

08/23

Anzeige:



DOST

DEUTSCHE OPEN SOURCE TAGE

Cloud, Container, Automatisierung & Storage
27.-29. NOVEMBER IN BERLIN

Mehr Infos
auf der
Heftrückseite!

LINUX
MAGAZIN

LINUX

MAGAZIN

RSS-Feedreader im Direktvergleich s. 38

Akregator, Fluent Reader, Gfeeds, Liferea
und QuiteRSS: fünf Nachrichtenzentralen
für die grafische Benutzeroberfläche

K8s-Quirks s. 52

Wo der populäre Container-
Orchestrierer Kubernetes
noch Probleme aufweist

Vcpkg s. 76

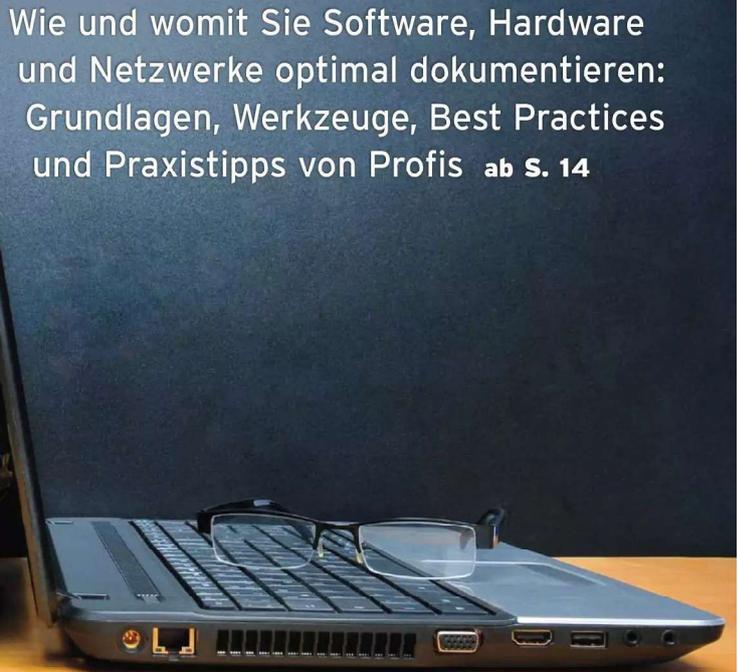
Einfach zu bedienender,
effizienter Paketmanager
für C/C++-Bibliotheken

Wissen strukturiert, systematisch und nachvollziehbar festhalten

PERFEKTE DOKU



Wie und womit Sie Software, Hardware
und Netzwerke optimal dokumentieren:
Grundlagen, Werkzeuge, Best Practices
und Praxistipps von Profis ab S. 14



■ WLAN-Health-Check mit Go s. 80

■ Concepts in C++20 ausreizen s. 86

Katastrophensicheres Backup s. 48 • KI-Basics s. 62 • Automatisches Schwachstellenmanagement s. 64

www.linux-magazin.de

Deutschland
€ 8,99

Österreich
€ 9,70

Schweiz
sfr 14,90

Benelux
€ 10,50

Spanien
€ 11,00

Italien
€ 11,00





14 Zu den wichtigsten **Grundlagen** einer guten Dokumentation zählen eine strukturierte Herangehensweise und eine saubere Zielgruppenanalyse.



28 Mit dem richtigen Werkzeug geht die Arbeit mühelos und fast wie von selbst von der Hand. So lässt sich die Hardware- und Netzwerkdokumentation mit **Netbox** und den richtigen Plugins effizient und komfortabel erledigen.



48 Ein katastrophensicheres **Backup** für Ihre Daten stellen Sie mit Werkzeugen wie Rsnapshot und Restic ohne großen Aufwand auf die Beine.

Aktuell

News 6

- Debian 12 „Bookworm“ ist fertig
- Rust mit Sparse-Protokoll
- Unkaputtbares Fedora Onyx
- C-Compiler im Boot-Sektor
- Gitlab setzt auf KI
- OpenInfra Foundation plant für Europa
- Infinite Scale mit „totem Briefkasten“

Zahlen & Trends 10

- Red Hat pflegt Libre Office nicht weiter
- BSI startet Projekt für sichere Kommunen
- AlmaLinux erhält Verstärkung
- Jahresbericht Telekommunikation 2022
- Warnung: KI gefährlich wie Atomkrieg
- Leap 15 erhält letzte weitere Version
- Streik bei Stack Overflow wegen KI

Service

Editorial 3

IT-Profimarkt 90

Usergroups 92

Inserenten/Autoren/Events 96

Impressum 97

Vorschau 98

Titelthema

Einführung 14

Ein Erfolgsfaktor für gute Dokumentation ist eine strukturierte und systematische Herangehensweise. Dieser Beitrag vermittelt Grundlagenwissen darüber, wer was und wie dokumentieren sollte und wie man dabei den roten Faden nicht verliert.

Best Practices 20

Gestandene Praktiker von Suse und Owncloud berichten über ihre Einsichten in die erfolgversprechendsten Dokumentationsmethoden und ihre Erfahrungen im Umgang mit den entsprechenden Tools.

Netbox 28

Netbox hat sich als Werkzeug mit einer Zwitterfunktion für DCIM und IPAM schon lange etabliert. Wer jedes verbaute Teil in seinem Rechenzentrum einfach und sicher dokumentieren will, findet in Netbox dafür die perfekte Lösung.

38 Für Nachrichten aus mehreren Quellen sind RSS-Feeds erste Wahl. Die **Bitparade** nimmt einige grafische Reader unter Linux unter die Lupe.

Software

Einführung 35

Auf der DELUG-DVD finden Sie diesmal Debian 12 „Bookworm“, 40 Vorträge von den Chemnitzer Linux-Tagen 2023, vier Dokumentationshelfer, OCIS 3, Memtest86+ und vieles andere mehr.

Tooltips 36

History-Datenbank Atuin 14.0.1, CLI-Rechner Calculator, Code-Differ DiffSitter 0.8.0, Text-Wiki Mdwiki 0.1.0, Webtacho P0d 0.4.0, RSS-FeedReader Trss 0.3.

Bitparade 38

Für den simultanen Empfang themenspezifischer Nachrichten aus mehreren Quellen sind RSS-Feeds nach wie vor die erste Wahl. Wir haben uns einige der grafischen Reader unter Linux näher angesehen.





