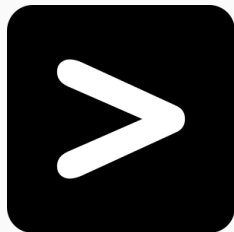


Praktische Administration

Das Drumherum



Dirk Deimeke

26. November 2016

My own IT / LinuxDay 2016

Prolog

Dirk Deimeke – d5e.org

- Mich interessiert – nicht nur aus beruflichen Gründen – die Administrationskultur («Sysadmin Culture»).
- Genauso interessiert mich auch der Themenbereich «Zeit- und Selbstmanagement»
(man sieht es mir nicht an, aber ich bin ein aktiver Mensch).
- Es gibt grosse Überschneidungen beider Interessensgebiete.
- Administration = Verwaltung
(damit betrifft es auch andere bzw. alle Bereiche des Lebens)

- Session «Praktische Administration» auf der deutschen Ubucon 2009.
- Fortsetzung auf der deutschen Ubucon 2010 (aufgrund von interessanten Diskussionen).
- Überwältigender Erfolg des Kapitels «Der Administrator» in «Linux-Server – Das umfassende Handbuch» (Rheinwerk Verlag)

Administration

Der Administrator

Klassische Teile der Administration

Architecture Bunte Bilder malen.

Engineering Realitätscheck und Überführung in Produktion.

Operation Betrieb

DevOps?

DevOps – unterschiedliche Definitionen, Teil 1

Administration von Systemen mit den Methoden,
die Entwickler benutzen
(Versionskontrolle, gut kommentierte Konfigurationsdateien, ...).
Machen wir schon lange, oder nicht?

DevOps – unterschiedliche Definitionen, Teil 2

Agiles Projektmanagement, Entwicklung in Sprints.

Machen wir auch schon lange,
Systeme werden weiterentwickelt und verbessert.

Leider haben Entwickler die cooleren Begriffe dafür erfunden.

Achtung bei Hype-Themen!

Jeder versteht etwas anderes darunter und vieles lässt sich **nicht** in kleinen Unternehmen umsetzen.

Methoden, die auf zehntausenden von gleichartigen Servern funktionieren, versagen eventuell in Eurem Unternehmen mit einigen zehn oder hundert Servern.

Am Rande:

Auf wie viele Definitionen des Begriffes «Cloud» kommt Ihr?

Infrastruktur als Code.

Das ist neu und das erste Mal, dass ein neuer Begriff sinnvoll ist.

Mit Konfigurationsmanagementsystemen lassen sich Zielzustände von Servern beschreiben.

Ansible, CFEngine, Chef, Puppet, SaltStack, ...

Diese Definition ist aktuell am gebräuchlichsten:

Entwicklern Produktionsverantwortung (Betrieb)
inklusive Zugriffsrechten geben.

Tut weh, rüttelt am Selbstverständnis,
ist aber meiner Meinung nach der richtige Weg.

Jetzt kann man verstehen, warum Docker wichtig ist.

Entwickler entwickeln ein neues Produkt und übergeben es dann **inklusive Weiterentwicklung** in den Betrieb.

Nicht weglaufen, mitmachen!

Deployments – Europa: Zurück auf «Los!»

Eine neue Software wird inklusive der Konfigurationen ausführlich getestet und dann zu einem vorher definierten Zeitpunkt in die Produktion übernommen.

Wenn etwas schief geht, wird die Produktionsumgebung auf den Ursprungszustand zurückgedreht.

In einem der kommenden Wartungsfenster wird ein neuer Versuch mit der dann hoffentlich fehlerfreien Version unternommen.

Deployments – Amerika: «Zurück in die Zukunft»

Eine neue Software wird inklusive der Konfigurationen ausführlich getestet und dann zu einem vorher definierten Zeitpunkt in die Produktion übernommen.

Wenn etwas schief geht, wird innerhalb des zur Verfügung stehenden Zeitraumes versucht, die Software trotzdem in die Produktion zu übernehmen.

Am «Point of no return» – Zeitpunkt, zu dem noch ein Zurückrollen auf den Ursprungszustand möglich ist, ohne das Wartungsfenster zu verlassen – wird entschieden, ob weiter versucht wird, die Software zum Laufen zu bekommen oder die Produktionsumgebung auf den Ursprungszustand zurückzusetzen.

