

Niemand versteht mich – Ich verstehe meine Techniker nicht

Matthias Schulz,
Schulz IT Services GmbH
Nürnberg

Schlüsselworte

Kommunikation; Management; Technik; Entwickler; Planung; Verständnis; Missverständnis; Oracle Datenbank; IT-Probleme; Lösungen; Praxisbeispiele; Wartezeiten; Skalierbarkeit; Performance; Arbeitsteilung;

Einleitung

Verstehen Sie (Manger, Analyst, Betrieb, ...) ihre Techniker?

Werden Sie (Entwickler, Administrator, DBA, ...) nicht verstanden?

Nein?

Dann besuchen Sie diesen Vortrag und erfahren Sie, wie man komplizierte technische Sachverhalte auch anders erklären kann.

Themen

1. Ein Bild sagt mehr als tausend Worte...
2. Ein schlechtes SQL reicht
3. Alle Wege führen nach Rom
4. Das passende Werkzeug
5. Durchsatz oder Geschwindigkeit?
6. Entwickeln Sie ein schnelles System! Wofür? Egal!
7. KIWI (Kill It With Infrastructure) kann auch schiefgehen
8. Mach es schneller
9. Wie auf Schienen

Ein Bild sagt mehr als tausend Worte...

Totales Unverständnis, eine Situation, die bestimmt jeder in seinem Berufsleben schon einmal erlebt hat.

Eigentlich ist die Sache klar, die Fakten liegen auf dem Tisch und dennoch versteht Ihr Gegenüber kein Wort.

"Warum werde ich nicht verstanden? Was erzählt die? Der versteht doch überhaupt nichts!"

Warum ist das so?

Warum werden wir nicht verstanden, obwohl wir die Sache doch ganz klar erklärt haben, denken wir zumindest. Wir mögen alles klar beschrieben haben, doch die Bilder und Informationen die wir transportieren passen nicht zur Gedankenwelt unseres Gesprächspartners. Und so entsteht Unverständnis.

Komplizierte Sachverhalte lassen sich mit technischen Beschreibungen oft schwer vermitteln. Wenn man jedoch eine Analogie aus der realen Welt verwenden kann, so wird das Bild oft klarer und verständlicher und dies nicht nur für Andere sondern auch für uns selbst. Ein Bild transportiert einen Sachverhalt wesentlich besser, als tausende Zeilen präziser technischer Beschreibung.

Ein schlechtes SQL reicht

Auf der Produktions-Datenbank ist ein Problem entstanden. Die Ursache ist schnell gefunden: ein einziges SQL-Statement hat alle anderen in Mitleidenschaft gezogen. Und nun stehen wir vor der Frage, wieso denn sowas passieren kann. Warum kann ein einziges SQL-Statement die gesamte Firma in Schwierigkeiten bringen? Müsste die Datenbank nicht mit sowas umgehen können?

Stellen wir uns eine Datenbank, oder auch einen beliebigen anderen Server, als Autobahn vor. Die Fahrspuren entsprechen dann den CPUs und die Fahrzeuge repräsentieren SQL Statements. Es gibt kleine schnelle Sportwagen, also schnelle Abfragen die nur wenige Daten berechnen, bis hin zu großen schweren 40 Tonnern mit Anhänger, also Statements die Massendaten verarbeiten. Solange sich alle an die Verkehrsregeln halten und sichere Fahrzeuge verwenden, läuft der Verkehr fließend und ohne Störungen.

Was passiert aber, wenn wenn nun ein Schwerlastler, beladen mit Ölfässern, mit abgefahrenen Reifen, auf nasser Fahrbahn, zu schnell fährt?

Er gerät ins Schleudern, kippt um und blockiert alle Fahrbahnen. Zu allem Überfluss, laufen auch noch die Ölfässer aus. Was nun folgt ist jedem von uns nur allzu gut bekannt, ein riesiger Stau auf der Autobahn. Nichts geht mehr und das für Stunden.

Aber auch wenn es sehr ärgerlich ist, stundenlang im Stau zu stehen, kämen wir nicht auf die Idee, der Autobahn die Schuld für den Unfall zu geben. Wir würden auch nicht fordern, dass jedes Fahrzeug eine eigene Straße bekäme, denn wir wüssten dass dies viel zu teuer und daher unmöglich ist.

Stattdessen würden wir uns über den unverantwortlichen LKW-Fahrer beschweren. Wie kann man mit so einem unserem Fahrzeug überhaupt auf die Autobahn? Wie kann man bei Nässe viel zu schnell fahren?