Grundlagen der Softwaredokumentation

Baustelle Doku

Ein wesentlicher Erfolgsfaktor für gute Dokumentation ist eine strukturierte und systematische Herangehensweise. Dieser Beitrag vermittelt Grundlagenwissen darüber, wer was und wie dokumentieren sollte und wie man dabei den roten Faden nicht verliert. Anna Lehmann

Wen sollte man fragen, wenn man sich einen Überblick über Softwaredokumentation verschaffen will? Am besten vielleicht die Berufsgruppe, die sich ganz dem Thema der Dokumentation verschrieben hat: die Technischen Redakteure. Einen Technischen Redakteur zeichnet besonders aus, dass er nicht irgendeinen Text verfasst, sondern einen, der nützliche Informationen zum Produkt – in diesem Fall zur Software – so

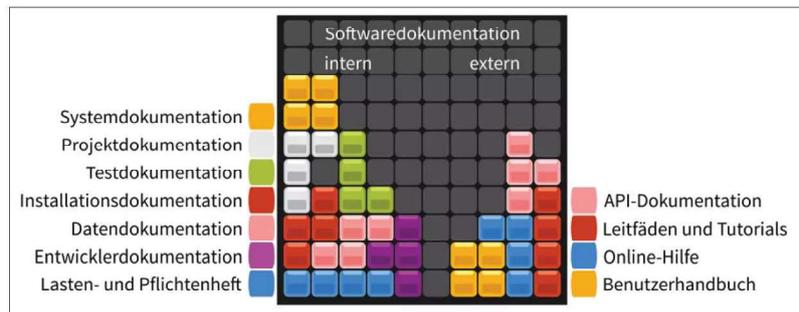
zusammenfasst, dass der Leser sie leicht aufnehmen kann. Zu diesen nützlichen Informationen gehören jegliche für die Zielgruppe relevanten Daten, die für die Entwicklung, Nutzung, Pflege und Unterhaltung sowie Aktualisierung nötig sind.

Und schon ist da ein für die Dokumentation außerordentlich relevantes Stichwort aufgetaucht: Zielgruppe. Sobald es etwas zu dokumentieren gibt, rücken auch diejenigen in den Fokus, die mit

den Informationen umgehen müssen. Daher stellt sich für Technische Redakteure neben dem Was auch gleichzeitig die Frage: Für wen? Schon allein dieses Kriterium unterteilt Technische Dokumentation in verschiedene Kategorien.

Außerdem kommt es noch auf die Form an: Wie wird dokumentiert? Das kann Text sein, Illustrationen, Grafiken, Mindmaps und so weiter. Je nach Art der Dokumentation kann sie entweder in den Quellcode eingebettet sein oder die Software begleiten. Wichtig ist, die Informationen so aufzubereiten, dass sich nachvollziehen lässt, wie das Softwareprogramm zu verwenden ist. Der Zweck der einzelnen Komponenten sollte klar werden und auch ein Überblick über die Möglichkeiten (und Grenzen) des Programms gegeben werden. Je nachdem, was die Software alles kann, gehört auch dazu, dass die Funktionsweise erklärt wird. Das betrifft unter Umständen auch Resultate oder Prozesse, an denen die Software beteiligt ist. Es gilt zu erläutern, wie das Programm zu bedienen ist, welche technischen Voraussetzungen gegeben sein müssen und welche Tätigkeiten gegebenenfalls sonst auch noch auszuführen sind.

Immer wieder höre ich als Technische Redakteurin von Softwareentwicklern, dass der Code doch schon ausreichend erkläre, was da passiert. Das kann für jeden, der sich den Quellcode ansieht, auch durchaus eine theoretische Möglichkeit sein. Auf's Ganze gesehen ist das aber etwas zu kurz gedacht. Schauen wir also zuerst einmal genauer darauf, welche Arten von Softwaredokumentation es gibt: Dann wird klar, dass der Code allein nicht die komplette Doku sein kann **1**.



1 Das sind die Bausteine interner und externer Softwaredokumentation.

Interne Dokumentation

Generell unterscheidet man zunächst zwischen interner und externer Dokumentation. Interne Dokumentation bezeichnet all das, was beim Softwarehersteller verbleibt. Das ist nicht weniger wichtig, häufig ist allerdings die Form etwas freier. Vor allem muss diese Art von Dokumentation nicht mit dem Marketing abgestimmt werden. Auf diesen Umstand kommen wir bei der externen Dokumentation zurück.

Ganz zu Beginn eines Projekts stehen häufig die Anforderungen, was die neu zu entwickelnde Software denn tun soll. Manche kennen das unter dem Namen Lastenheft. Teilweise wird diese Dokumentation auch agil erweitert, weil im Entstehungsprozess nicht von Anfang an alle Details klar sind. Bevor es dann ans Coden geht, wird – mal genauer, mal grober, mal agil und schrittweise, mal als ein großes Ganzes – festgelegt, wie diese Anforderungen umzusetzen sind. Das ist dann entweder als einzelne Tasks notiert oder bildet ein ganzes Pflichtenheft. Das hängt immer davon ab, wie umfangreich die Software ist, ob es spezifische vertragliche Bedingungen gibt und ob der Kunde eine ausführliche Antwort vor Programmierbeginn wünscht – um nur einige der Faktoren zu nennen, von denen die Art und Weise abhängt.

Meistens wird ein großer Teil der internen Dokumentation von den Softwareentwicklern selbst erstellt. Hier ist also die Aussage gar nicht so weit hergeholt, dass der Code die Doku sei. Allerdings kann das nur gelingen, wenn der Code mehr als die reine Aneinanderreihung von Funktionen ist. Das bedeutet, dass bestimmte Komponenten oder Abschnitte mit Kommentaren versehen sein soll-

ten. Nur so kann man sie auch als dokumentiert ansehen. Das Ergebnis bezeichnet man dann häufig auch als Entwicklerdokumentation. Diese Doku richtet sich vor allem an die Zielgruppe der Programmierer, die künftig an dem Code arbeiten sollen oder müssen. Sie umfasst Informationen über Architektur, Codestruktur und Verwendung der verschiedenen Funktionen des Programms.

Neben der Entwicklerdokumentation gibt es, sofern die Software Daten benötigt oder erzeugt, auch die sogenannte Datendokumentation. Hier ist es wichtig, dass klar wird, welche Daten das Programm benötigt, um zu funktionieren. Was ist wie eingebettet, was wird wo abgelegt, und welche Daten kommen als Output heraus? Falls die erzeugten Daten an bestimmten Stellen zu speichern sind, braucht es dann spezifische Formate?

