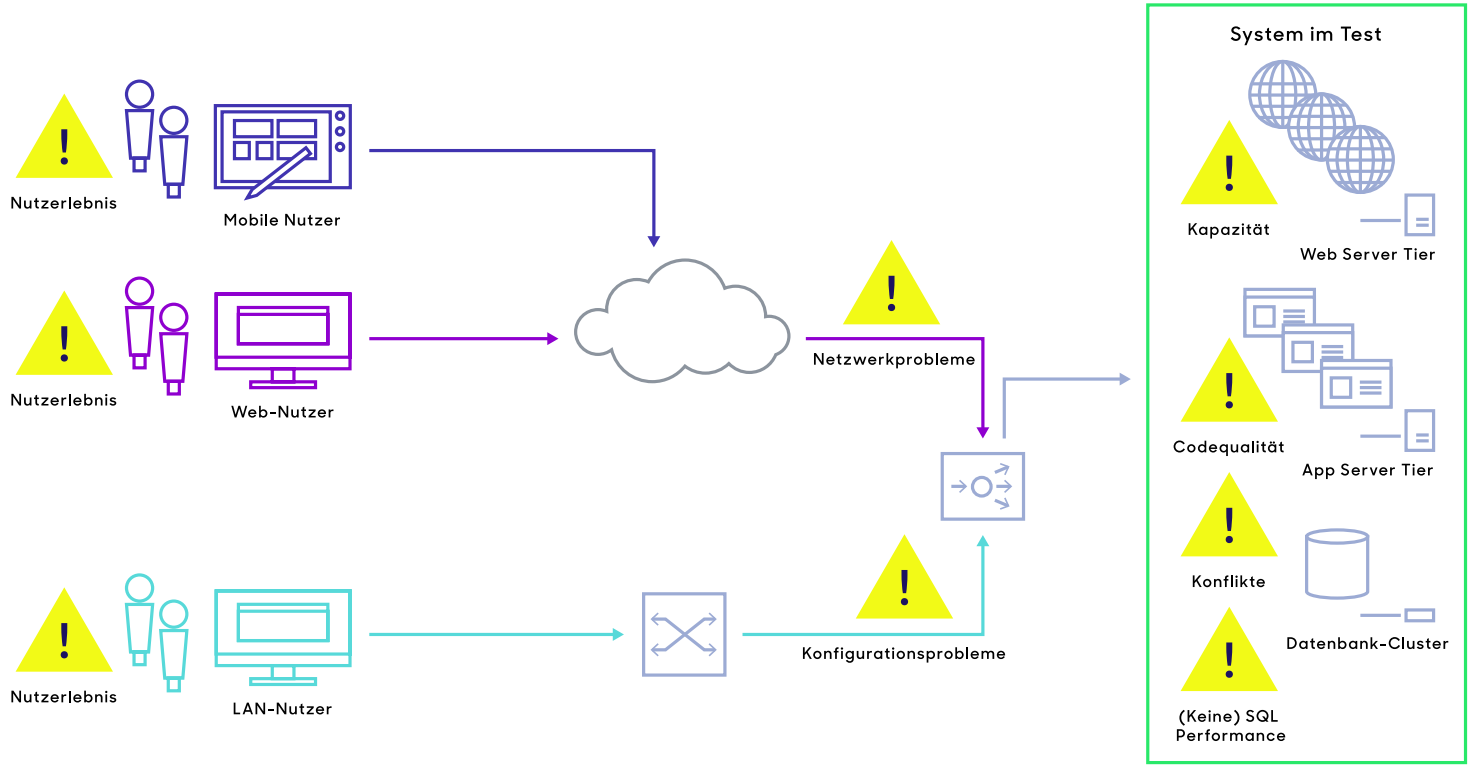
Eine nicht hinreichend skalierbare Applikation ist bei unerwarteten Spitzenlasten teilweise oder vollständig ausfallgefährdet. Selbst wenn sie weiterhin verfügbar bleibt, kann sich das Nutzererlebnis drastisch verschlechtern.

Der Einsatz von Cloud-Technik kann oftmals dazu beitragen, dass Applikationen mit unzureichender Skalierbarkeit auch bei zunehmender Last für den Endnutzer verfügbar bleiben.

Mit dieser provisorischen Lösung verschafft man sich zwar einen gewissen Spielraum, aber mit der steigenden Zahl der Cloud-Instanzen, die zur Deckung des Bedarfs eingesetzt werden müssen, steigen unweigerlich auch die Kosten.

Da eine mangelnde Skalierbarkeit in den meisten Fällen auf die Konzeption zurückzuführen ist, sind nachträgliche Korrekturen zumeist nur mit großem Zeit- und Kostenaufwand möglich.