Auf die Datenbank übertragen, stellen wir nun die richtigen Fragen, um derartigen Problemen vorzubeugen:

Wie kann man ein so schlecht geschriebenes SQL-Statement, eines das sich an keine Regeln hält und unsicher ist, auf die Produktionsdatenbank loslassen? Woher kommt es? Und warum ist es im Test nicht aufgefallen?

Alle Wege führen nach Rom

Alle Wege führen nach Rom, dieses Sprichwort ist uns allen bekannt. Wenn wir also nach Rom wollen, müssen wir nur losfahren, wir werden schon ankommen. Und wenn es nicht schnell genug geht, dann nehmen wir eben ein Flugzeug. Auf die IT übertragen heißt das, wenn wir unser Ziel nicht schnell genug erreichen, dann kaufen wir eben schnellere Hardware. Das kostet zwar etwas mehr, aber der Erfolg gibt uns ja schließlich recht.

Was wäre, wenn wir ohne große Investitionen noch viel schneller ans Ziel kämen?

Wie das geht? In dem wir uns einmal den Weg zu unserem Ziel genauer ansehen. Wir sind zwar in Rom angekommen, aber unsere Route hat uns Richtung Norden, über Nord- und Südpol, einmal um die ganze Welt geführt. Mit dem Flugzeug ging das wesentlich schneller, als mit dem Auto und der Fähre, aber es war immer noch der gleiche riesige Umweg. Hätten wir stattdessen einfach auf eine Landkarte gesehen, hätten wir entdeckt, dass es Richtung Süden einen wesentlich kürzeren Weg nach Rom gibt. Und wir wären mit dem vorhandenen Fahrzeug auf der kürzeren Route, deutlich schneller am Ziel, als mit dem teureren Flugzeug, das einmal um die ganze Welt fliegt.

Und wenn das Flugzeug schon gekauft ist?

Dann fliegen wir, jetzt da wir die kürzeste Route kennen, eben direkt Richtung Süden nach Rom und kommen noch schneller zu Ziel.

In unserer IT Landschaft stehen Auto und Flugzeug für die Hardware, also für Server, Speicher, Netzwerke und der gleichen. Landkarte und Fahrtroute, stehen für die Software. In vielen Fällen lohnt es sich also, bei Performance Problemen, erst mal einen Blick auf die vorhandene Lösung zu werfen und nach Optimierungspotenzial zu suchen, bevor Geld für neue Hardware ausgegeben wird.

Das passende Werkzeug

Wenn Sie im Garten arbeiten und ein kleines Loch für eine Pflanze ausheben, dann nehmen Sie eine Handschaufel und einen Eimer. Wenn sie hingegen eine Baugrube für ein Haus ausheben sollen, so werden sie höchstwahrscheinlich einen Bagger und einen Kipplader verwenden.

Aber Sie kämen wohl kaum auf die Idee, den Aushub des Baggers mit Ihrem kleinen Eimer wegzuschaffen, oder die Baugrube mit ihrer Handschaufel zu graben. Genauso wenig wie Sie sich einen Bagger beschaffen würden, nur um eine Rose einzupflanzen. Es ist für uns völlig normal, in unserer Alltagswelt das richtige Werkzeug für eine Aufgabe zu wählen. Und niemand käme auf die Idee zu sagen, wenn man einen Bagger hat, dann braucht man doch keine Handschaufeln mehr. Oder, wir haben doch Eimer wozu brauchen wir dann noch einen Kipplader?

Nur weil ein Bagger und eine Handschaufel graben können, sind sie noch lange kein Ersatz für einander.

Und genauso verhält es sich auch in unserer IT Landschaft. Nur weil zwei Werkzeuge ähnliche Aufgaben lösen können, heißt das noch lange nicht, das man auf eines davon verzichten kann, schon gar nicht wenn, Umfang und Details der Aufgabe völlig verschieden sind. Schlagworte, die in diesem Kontext vorgebracht werden, lauten Standardisierung, Cost Containment, Vereinfachung und andere.

Das Ergebnis ist leider oft das Gegenteil. Wenn wir nicht das passende Werkzeug für eine Aufgabe wählen, dann graben wir Baugruben mit Handschaufeln.

Durchsatz oder Geschwindigkeit?

Durchsatz und Geschwindigkeit, zwei Begriffe die miteinander zu tun haben und doch sehr unterschiedlich sind.

Wenn sie auf einer leeren Autobahn Samstagnacht um 4:00 Uhr von Nürnberg nach München fahren, und das so schnell wie möglich, dann schaffen sie das womöglich in unter 1 Stunde. Und erreichen eine Durchschnittsgeschwindigkeit von 160 km/h.