Wenn es um umfangreichere Software geht, kann es sein, dass zusätzlich eine Installationsdokumentation erstellt werden muss. Sie sollte nachvollziehbar beschreiben, wie und in welcher Reihenfolge Systemkomponenten installiert und eingerichtet werden müssen, damit am Ende die Gesamtfunktion gewährleistet ist.

Etwas, das auch für kleinere Programme durchaus hilfreich sein kann, ist die Testdokumentation. Jeder Entwickler kennt das nur zu gut: Das Programm läuft super, man ändert an einer Stelle eine winzige Kleinigkeit – und schon treten Fehler auf. Hier hilft die Testdokumentation. Sie beschreibt die erforderlichen Tests, um die korrekte Funktion der Software zu überprüfen. Dazu gehört selbstverständlich auch, wie die Tests angelegt sind, worauf die Ergebnisse schließen lassen und wie mögliche Fehler, die

in den Tests aufgefallen sind, beseitigt wurden. So lässt sich am Ende auch nachweisen, dass man alles dafür getan hat, dass die Software sicher läuft.

Neben diesen sehr eng mit dem tatsächlichen Fortschritt bei der Softwareentwicklung verknüpften Dokumenten ist eine Projektdokumentation dafür da, die Planung und Organisation der Entwicklung festzuhalten. Dabei können beispielsweise geplante Ziele, genutzte Tools und Herangehensweisen beschrieben werden. Diese Art der Dokumentation geschieht häufig nebenbei und in Projekt-Meetings, weil mit mehreren Beteiligten mehrere Schritte zu absolvieren sind, in denen die neuen Aufgaben und Teilschritte zugewiesen werden.

Als letzten Punkt für die internen Dokumente wollen wir noch die Systemdokumentation anreißen. Sie beschreibt ein komplexeres System mit all seinen Komponenten, Funktionen und deren Zusammenwirken. Die Qualität dieser Dokumentation kann entscheidend dafür sein, ob der Technische Redakteur für die externe Dokumentation erneute Recherchen zu diesem Thema betreiben muss, oder ob er das als Grundlage für die Darstellung der Zusammenhänge bei der externen Dokumentation verwenden kann.

Externe Dokumentation

Die externe Dokumentation richtet sich an die Verwender der Software, die häufig nicht an der Entwicklung beteiligt waren. Deshalb geht es für den Technischen Redakteur nicht nur darum, den Entwicklern die Informationen aus der Nase zu ziehen. Stattdessen muss er sich auch

mit den Abteilungen abstimmen, die mit dem Kunden zu tun haben. Das ist einerseits das Marketing, das das Unternehmen oder die Produktmarke nach außen optimal repräsentieren möchte, andererseits aber auch der Vertrieb, der mit dem Kunden unter Umständen spezifische Abmachungen – auch bezüglich der Dokumentation – getroffen hat.

Externe Dokumentation sollte hilfreich sein, um zu verstehen, wie ein Programm zu verwenden und zu konfigurieren ist. Manchmal ist sie aber auch eine Quelle der Verwirrung und Frustration. Das gilt vor allem dann, wenn sie nicht sinnvoll redaktionell aufbereitet, verständlich geschrieben oder nachvollziehbar strukturiert ist. Also ist gerade hier die Gilde der Technischen Redakteure gefragt, das Wissen einerseits zu sammeln, andererseits auch zu analysieren und zu strukturieren – und vor allem: es verständlich für die jeweilige Zielgruppe wiederzugeben.

Zur externen Softwaredokumentation gehört zum Beispiel die klassische Benutzerdokumentation oder das Benutzerhandbuch, als PDF oder gedruckt, so wie es früher üblich war und es manche Verträge immer noch erfordern. Die Handbuchinhalte sollten sich nicht wesentlich von denen der Online-Hilfe oder kontextsensitiven Hilfe unterscheiden. Häufig differiert hier hauptsächlich die Art der Darstellung. Der digitale Raum bietet auch Möglichkeiten, die Inhalte spezifischer zu filtern, als das in einer ge-

druckten oder PDF-Version möglich wäre. Als Beispiel zeigt Abbildung 2 die Word-Hilfe von Microsoft. Mit einem Klick auf das kleine Fragezeichen öffnet sich direkt die Online-Hilfe, die unter support.microsoft.com zu finden ist. Sie enthält Screenshots, damit sich der Benutzer schnell zurechtfindet, sowie Handlungsanweisungen oder Tätigkeitsbeschreibungen (mit fett dargestellten Bezeichnungen aus der Software selbst) und Tipps & Tricks.

Eine weitere Möglichkeit, wie Informationen zum Anwender gelangen können, sind Leitfäden und Tutorials, die nicht alle Funktionen der Software abdecken, sondern nur spezifische Themen – die aber dafür sehr detailliert. Falls die Software über Drittanbieter auf den Markt kommt oder mit weiteren Programmen oder Daten interagieren soll, ist eine API-Dokumentation für die Schnittstellen unabdingbar. Sie erklärt vor allem, wie man die Schnittstelle nutzt und in welchen Formaten auf welche Art sie die Daten überträgt. Hier kann teilweise die Datendokumentation schon hilfreich sein.

Je nachdem, was der User mit der Software alles tun darf und kann, dürfen auch weitere Informationen nicht fehlen. So kann auch für den Administrator, der sich beim Kunden um die Einrichtung der Software kümmert, eine Testdokumentation hilfreich sein, sofern die Software Schnittstellen mitbringt und in Prozesse eingreifen könnte.