In einem der kommenden Wartungsfenster wird ein neuer Versuch mit der dann hoffentlich fehlerfreien Version unternommen.

Üblicherweise haben Applikationen eine sehr begrenzte Anzahl von Wartungsfenstern pro Jahr (häufig nur zwei).

Major releases, das sind die mit richtig vielen Änderungen, erscheinen nur einmal alle paar Jahre.

Grössere Releases enthalten meist einige zehn vielleicht sogar einige hundert Changes und erfüllte Feature Requests.

Die Wahrscheinlichkeit, dass etwas (richtig) schief geht, ist nicht zu unterschätzen.

**Wenn Ihr Kunde wäret,
würdet Ihr dann gerne Jahr(e) darauf warten,
dass Eure Wünsche umgesetzt werden?**

Gelernt vom Open-Source-Mantra «Release early, release often» versucht man nun auch kleinere Änderungen direkt in Produktion zu bringen.

Auch dort gibt es eine Wahrscheinlichkeit, dass etwas schief geht.

Aber sowohl das Zurückdrehen, wie auch vorwärts verändern bis es geht, ist wesentlich weniger komplex.

**Grosse Firmen deployen täglich
kleine Änderungen in die Produktion.**

Wenn wir entscheiden dürfen:

**Patchen wir unsere Server
lieber täglich oder nur ein Mal im Jahr?**

**Wäre es nicht super, wenn das auch
in Distributionen einfließen würde?**

Rolling Releases!?

Begriff wird übrigens auch oft missverstanden.

Fähigkeiten

Auswahl der Fähigkeiten

Es gibt eine grosse Anzahl an Fähigkeiten, die für einen Administrator hilfreich sind, ich möchte hier nur auf zwei eingehen, aus denen sich viele andere ergeben.

Kommunikation!

Informationspflicht, «Melden macht frei»

Gerüchte vermeiden!

Zeitmanagement!

**«Alle Tage sind gleich lang, aber
unterschiedlich breit.»**

Something wrong on the internet?



nixCraft #

@nixcraft



Folge ich

"Being a good programmer or sysadmin is 3% talent 97% not being distracted by the internet."
-- unknown **#Truth #quoteoftheday #humor**

🌐 Übersetzung anzeigen

RETWEETS

245

GEFÄLLT

163



15:05 - 23. Nov. 2015

<https://twitter.com/nixcraft/status/668792577861120001>

- Die Zeit ist chronisch zu knapp.
- Die Nutzer denken, dass Ihr Problem das wichtigste ist.
- Projektarbeit will auch noch getan werden.
- Und wieder eine Störung.
- Jetzt klingelt auch noch das Telefon.

Zeitplanung mit ALPEN

- Aufgaben und Termine schriftlich festhalten,
- Länge der Bearbeitung realistisch schätzen,
- Pufferzeiten (wenigstens 40%) für Unvorhergesehenes,
- Entscheiden, was wegfallen oder delegiert werden muss und
- Nachkontrolle der Einschätzung im Rückblick

Prioritäten sind fremd bestimmt.

Vorgesetzte bitten, die Prioritäten zu setzen!

Ziele haben (sowohl beruflich als auch privat)

Privat **und** beruflich: 1 Monat, 1 Jahr, 5 Jahre.

Das hilft Chancen zu nutzen, wenn sie sich ergeben.

Ziele sind SMART

- Spezifisch
- Messbar
- Angemessen (oder erfordern Aktion)
- Relevant (oder Realistisch)
- Terminiert

Faulheit!

**Automatisierung, Monitoring,
Virtualisierung, Container, ...**

Business nicht aus den Augen verlieren!

Im Unternehmen

«Hassliebe»

- Business-Menschen verstehen selten, was wir wirklich tun.
- Wenn alles läuft, fragt sich das Business, warum so viel Geld ausgegeben wird.
- Wenn etwas nicht oder nur schlecht läuft, fragt sich das Business, warum es nicht läuft, obwohl so viel Geld ausgegeben wird.

Fazit

Wir können nicht gewinnen!