Abb. 1 – Herausforderungen bezüglich der Applikationsleistung



KAPITEL 3:

Der Prozess

Nachdem wir uns nun mit den Performance-Risiken und ihrem Auftreten im SDLC befasst haben, soll es im Folgenden darum gehen, wie diese Risiken mittels bewährter Performance-Testmethoden und Application Performance Management (APM) vermieden werden können.

Application Performance Management (APM)

KURZER ÜBERBLICK

Von einfachen Code Profilern, die in der Entwicklung üblicherweise zum Einsatz kommen, hat sich das APM zu einer umfassenden Analyselösung für Software-Applikationen entwickelt, die vom Last-Mile- bis zum First-Mile-System den gesamten SDLC abdeckt und den Fokus zunehmend auf das Endnutzererlebnis richtet.

Der transaktionsbasierte Blick auf die Aktivitäten der Nutzer (vom Browser-Klick bis zum Datenbankabruf), den das APM ermöglicht, ist die effektivste Möglichkeit, Probleme mit der Applikationsleistung im Rahmen von Performance-Tests oder der Überwachung der Produktivumgebung zu beheben.

APM entwickelt sich zusehends zur bevorzugten Lösung für ein zentralisiertes Performance- und Verfügbarkeitsmanagement sowie für das Monitoring von wichtigen Anwendungen und von Endnutzern.

Da das APM den gesamten SDLC abdeckt, lässt es sich hervorragend mit automatisierten Performance-Tests kombinieren.

Betrachten wir erneut die in Kapitel 2 beschriebenen Phasen des Lebenszyklus unter dem Aspekt, wie mittels APM und Performance-Tests Risiken vermieden werden und das DevOps-Modell effektiv unterstützt wird.

RISIKEN MITTELS PERFORMANCE-TESTS VERMEIDEN

Konzeption

In der Phase der Konzeption gibt es im Allgemeinen nichts Greifbares, mit dem sich arbeiten ließe. Dennoch können Performance-Tests und APM für die Entscheidungsprozesse genutzt werden, indem für wesentliche Teile der Applikation Proof-of-Concepts (POCs) durchgeführt werden.

Ein Beispiel wäre der Vergleich der Performance von NoSQL-Datenbanken. APM-Lösungen wie AppDynamics unterstützen ein breites Spektrum von NoSQL-Datenbanken und ermöglichen ein umfassendes Monitoring sowie tiefgreifende Analysen.

Im Zusammenhang mit einem DevOps-Modell kann auch eine Strategie für das Monitoring von SLA-basierten Leistungskennzahlen über den gesamten SDLC entwickelt werden:

- Leistungskennzahlen für die Applikation
- Leistungskennzahlen für den Service
- Leistungskennzahlen für die Infrastruktur

Diese können dann im Verlauf des SDLC nach Bedarf weiterentwickelt und angepasst werden.

Entwicklung

Während der Entwicklung kann eine APM-Lösung wie AppDynamics dem DevOps-Team schlichtweg durch die frühe Bereitstellung von Performance-Testfunktionen auf Software-Komponentenebene sowie durch die Einbindung des APM in die Entwicklungsumgebung wertvolle Dienste leisten.

Auch wenn End-to-End-Tests in der Entwicklung normalerweise nicht möglich sind, können trotzdem sinnvolle Performance-Tests durchgeführt werden. Hier richtet sich der Fokus auf die Codequalität, auf Konflikte und auf die API Performance. Diese drei Punkte lassen sich problemlos mittels Performance-Tests und APM überwachen.

Performance-relevante Schwachstellen lassen sich frühzeitig erkennen und beseitigen, bevor sie im weiteren Verlauf ihre Wirkung zeigen.

Darüber hinaus lassen sich Performance-Tests in nächtliche Batch Jobs einbinden, während das APM für den Performance-Vergleich von verschiedenen Code Releases sowie für das Auslösen von Pass/Fail Alerts im Rahmen der Testautomatisierung genutzt werden kann.

Test

In der Qualitätssicherung hat man zumeist erstmals die Gelegenheit zur Durchführung von End-to-End-Tests. Der vollständige Einblick, den das APM in die Applikationsleistung und vor allem in das Endnutzererlebnis gewährt, ist in dieser Phase von großer Bedeutung.

Wenn Performance-Tests in der Entwicklung zur Routine werden, muss die Qualitätssicherung deutlich weniger Tests durchführen beziehungsweise wiederholen. Der Schwerpunkt liegt dann vielmehr darauf, zu bestätigen, dass bei vollständiger Bereitstellung der Applikation keine Performance-Probleme zu befürchten sind.