Entlang des Lebenszyklus

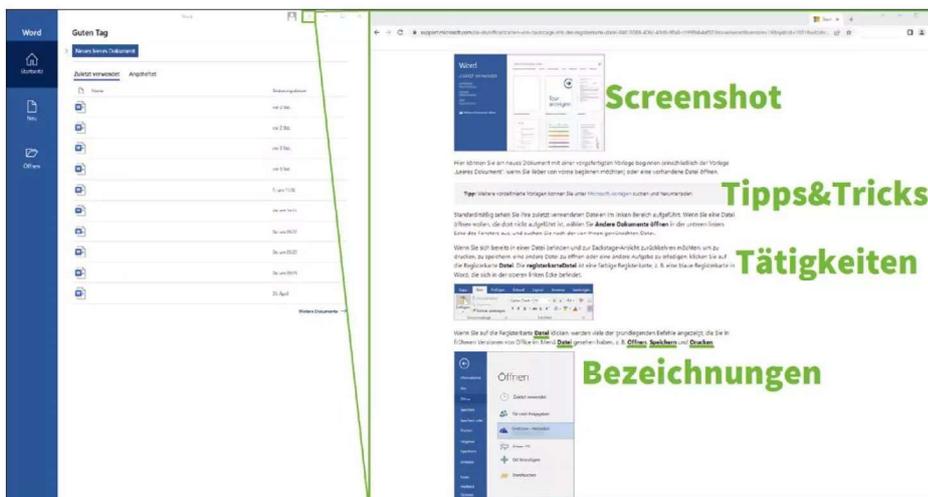
Alles in allem kann man sich auch bei der Softwaredokumentation an den Produktlebenszyklus halten. Von der Installation und Konfiguration über die Benutzung und Anpassungsmöglichkeiten bis hin zum Aktualisieren und Updaten und letztlich zur Deinstallation gibt es fast immer etwas, was der Nutzer wissen muss.

Die Installationsanleitung sollte beantworten, wie man das Programm auf einem Rechner einrichtet und welche Voraussetzungen erfüllt sein müssen. Meist richtet sich diese Anleitung an erfahrene Administratoren, die mit klassischem Softwarejargon gut klarkommen. Im Gegenzug dazu gehört in die Deinstallationsanleitung, was getan werden muss, um die Software sicher vom System zu entfernen, und ob es etwas zu beachten gilt, damit beispielsweise erzeugte Daten erhalten bleiben.

Auch die Konfigurationsanleitung wendet sich häufig noch an Administratoren. Sie beschreibt, welche Einstellungen man im Programm vornehmen kann und sollte, damit die User die Software dann bestmöglich nutzen können. Dazu kann auch gehören, wie sich einer Fehlbedienung vorbeugen lässt, sodass keine wichtigen Informationen verloren gehen. Auch häufig auf der Ebene für Admins angesiedelt sind die Informationen für Updates und Aktualisierungen, besonders bei komplexeren Programmen. Teil-

weise dürfen Anwender auch Updates direkt ausführen. Dann muss die Dokumentation klar und verständlich beschreiben, wie das Schritt für Schritt funktioniert. Außerdem stünde hier, falls es feste Zyklen für Updates gibt.

Benutzerdokumentation richtet sich, wie der Name schon sagt, an die User, die dann mit dem Programm arbeiten sollen. Sie zeigt auf, welche Möglichkeiten die Software bietet und wie man sie zu verwendet. Gerade bei Software mit einer grafischen Benutzeroberfläche (GUI) sollten



2 Aus diesen Elementen besteht die Online-Hilfe von MS-Word.



Screenshots den Anwendern eine Orientierungshilfe liefern. Ohne sie findet sich der User schnell nicht mehr zurecht und ist frustriert. Achten Sie darauf, dass Hervorhebungen klar aufzeigen, worum es geht, also etwa, welcher Button geklickt werden muss.

Besonders wichtig ist auch das Thema der Fehlerbehebung: Mit welchen Fehlern muss man rechnen, und wie behebt man sie? Gegebenenfalls gehört dazu, durch wen. Das ist etwa der Fall, wenn der User das nicht selbst kann oder sogar der Admin passen muss, weil er bestimmte Konfigurationen selbst doch nicht vornehmen darf, die in der Hand des Herstellers liegen.

Die Beschreibung der Anpassungsmöglichkeiten möchte oft vor allem der Vertrieb gern in jede Benutzerinformation integrieren, damit der Kunde möglichst weitere Funktionen zukaufen kann. Durch Addons können in diesem Fall mehr Aufgaben mit einem Programm erledigt werden, die dann allerdings häufig auch extra zu vergüten sind.

Anforderungen

Woher können Anforderungen an die Doku kommen? Zunächst gilt: Die Anforderungen an Dokumentation von Software unterscheiden sich nicht wesentlich von Anforderungen an andere Arten von Dokumentation. Grundsätzlich gilt es, die Regeln der Technischen Dokumentation einzuhalten:

- Die Dokumentation soll die Fragen der jeweiligen Zielgruppe beantworten. Dazu gehört etwa die Frage der User, wie das Programm genutzt wird, oder die der Entwickler, wofür welcher Teil im Code steht.
- Die Doku soll verständlich sein, dabei aber trotzdem so kurz wie möglich.
- Softwaredokumentation muss sich nach den Bedürfnissen und Aufgaben der Endanwender richten.
- Die Dokumentation erfolgt meist zusammen mit der Software.
- Auch die Doku muss möglichen rechtlichen Anforderungen genügen. Das gilt insbesondere, wenn die Software Teil eines Gesamtprodukts ist. Dann gelten hier die gleichen Regeln wie für die Hardware. Mögliche vertragliche Vereinbarungen gehören ebenfalls

zum rechtlichen Rahmen der Softwaredokumentation. Es gibt aber auch Anforderungen aus Normen, die versuchen, die Informationen für die Benutzung zu vereinheitlichen. Sie beschreiben konkrete Rahmenbedingungen, auf die sich Experten aus dem Fachgebiet geeinigt haben, damit der Standard – in diesem Fall für die Dokumentation – einheitlich und qualitativ hochwertig ist.

Es existieren verschiedene Herangehensweisen, um eine passende Benutzerdokumentation zu erstellen. Das bezieht sich auf aktuell gebräuchliche Methoden, an denen man sich als Dokumentierender orientieren kann. Wohl jeder Technische Redakteur hat die Zahlenfolge 82079-1 schon einmal gehört oder gesehen. Dahinter verbirgt sich eine international entwickelte Norm, die auf Deutsch beim DIN herausgekommen ist. Daneben haben daran das europäische Normengremium EN sowie internationale Organisationen wie IEC und IEEE mitgewirkt. Hier liegt der Fokus auf der Informationsvermittlung durch die Dokumentation für jegliche Produktarten. Diese Norm ist also für eine professionell erstellte Dokumentation generell relevant.

Außerdem lässt sich für den Bereich der Softwaredokumentation die Normenreihe ISO/IEC/IEEE 2651x mit Bezug auf Nutzerinformationen heranziehen. Sie spricht in den einzelnen Normteilen die jeweiligen Kontaktgruppen der Benutzerinformation (vom Manager, Einkäufer und Lieferanten, Tester und Gutachter bis zum Designer und Entwickler) und die agile Vorgehensweise an.