Und nun versuchen Sie einmal die gleiche Durchschnittsgeschwindigkeit und Fahrzeit, morgens um sieben, im Berufsverkehr zu wiederholen. Es wird Ihnen wohl kaum gelingen, und wundern wird es Sie auch nicht.

Wenn Sie eine Resource für sich alleine haben, müssen Sie sich um anderen keine Gedanken machen. Wenn jedoch eine große Menge Menschen und Güter möglichst schnell von Nürnberg nach München kommen soll, so helfen Ihnen Sportwagen, als alleiniges Transportmittel, herzlich wenig.

Wenn sie jedoch Lastwagen, Busse und PKW mit mehr als zwei Sitzplätzen einsetzen, können Sie die Anzahl der Fahrzeuge deutlich reduzieren und haben weniger Stau. Lastwagen und Busse dürfen zwar nur maximal 80 oder 100 km/h fahren und sind damit langsamer als ein Sportwagen auf freier Strecke, dafür kommen jetzt alle in unter 2 Stunden an und stehen nicht 3 Stunden im Stau.

Wird nur die Geschwindigkeit auf freier Strecke gemessen, können Lastwagen und Bus nicht mit dem Sportwagen mithalten und würden somit als schlechtere Lösung ausgemustert.

Wenn es aber um den Durchsatz geht, also die Vermeidung von Staus, so sind die vermeintlich langsamen Busse und Lastwagen auf einmal das Mittel der Wahl.

In der IT müssen wir also darauf achten, das unsere Leistungsmessungen auch zu unseren Anforderungen passen. Wir dürfen also unsere Systeme nicht nur isoliert testen, sondern müssen die Leistung auch unter realen Lastbedingungen überprüfen.

Hier gilt es sowohl die Einzelgeschwindigkeiten als auch den Gesamtdurchsatz pro Zeiteinheit zu ermitteln und vor Optimierungen zu überlegen, welches dieser beiden oft gegensätzlichen Ziele wichtiger ist.

Entwickeln Sie ein schnelles System! Wofür? Egal!

Sie sollen ein neues schnelles Transportmittel entwickeln, aber niemand sagt Ihnen was Sie transportieren sollen oder wohin.

Sie machen sich also an die Arbeit und bauen einen Sportwagen. Wenn Sie nun also einen Expressbrief, z.B. nach München, zustellen sollen, haben Sie das passende Fahrzeug hierfür fertig.

Leider sollten Sie Europaletten nach Hamburg bringen und hierfür ist der Sportwagen äußerst ungeeignet. Also konstruieren Sie als nächstes einen Lastwagen. Die Transportaufgabe nach Hamburg wäre damit gelöst. Doch als sie freudig die neue Lösung präsentieren, erfahren Sie, dass Ihr Transportsystem auch Waren nach New York bringen soll.

Als nächstes entwerfen Sie ein Flugzeug, somit können Sie Waren nach Hamburg und nach New York liefern. Aber auch diese Lösung bringt sie noch nicht ans Ziel. Denn bei der Masse, die Sie transportieren sollen ist das Flugzeug viel zu teuer.

Schließlich erfinden Sie noch einen Transportcontainer, den Sie mit dem LKW nach Hamburg und von dort mit dem Containerschiff nach New York bringen können. Nach 3 Anläufen, hohen Gesamtkosten und viel Entwicklungszeit haben Sie endlich die passende Lösung gefunden.

Auch in der Softwareentwicklung ist es unmöglich eine kostengünstige und performante Lösung in der erwarteten Zeit zu entwickeln, ohne die genauen Ziele und Beschränkungen zu kennen.

KIWI (Kill It With Infrastructure) kann auch schiefgehen

Wenn ein System nicht schnell genug ist, um die anfallenden Aufgaben in der gewünschten Zeit zu erledigen, tendieren viele Firmen dazu, einfach schnellere Hardware anzuschaffen. Doch manchmal hat dies genau den gegenteiligen Effekt. Statt mehr, wird jetzt noch weniger pro Stunde verarbeitet.

Übertragen wir unser Beispiel auf eine Autobahn. Unsere Aufgabe ist es viele Fahrzeuge möglichst schnell von Nürnberg nach Rom zu bekommen. Aktuell benötigen die Fahrzeuge deutlich länger als geplant für diese Route.

Da wir den Unterschied zwischen Geschwindigkeit und Durchsatz verstanden haben, haben wir auch sofort eine Lösung für unser Problem: wir benötigen einfach mehr Fahrspuren. Da Geld keine Rolle spielt, erhöhen wir die Fahrspuren von zwei auf zwanzig.