Bereitstellung

Wenn zu entscheiden ist, ob ein Release Candidate für die Bereitstellung geeignet ist, fehlt es, wie bereits erwähnt, üblicherweise an relevanten Performance-Daten.

Werden Performance-Tests in den formellen Prozess integriert, können in jeder Phase des SDLC die entsprechenden Performance-Daten ermittelt werden. Ein formeller Bericht als Nachweis dafür, dass die geforderten Leistungskennzahlen erreicht werden, sollte stets als Entscheidungsgrundlage dienen.

Eine APM-Lösung bietet zusätzliche Einblicke in die Performance der Applikation und damit ein höheres Maß an Sicherheit vor der Veröffentlichung. Darüber hinaus lassen sich die seit der Konzeption weiterentwickelten Leistungskennzahlen, die in der Entwicklung und Qualitätssicherung weiter angepasst wurden, als Grundlage für das Monitoring der Produktivumgebung nutzen, das idealerweise mittels APM erfolgt.

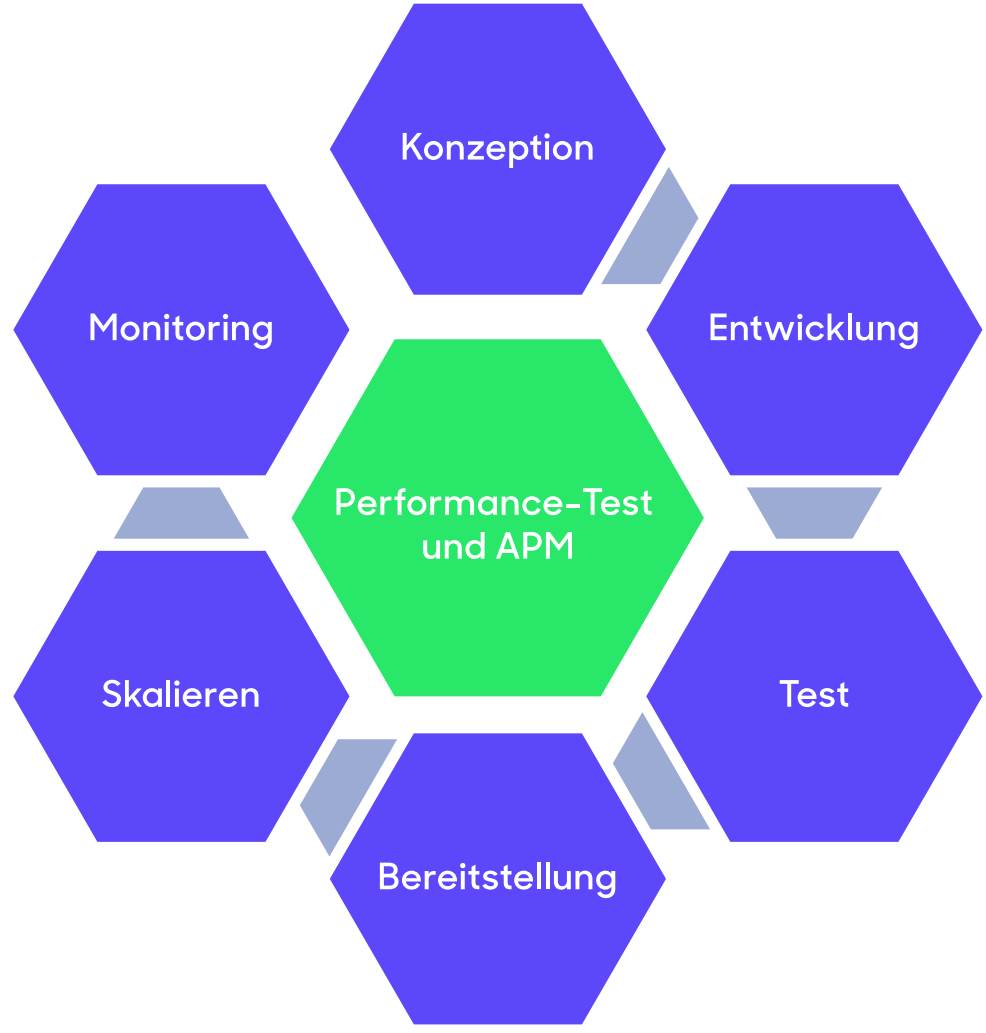
Skalieren

Nicht zuletzt ist es von entscheidender Bedeutung, dass die Applikation (geplanten und ungeplanten) Lastspitzen gewachsen ist. Performance-Tests allein können zwar die Kapazitätsgrenzen einer Applikation aufzeigen, geben aber keine Auskunft darüber, warum es an der Skalierbarkeit hapert.