Wer schreibt?

Wir haben schon viel vom Technischen Redakteur gesprochen. Ausgebildete Mitglieder einer Technischen Redaktion sind darauf spezialisiert, dem Anwender der Software eine verständliche Anleitung zu geben, die sich einfach lesen lässt. In der Regel machen sich technische Autoren mit der Anwendung vertraut und testen die verschiedenen Funktionen. So nehmen sie automatisch den Blickpunkt des unerfahrenen Benutzers ein und sehen, welche Art von Information für diejenigen relevant ist, die eine Dokumentation lesen.

Bei der internen Dokumentation war dagegen häufiger vom Entwickler/Programmierer die Rede. Viele Entwickler erarbeiten die Softwaredokumentation selbst. Wer programmiert hat, weiß genauestens, wie die Software funktioniert. Das kann aber auch dazu führen, dass die Erklärung den technischen Horizont der Anwender oder neu eingestiegenen Entwickler weit übersteigt. Beim Erstellen der externen Dokumentation durch die Entwickler können leicht technische Details vorkommen, die eher verwirren als helfen. Um eine Software zu verwenden, muss der User die Feinheiten der inneren Funktionsweise meist nicht kennen, sondern nur wissen, wie Dinge zusammenhängen und welche Reihenfolge er im Fall der Fälle zu beachten hat.

Es kann aber auch sein, dass Fachanwender Software dokumentieren. Da sie regelmäßig mit der Software arbeiten, wissen sie in der Regel, was andere Anwender an Informationen benötigen. Allerdings sind solche Nutzer eher ein Glücksfall, und Softwareentwickler sollten sich nicht darauf verlassen, dass Benutzerdokumentation auf diese Weise entsteht. Es kommt jedoch durchaus vor, dass Anwender einzelne, spezifische Use Cases verbreteter Software etwa in Foren besprechen.

Im Idealfall dokumentiert also weder der Technische Redakteur noch der Entwickler allein die Software. Eine gute Mischung und Arbeitsteilung ermöglicht dem Team maximale Effizienz bei bestmöglichem Effekt.

Wie ist zu dokumentieren?

Es gibt keine allgemeingültigen Best Practices für Softwaredokumentation, da diese davon abhängt, was genau für wen dokumentiert werden muss. Allerdings gibt es einige allgemeine Tipps, die bei der Dokumentation eines Programms befolgt werden sollten:

- Einfach und klar: Leicht verständlich formulieren und je nach Zielgruppe frei von Fachausdrücken, sofern sie nicht geläufig sind.
- Prägnant: Nur für die Zielgruppe relevante Informationen dokumentieren und für das User Manual zum Beispiel alles weglassen, was nicht direkt mit der Nutzung der Software zu tun hat.

Rollen und deren Ziele	Beispiel Einsteiger	Beispiel Admin
Als Rolle, möchte ich das Ziel erreichen, um für das Unternehmen wertvolle Arbeit zu leisten.	Als Einsteiger will ich mich schnell in die Software einarbeiten, damit ich im Tagesgeschäft flexibel bin.	Als Admin will ich vor allem Informationen zu Installation und Einrichtung, damit ich die Software bereitstellen kann.

3 Die Zielgruppendefinition anhand von Rollen und Zielen.

- Bilder: Verwenden Sie möglichst Bildmaterial wie Screenshots oder Diagramme, um den Text zu ergänzen und komplexe Konzepte zu erklären.
- Logische Gliederung: Wenn die Struktur konsistent und leicht verständlich ist, lassen sich die Informationen schneller finden und aufnehmen.
- Einheitliche Formatierung: Wenn gleiche Informationseinheiten gleich aussehen, sind sie schnell wiederzufinden.

Anwender schätzen Links zu externen Ressourcen, die zu einem bestimmten Thema weiterführende oder detaillierte Informationen liefern.

Praktische Tipps

Entwickler erleichtern sich die Arbeit, wenn sie von Anfang an auch an der Dokumentation schreiben. So türmt sich nach Ende der Programmierphase nicht ein Berg Arbeit auf. Dafür muss klar sein, welchen Anwendungszweck die Dokumentation hat beziehungsweise an welchen internen Entwicklerkreis sie sich richtet. Von dieser Zielgruppe hängt auch der Umfang der Dokumentation ab. Daher gilt es, vorab genau zu überlegen, welche Informationen, Ressourcen, Datenbanken und Kommentare zum Quellcode benötigt werden.

Ein paar Beispiele für Inhalte wären Programmstruktur, Funktionen und Unterfunktionen, Auflistungen von Pro-

grammvariablen und zur Verwendung von Dateien. Diese müssen freilich nicht alle nur in der Entwicklerdokumentation enthalten sein, die Information kann durchaus auch in weiteren internen Dokumenten stehen.

Wenn man im Code dokumentiert, ist es wichtig, auch die möglicherweise existierenden Standards für die Dokumentation in der Programmiersprache zu beachten, wie zum Beispiel bei Java, Visual Basic oder C#. Häufig hilft bei besonders langem Quellcode eine zusätzliche Hilfedatei, in der relevante Schlüsselwörter referenziert werden.

Bei der externen Dokumentation sollte zuerst die Form der Dokumentation definiert werden. Ob klassisches gedrucktes Handbuch, PDF oder Online-Hilfe: Es gibt viele Varianten, die Informationen für die Softwarenutzung aufzubereiten. Auch hier ist die Zielgruppe maßgeblich. Sind die Nutzer selbst Softwareentwickler oder technikaffin? Sind sie mit komplexen Programmen vertraute Spezialisten? Oder handelt es sich bei den Usern der Software um Gelegenheitsanwender? Daran müssen die Detailtiefe und Komplexität der Dokumentation angepasst werden. Es gibt immer wieder Vorgänge, die man für unerfahrene Nutzer besonders genau beschreiben muss.

Klären Sie auch, welche Inhalte es genau bereitzustellen gilt, und ob unterschiedliche Zielgruppen existieren **3**. Für häufig auftretende Anwendungsfälle ist überdies eine Schritt-für-Schritt-Handlungsanweisung hilfreich, die den Nutzer genau anleitet.

Tools

Auch wenn feststeht, was und für wen geschrieben wird, bleibt möglicherweise noch die Frage offen, mit welchem Tool die Dokumentation erstellt wird. Hier wollen wir vor allem die externe Dokumentation betrachten, deren Aussehen

und Form auch den Eindruck der Software beim User beeinflussen kann. Bei den aufgezählten Tools handelt es sich um eine absolut zufällige Auswahl – es gibt noch viel mehr Werkzeuge, als wir hier aufzählen können.