Entgegen unserer Erwartung kommen nun weniger statt mehr Fahrzeuge pro Stunde an und die Fahrzeit des einzelnen Fahrzeuges hat sich deutlich erhöht. Wir stehen vor einem Rätsel. Doch im Gegensatz zur IT können wir ja einfach unsere Fahrer fragen: "Warum haben Sie denn so lange gebraucht? Wir haben doch die Autobahn massiv ausgebaut!".

"Wissen Sie, so schnell wie mit der neuen Autobahn waren wir noch nie am Brenner, aber so lange wie jetzt sind wir auch noch nie in Sterzing an der Mautstelle gestanden" lautet die überraschende Antwort.

Den Ausbau der Mautstelle haben wir völlig vergessen! Und jetzt kommen noch mehr Fahrzeuge in kürzerer Zeit an. So viele und von so vielen Fahrspuren, dass sie sich gegenseitig blockieren. Wir haben den eigentlichen Grund für die lange Fahrzeit überhaupt nicht beachtet, stattdessen haben wir mit einer massiven Investitionen das Problem noch vergrößert.

Bevor Sie also mit einer Investition in Hardware versuchen ein Problem zu lösen, empfiehlt es sich erst einmal die wirkliche Ursache und vorhandene Engpässe zu erkennen. Die Mautstelle auf unserer Autobahn, kann beispielsweise ein zu schwaches Speichersystem oder Netzwerk sein oder Software die nicht skaliert. Wenn Sie nun versuchen Ihr Problem mit mehr CPUs zu lösen, so machen Sie es in diesem Fall nur noch schlimmer.

Mach es schneller

"Unser System ist langsam. Können Sie es wieder schnell machen?"

"Was ist denn schnell?"

"Na eben so schnell wie bisher."

Entgegen unserem allgemeinen Sprachgebrauch ist schnell keine absolute Größe, sondern eine relative Angabe und ohne Vergleich wertlos. Gemessen an einer Schnecke, ist eine Ameise ziemlich schnell, und doch hätte eine Ameise wohl kaum eine Chance einen 100 m Lauf gegen einen Menschen zu gewinnen. Um also einschätzen zu können was langsam und was schnell ist, benötigen wir vergleichbare Messwerte.

Wenn ein Bericht, der bisher in 3 Minuten fertig war, 8 Stunden braucht, so ist er im Vergleich deutlich langsamer geworden. Läuft dieser Bericht nach Tuning-Maßnahmen in einer Minute durch, so ist er deutlich schneller als in der Vergangenheit.

Vergleicht man jedoch zwei unterschiedliche Berichte ohne genauere Informationen zu haben, so hat die Laufzeit allein keinen Wert: wenn ein Bericht, der eine einzige Zeile einer Tabelle berechnen soll, "nur" 1 Minute benötigt, so ist er im Vergleich mit einem Bericht der 1 Milliarde Zeilen der gleichen Tabelle in einer halbe Stunde verarbeitet dennoch langsam.

Wie auf Schienen

Software und Hardware werden oft getrennt entwickelt oder eingekauft. Kaum jemand macht sich Gedanken, ob nachher alles optimal zusammenpasst, schließlich ist ja alles standardisiert. Wenn dann nicht alles reibungslos funktioniert ist ein Schuldiger auch meist schnell gefunden. Doch die wirkliche Ursache wird gar nicht oder zumindest zu spät erkannt.

Stellen sich vor Ihre Softwareabteilung soll Transportmittel bauen und entwickelt Autos.

Ihr Hardwareeinkauf soll Transportwege kaufen und bestellt Zugschienen.

Beide haben ihre Aufgabe passend zur Aufgabenbeschreibung gelöst und halten sich auch an etablierte Standards. Nur leider passen Autos und Schienen nicht wirklich zusammen. Wenn Sie Glück haben, haben ihre Entwickler Geländewägen entwickelt, die wenigstens überhaupt auf dem Gleisbett fahren können. Aber optimal wird diese Lösung nie sein.

Wenn Sie also ihre Investitionen optimal nutzen wollen, so müssen Hardware und Software von Anfang an gemeinsam geplant und entwickelt werden.

Kontaktadresse:

Matthias Schulz
Schulz IT Services GmbH
Tauberstraße 28
D-90449 Nürnberg

Telefon: +49 (0) 911-384 9090
E-Mail: info@schulz-it-services.de
Internet: www.schulz-it-services.de