Durch die Kombination von Performance-Tests und APM kann die Skalierbarkeit einer Applikation direkt einer Hostdatenbank und dem Verhalten auf Code-Ebene zugeordnet werden. Auf diese Weise lässt sich klären, warum die erforderliche Last nicht unterstützt wird.

Dank dieser tiefgreifenden Einblicke können auch die Grenzen der Skalierbarkeit einer Applikation ermittelt werden, um eine zuverlässige Planungsgrundlage für einen künftigen Ausbau der Infrastrukturkapazität zu haben.

Abb. 2 – Performance-Test und APM im Verlauf des SDLC



KAPITEL 4:

Den Erfolg messen

Abschließend soll es darum gehen, wie der Erfolg gemessen werden kann. Bisher haben wir uns damit befasst, welche Performance-Probleme es gibt, wie die Risiken bei Anwendung eines effektiven DevOps-Modells und der Kombination aus Performance-Test und APM auf ein Minimum begrenzt werden können. Wie aber lässt sich die Wirksamkeit dieser Maßnahmen messen?

Checkliste für die Erfolgsmessung

Für die Messung des Erfolgs bedarf es zunächst einer Bestätigung, dass bei der Anwendung von Performance-Test und APM die Best Practices befolgt werden. Wenden wir uns also noch einmal den Phasen des Lebenszyklus zu, um die Checkliste der wesentlichen Maßnahmen durchzugehen.

KONZEPTION (PERFORMANCE-ORIENTIERT)

- ☐ Überlegungen zur erforderlichen Performance als eine grundlegende Anforderung der Anwendungskonzeption betrachtet
- ☐ Performance-bezogene Leistungskennzahlen für jede neue Applikation oder Funktionsgruppe festgelegt:
  - ☐ Antwortzeit
  - ☐ Gleichzeitigkeit
  - ☐ Durchsatz
  - ☐ Verfügbarkeit
- ☐ Alle Änderungsanforderungen hinsichtlich ihrer Auswirkungen auf die Performance geprüft

ENTWICKLUNG (PERFORMANCE-ORIENTIERT)

- ☐ Performance-Test in der Entwicklung (d. h. Linksverschiebung)
- ☐ Performance-Test ist in Verbindung mit APM ein Bestandteil des Standardprozesses
- ☐ Performance-Test ist in Verbindung mit APM ein Bestandteil der nächtlichen automatisierten Batch-Tests
- ☐ Die Anzahl der Performance-relevanten Schwachstellen wird getrennt von Funktionsfehlern erfasst
- ☐ Der Anstieg oder Rückgang der Anzahl wird überwacht
- ☐ Performance-Überlegungen sind wesentlicher Bestandteil der Sprint-Planung
- ☐ APM wird für das Monitoring von Testumgebungen der Entwicklung eingesetzt

TEST (PERFORMANCE-ORIENTIERT)

- ☐ Performance-Test in der Qualitätssicherung
- ☐ Performance-Test ist ein wesentlicher Bestandteil des APM-Standardprozesses
- ☐ Die Anzahl der Performance-relevanten Schwachstellen wird getrennt von Funktionsfehlern erfasst
- ☐ Der Anstieg oder Rückgang der Anzahl wird überwacht
- ☐ APM kommt beim Monitoring der Testumgebungen in der Qualitätssicherung zum Einsatz

BEREITSTELLUNG (PERFORMANCE-ORIENTIERT)

- ☐ Bei der Konzeption festgelegte formelle Leistungskennzahlen sind obligatorische Entscheidungsgrundlagen im Release-Prozess
- ☐ APM wird für das Monitoring der wesentlichen Applikationen in der Produktivumgebung verwendet
- ☐ APM wird bei allen wesentlichen Applikationen für das Monitoring des Endnutzererlebnisses verwendet
- ☐ Die bei der Konzeption festgelegten Leistungskennzahlen dienen als Richtwerte für die Applikationsleistung im laufenden Betrieb
- ☐ Die Qualität jedes Release wird auf der Grundlage der gemeldeten Performance-Probleme gemessen

SKALIEREN (PERFORMANCE-ORIENTIERT)

- ☐ Die Anforderungen hinsichtlich der Skalierbarkeit der Applikation zum Zeitpunkt der Veröffentlichung und im Hinblick auf den künftigen Bedarf sind bekannt
- ☐ Die Applikation wurde für Skalierbarkeit ausgelegt

Die vorstehende Checkliste ist ein nützlicher Leitfaden für die Umsetzung einer effektiven Lösung für das Application Performance Management in Ihrem Unternehmen.