Es liegt auf der Hand, Technische Dokumentation mit Autoren-Tools wie Microsoft Word oder Adobe Framemaker zu erstellen. Doch gerade für Software gibt es noch viele weitere Möglichkeiten. Da sind beispielsweise Help Authoring Tools (HAT) wie Madcap Flare oder Adobe RoboHelp. Dahinter verbirgt sich Software zur Erstellung von Online-Hilfen und anderen Informationsdokumenten, die oft auch druckbare Dokumente (PDF-Handbücher) in unterschiedlich guter Qualität ermöglicht.

Bei DITA-basierten Tools wie etwa DITA Open Toolkit handelt es sich um Werkzeuge, die auf der standardisierten XML-Struktur DITA basieren und speziell für die Dokumentation von Software entwickelt wurden. Sie eignen sich sehr gut für große Dokumentationsprojekte. In der Regel ist die Arbeit mit ihnen aber komplex und teuer, der Nutzen für kleinere Projekte unter einigen Hundert Seiten ist daher fraglich.

Component-Content-Management-Systeme (CCMS) oder auch Redaktionssysteme wie Author-it oder Schema ST4, basieren entweder auf DITA oder funktionieren auf Basis eigener Strukturen. Sie sind eher für sehr große Doku-Projekte vorgesehen. Für Projekte mit weniger als 1000 Seiten sind CCMS sehr kostspielig. Sie können sich aber bei einer gewissen Anzahl von Zielsprachen trotzdem schon für weniger Seiten lohnen.

Es ist auch möglich, ein mit Word oder Framemaker erstelltes Handbuch in eine HTML-basierte Online-Hilfe umzuwandeln. Soll auf die Schnelle eine Online-Hilfe erstellt werden, könnte ein Konverter wie Webworks ePublisher zumindest für eine erste Version hilfreich sein.

Unabhängig davon, welche Werkzeuge Sie konkret wählen, sollten Sie immer daran denken, dass Softwaredokumentation – wenn sie denn gut ist – die Effizienz neuer Entwickler bei der Einarbeitung in den Code sowie die Zufriedenheit Ihrer Kunden erhöhen kann. Das senkt vielleicht sogar die Zahl der Anfragen bei der Service-Hotline. (jcb) ■

Die Autorin

„Technikbegeistertes Organisationstalent“ – das beschreibt Anna Lehmann. Als Ingenieurin leitet sie bei den Handbuch-Experten Ausbildung, Qualitätsmanagement sowie Systematisierungen und begutachtet beim Tekom Dokupreis gern fremde Doku-Werke. Kunden profitieren von ihrem Überblick und dem Schaffen leicht einzuhaltender Strukturen.



© Wuttichai Kaewklang / 123RF.com

Gute Dokumentation effizient schreiben: Tipps von den Doku-Profis

Doku ist Teamwork

Worauf sollte man beim Dokumentieren besonders achten?

Wie integriert man die Dokumentation am besten in seine Prozesse? Welche Tools bieten sich an? Wie sehr sollte man ins Detail gehen? Das geben Doku-Experten von Suse und Owncloud hier zum Besten. Markus Feilner

Die Autoren von Softwaredokumentation sehen sich oft mit dem Vorurteil konfrontiert, Doku sei ja quasi keine technische Arbeit, kein IT-Job. Manche IT-Graubärte erklären Doku gar zur reinen Frauenarbeit. Nicht immer erfahren die Verfasserinnen und Verfasser technischer Dokumentation die Wertschätzung, die

ihr Job eigentlich verdient. Doch das Profil des Technischen Redakteurs – ein eigenes Studienfach an zahlreichen Universitäten – sieht viele Begabungen als erforderlich an.

Darunter finden sich einige, die auch für einen Job im Management nicht von Nachteil wären; zum Beispiel vernetztes

Denken oder die Fähigkeit, komplizierte Sachverhalte in einfache Sätze zu verpacken. Technische Redakteure sollten zwischen zwei grundverschiedenen Sachgebieten eine Brücke schlagen können und Einfühlungsvermögen sowie Sprachgefühl mitbringen. Sie müssen außerdem ein Gefühl für das Gestalten und Strukturieren von Texten haben, aber auch in der Lage sein, selbst dem schweigsamsten Kernel-Nerd die wichtigen Informationen aus der Nase zu ziehen.

Beruf mit Zukunft

Jeder, der schon einmal „in der Doku“ gearbeitet hat, kennt die ungläubigen Gesichter der Techniker oder Manager, wenn diese zum ersten Mal hören, es

gäbe Regeln für diesen oder jenen Satzbau oder für diese spezielle Textsorte. Oft ist auch schwer zu vermitteln, dass es einen großen Unterschied macht, ob man einen Text fürs Internet, Print, die Online-Hilfe oder für das PDF der High-Value-Kunden schreibt.

Das Portal [studieren.de](#) sieht – ChatGPT zum Trotz – für Dokumentationsexperten eine glorreiche Zukunft: „Technischen Redakteur*innen werden gute Zukunftsaussichten vorausgesagt: Zum einen liegt das daran, dass sie vielfältig einsetzbar sind, zum anderen, dass immer mehr Produkte auf den Markt geworfen werden, die einer Erklärung bedürfen. Einsatzmöglichkeiten finden sie bei fast allen Industrieunternehmen – vom Konsumgüterbereich bis hin zum Maschinen- oder Fahrzeugbau, in der Telekommunikation, der Computerbranche und im Bereich der Optik.“

„Kommunikation ist das A und O. Auch auf und zusammen mit den höchsten Managementebenen muss der Dokumentationsexperte in der Lage sein, Bewusstsein (Awareness) für die Wichtigkeit des Themas zu schaffen. Gerade die Chefetagen müssen verstehen, dass Dokumentation ein wesentlicher Bestandteil der Softwareprodukte und -lösungen ist, ohne den der Anwender kaum Erfolg haben wird.“

Das sei allerdings immer noch nicht überall angekommen, erklärt Meike Chabowski, Documentation Strategist beim Linux-Vendor Suse [1](#). „Wer Dokumentation nur als Cost Center sieht, hat ein Problem. Dokumentation ist ein grundlegender Service, den der Kunde mit dem Produkt mitbezahlt“, stellt sie klar. Ohne Zusammenarbeit mit anderen Teams geht da gar nichts. Experten aus allen beteiligten Teams müssen zusammenarbeiten, auch das Feedback von externen Fachleuten ist wichtig. Dazu meint Chabowski: „Nicht jeder kann schreiben, aber jeder kann zur Dokumentation beitragen – sei es durch Feedback, Anregungen, Fehler finden, Text korrigieren und vieles andere mehr.“

Was dokumentieren?