Eigentlich wäre der Vortrag jetzt zu Ende,
wenn es nicht Ansätze gäbe,
mit der Situation umzugehen.

Kommunikation!

Holen oder bringen?

Wir kennen die Entscheidungskriterien unserer Chefs oft nicht.

Häufig spielen Partnerschaften mit anderen Unternehmen eine Rolle, und manchmal wird eine strategische Ausrichtung über mehrere Jahre festgelegt, und die nachträgliche Einflussnahme ist sehr begrenzt.

Wenn Entscheidungen nicht verstanden werden: **Nachfragen!**

Kommunikation mit dem Chef

Es ist wichtig, dass Ihr mit Euren Chefs richtig kommuniziert und Eure Themen verständlich erläutert.

Ein unterschiedliches Wissensniveau darf kein Grund sein, nicht respektvoll und professionell miteinander umzugehen.

IT ist nicht in jedem Unternehmen das Kerngeschäft. Daher werden viele IT-Ausgaben als Geldausgaben ohne direkten Nutzen angesehen.

Daher in jedem Fall, sachlich und kompetent zu informieren, sodass das Anliegen verstanden wird.

Die Chefin¹ ist Dir gegenüber weisungsbefugt.

Die von ihr gesetzten Prioritäten können sich von den selbst gesetzten massiv unterscheiden.

Erklären, warum man die Prioritäten anders setzen würde.

Das Positive ist, dass Du durchaus auch die Chefin um Priorisierung bitten und sie als Eskalationsstufe nutzen kannst.

¹«Chefin» ist geschlechtsneutral gemeint

Wenn Du von rechtlichen Übertretungen oder schlimmstenfalls sogar von kriminellen Tätigkeiten erfährst, bist Du verpflichtet, das unverzüglich Deiner Vorgesetzten zu melden.

Das heisst nicht, dass Deine Vorgesetzte Dich zu kriminellen Handlungen zwingen darf.

«Unsere Systeme arbeiten am besten ohne Nutzer.»

Systemadministratoren verwalten die Werkzeuge oder die IT-Infrastruktur, die den Benutzern die Arbeit erleichtern oder ermöglichen. Bei allem Wissen, was dafür aufgewendet wird, sind es dennoch «nur» Werkzeuge.

Benutzer sind mangels IT-Wissen keine Menschen zweiter Klasse. Sie kennen ihr Fachgebiet genauso gut wie der Administrator seines, und in der Regel sind sie es, die die «eigentliche» Arbeit im Unternehmen erledigen. Damit ist die Arbeit gemeint, mit der das Unternehmen Geld verdient.

Ein «Gott-Komplex» ist unangebracht!

Versucht, eine Sprache zu finden, die der Endanwender versteht. Hört zu, und nehmt die Anfragen und Probleme ernst.

Kommunikation unter Systemadministratoren

Kommunikation unter Systemadministratoren ist – anders als gegenüber Vorgesetzten oder Benutzern – sehr stark fachlich geprägt.

Das bedeutet insbesondere, dass andere Meinungen akzeptiert werden müssen.

Diskussionen haben immer sachlich zu bleiben, denn nur so ist es möglich, das beste Resultat zu erzielen.

In Krisensituationen ist es notwendig, als Team zu funktionieren und nicht zu diskutieren.

Zu den wichtigen Aufgaben zählt es auch, Verfahren, Konfigurationen und bekannte Probleme und deren Lösungen zu dokumentieren, sodass keine Kopfmonopole existieren und im Fall, dass jemand Urlaub macht oder krank wird, die Arbeit erledigt werden kann.

Kein Mensch ist unersetzbar . . .

. . . es ist nur eine Frage des eingesetzten Geldes.

Epilog

Fragen?

Vielen Dank!

Dirk Deimeke, 2016, CC-BY

dirk@deimeke.net

d5e.org – speakerdeck.com/ddeimeke

PDF bei Speakerdeck herunterladen, dann sind die Links klickbar.

- Core Job Descriptions (PDF) der SAGE (System Administrators Guild)
- System Administrators' Code of Ethics
- Admin Zen
- Thomas A. Limoncelli, Zeitmanagement für Systemadministratoren
O'Reilly 2006, ISBN 978-3-89721-465-1