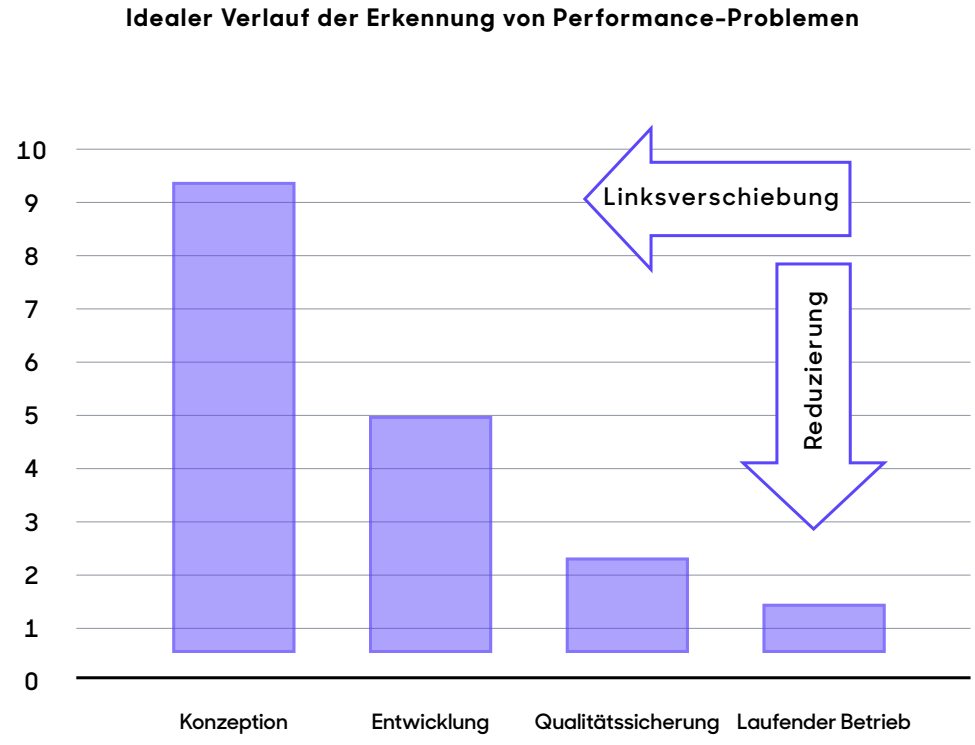
Die folgenden drei Leistungskennzahlen sind der ultimative Maßstab für den Erfolg Ihrer Maßnahmen:

1. Anzahl und Lokalisierung von Performance-Problemen:
   Idealerweise treten Performance-Probleme seltener auf und werden nicht erst im laufenden Betrieb oder bei der Qualitätssicherung sondern bereits bei der Konzeption und in der Entwicklung erkannt, sobald das DevOps-gesteuerte Performance Management Wurzeln schlägt und weiterentwickelt wird.

2. Die Kürze des Entwicklungszyklus und die Qualität der Applikation:
   Kürzere Entwicklungszyklen und eine höhere Qualität der Applikationen sind ein Beleg für die Vorteile der Linksverschiebung, die ein früheres Erkennen von Performance-Problemen und eine Reduzierung der Anzahl der Performance-Probleme ermöglicht.

3. Zufriedene Endnutzer:
   Software Releases, die regelmäßig und pünktlich bereitgestellt werden und die Erwartungen des Unternehmens und der Endnutzer hinsichtlich der Performance noch übertreffen, sind letztendlich der entscheidende Beleg dafür, dass das Application Performance Management in Ihrem Unternehmen bestens aufgestellt ist.

Abb. 3 – Erkennen von Performance-Problemen – Linksverschiebung



FAZIT

Performance-Tests sind unerlässlich, um die Leistungsfähigkeit einer Applikation zu gewährleisten. Mittels APM lässt sich die Wirksamkeit von Performance-Tests deutlich verbessern.

Wir haben untersucht, wie die mittels APM gewonnenen Erkenntnisse im gesamten SDLC dazu beitragen können, dass Applikationen schneller und in besserer Qualität bereitgestellt werden können.

Darüber hinaus haben wir typische Performance-Probleme und die Gründe für ihr Auftreten in den Blick genommen und Vorschläge dazu gemacht, wie sie sich vermeiden lassen.

Das APM ist zweifellos den Kinderschuhen entwachsen und ist bei der Planung von Performance-Tests inzwischen von wesentlicher Bedeutung. Informieren Sie sich, wie AppDynamics auch in Ihrem Unternehmen zur Optimierung der Performance-Tests beitragen kann.