Gute Dokumentation wird immer wichtiger: Auch in Unternehmen und gerade im Kontakt mit Endgeräten nehmen vie-



1 Meike Chabowski ist Documentation Strategist bei Suse.

le Menschen Geräte und Software als immer komplizierter wahr, allen Fortschritten in Sachen UX/UI-Design zum Trotz. Firmen erkennen den Wert guter Dokumentation, sowohl für die Produkte als auch für die Prozesse innerhalb der Firma – Stichwort: Knowledge Management. Gerade bei Letzterem gewinnt das Thema Nachhaltigkeit mehr und mehr an Bedeutung.

Eine zweite Jugend erfuhr das Wissens- und Know-how-Management in Unternehmen spätestens 2021, als Marktführer Atlassian, Hersteller der in den meisten IT-Firmen eingesetzten Systeme Confluence, Jira und Trello, seine Preise massiv erhöhte und begann, Kunden in die Cloud zu zwingen. Da zeigte sich, wie teuer es werden kann, sich nur auf einen Monopolisten zu verlassen. Hersteller von Open-Source-Alternativen zu Atlassian-Produkten haben derzeit Konjunktur. Auch das Linux-Magazin zeigte [2](#), wie man es schafft, trotz des Trubels im Alltagsbetrieb eben jenen Alltag zu dokumentieren und die Prozesse und Informationen aufzuzeichnen, die die Mitarbeiter häufig brauchen. Dieselben Methoden greifen auch bei der Produktdokumentation.

Die vielen Anleitungen im Internet [3](#) sind sich ebenfalls einig: Gute Dokumentation fängt mit einem sauberen Prozess an, hört damit aber noch lange nicht auf. Meist braucht es drei bis fünf verschiedene Gattungen, die von Quickstart-Guides („Getting Started“) über Installations-, Admin-, User und Referenzdokumentati-

on (mit zum Beispiel allen API-Befehlen) bis hin zu monothematischen How-tos oder Best Practices reichen. Meist gibt es spezielle Tools für jede Gattung. Das Swagger-Toolset [4](#) beispielsweise hilft bei den einzigen Handbüchern, die sich halbwegs automatisiert schreiben lassen: API- und Referenz-Dokumentationen.

Klare Prioritäten

Die Prioritäten, was zu dokumentieren ist, sollte stets das Produktmanagement vergeben: Dort laufen die Fäden zusammen, was den Bedarf der Kunden und die heißesten neuen Features angeht.

Auf gar keinen Fall sollten Developer dokumentieren oder die Scrum Master entscheiden, was dokumentiert werden sollte. Sie haben fast immer nur die Innenansicht, nicht jedoch den Überblick. Das Was ist ohnehin meist ein Kompromiss aus verschiedensten Fragen: Was ist neu? Was muss man erklären? Was brauchen die Kunden (dem Vertrieb, den Kundenberatern und den Consultants zufolge)? Wofür stehen Ressourcen bereit, was hat Priorität?

Dass dabei zahlreiche Kollegen und Manager ein Wörtchen mitreden wollen und dürfen, macht die Sache nicht einfacher. Ein Dokumentationsexperte muss heute zwingend auch Fähigkeiten des Projekt- und Zeitmanagements, agiler Methoden und Verständnis für Unternehmensprozesse und -ressourcen mitbringen – und darüber hinaus eine saubere Triage beherrschen.

Die schönste Doku hilft niemandem, wenn sie nicht gelesen oder nicht benötigt wird. Sie wirft auch keinen messbaren ROI (Return on Investment) ab und kostet nur Geld – so ein oft gehörter Vorwurf. Aber ein guter Dokumentationsprozess steht im engen Austausch mit der Qualitätssicherung. Weil die Dokumentation eines neuen Produkts ja mit dem Produkt fertig sein muss, aber meist länger dauert, testen die Doku-Schreiber meist lange, bevor die QA-Kollegen anfangen. Auch gemeinsame Sprints sind denkbar – und manchmal dringendst nötig –, bei denen Development, QA und Doku sich zum Erstellen eines Handbuchs zwei Wochen in Klausur begeben.

Das sieht auch Martin Mattel [2](#) so: „Dokumentation ist nicht das Schreiben



2 Martin Mattel ist Unternehmensberater aus Wien und schreibt seit vielen Jahren an der Dokumentation für Owncloud.

von ein paar Sätzen, sondern das Ergebnis einer intensiven, kollaborativen Arbeit über mehrere Abteilungen.“ Der Wiener zeichnet verantwortlich für weite Teile der Dokumentation des Nürnberger Unternehmens Owncloud . „Dokumentation kann man als einen Aspekt von QA sehen. Man braucht Manpower und einen Know-how-Stack, dann muss man sich noch in die Schuhe von anderen Menschen hineinversetzen können.“ Es mache zwar einen großen Unterschied, ob man für Developer, User, Admins oder Mana-

ger schreibt. Am tiefen Verständnis der Sprache und etwas Talent für die richtige Formulierung komme man aber nicht vorbei, betont Mattel.

Qualifikationen

Ein guter Tech Writer zeichne sich aber auch durch proaktives Handeln, Interesse an der Sache (gerade auch in technischer Tiefe) und dem Spaß am Netzwerken mit anderen Abteilungen aus. Insel- und Silo-Denken müsse er oder sie aktiv vermeiden. Hartnäckigkeit, Lust an der „Forschungsarbeit“ und Spaß an der Fehlersuche seien ebenfalls zwingend notwendige Charaktereigenschaften, so Mattel.

„Dokumentation ist Forschungsarbeit, beschreiben, interne und externe Zusammenhänge erkennen, einbauen, abgrenzen und wieder referenzieren, wo nötig. Das wird aber ganz anders wahrgenommen: Fürs Management sind das lästige Kosten und morgen soll das eh ChatGPT machen. Für die Entwickler ist Dokumentation oft nur ein lästiger Zeitdieb. Aber für den Anwender ist die Dokumentation oft der einzige Zugang zu den gesuchten Infos.“

Für Sales und Vertrieb ist gute Doku ein starkes Argument, warum der Kunde das eigene wohldokumentierte Produkt kaufen sollte und nicht das schlecht(er) dokumentierte der Konkurrenz oder das

undokumentierte FOSS-Paket. Gerade bei Open-Source-Produkten wird Dokumentation immer wichtiger, weil hier potenzielle Kunden vielleicht schon wieder abspringen, wenn sie beim ersten Problem mit dem OSS-Projekt keine hilfreiche Beschreibung finden. Schon lange vor dem Erstkontakt mit der Herstellerfirma verliert diese so unter Umständen bereits Kunden. Ja, Dokumentation gehört auch zu den Services, mit denen OSS-Unternehmen wie Suse Mehrwert beweisen können **3**.

Doku als Visitenkarte

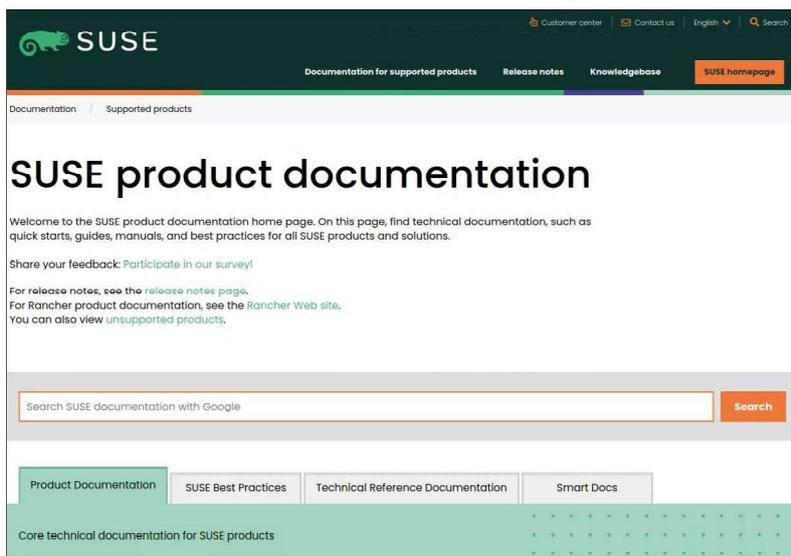
Dazu meint Martin Mattel: „Dokumentation ist eine Visitenkarte. Mehr noch als die Webseite spiegelt sie die Professionalität eines Unternehmens wider. Sie schafft Vertrauen und Zuversicht und ist die Basis für konkrete Fragen. Aber sie ist auch Teil des Brandings für das Unternehmen, weil sie ja direkt auf die Anwender, die Kunden hin ausgerichtet ist.“

Auf der anderen Seite habe sie auch großen Einfluss auf die Entwicklung, so Mattel: „Im Idealfall wird der Code und das Resultat von Anfang an aus einem anderen Blickwinkel gesehen, weil die Dokumentation das beschreibt, was ist und funktioniert und nicht, was man gern hätte. Auch hier findet der hilfreiche Perspektivenwechsel statt – und da müssen die Dokumentationsmitarbeiter viel mit dem Support und der Service-Abteilung reden, damit der Fokus auf die wirklich hilfreichen, manchmal auch herausfordernden Klarstellungen gelegt werden kann. Dann kann die Dokumentation auch die Basis für weiterführende Dienstleistungen werden.“

An dieser Stelle kommt dann auch der eigentliche Wert der Dokumentation zutage, meint Mattel. Und der werde von Marketing, Entwicklung, Kunden und Support sehr wohl gesehen, auch hinsichtlich eines ROI.

Klar definiertes Szenario

Der Teamlead Dokumentation bei Suse, Frank Sundermeyer **4**, ist bei der weltgrößten eigenständigen Linux-Firma seit bald zwei Jahrzehnten in der Dokumentation  tätig. Er hat eine klare Vorstellung, worauf es bei professioneller Doku-



3 Die Suse-Dokumentation ist in nach Zielgruppen gegliederte Sparten aufgeteilt.

mentation zu achten gilt: „Es braucht ein klar umrissenes und definiertes Szenario, eine klare Struktur mit Zielen, Voraussetzungen und Umsetzung. Das meine ich nicht nur theoretisch – das Dokumentierte als Schreiber auch in der Praxis nachvollzogen und verstanden zu haben, hilft ungemein, die Qualität der Dokumentation zu steigern.“ Gefragt, was denn die wichtigste Erfahrung sei, die das Suse-Dokumentationsteam in drei Jahrzehnten gemacht hat, erklärt Sundermeyer: „Hands-on'-Experience ist durch nichts zu ersetzen, und Technical Writing ist interdisziplinäre Teamarbeit.“

Dokumentation solle nach Möglichkeit stets parallel zum Schreiben in der Praxis getestet werden, betont Sundermeyer, weil sich so auch Fallstricke und Irrwege dokumentieren lassen, die ansonsten eventuell in Vergessenheit geraten. Auch die Entwicklung profitiert von dieser Vorgehensweise, da Technical Writer so nicht selten Fehler finden und rückmelden.

Hier kommt ebenfalls potenzieller Return on Investment ins Spiel: „Wir machen zum Beispiel die Suse Best Practices teilweise auch in Zusammenarbeit mit unseren Consultants und beschreiben darin besondere und verbreitete Kundenszenarien. Bei der Umsetzung beim Kunden muss das ja entsprechend dokumentiert sein, sodass die Doku aus der Alltagsarbeit schon fast von selbst herausfällt. Das Ganze kann dann anschließend ein Technical Writer bearbeiten und publikationsfähig machen.“

Das sagt auch Martin Mattel: „Dokumentation sollte immer eingebunden werden, ganz am Anfang und ohne Einschränkungen.“ Und Meike Chabowski bestärkt, dass es immer darum gehe, Bewusstsein zu schaffen bei allen Verantwortlichen, Entwicklern, Entwicklungsteams und Managern: „Wir dokumentieren nicht für euch oder uns, sondern für unsere Anwender. Ohne Dokumentation können die meisten Produkte nicht genutzt werden. Dokumentation ist ein grundlegender Service für den Kunden.“

Wie viele Details?

Auch der Tiefgang spielt eine wichtige Rolle. Frank Sundermeyer definiert Dokumentationen normalerweise als „Schritt-für-Schritt-Anleitungen, die für eine spe-



4 Frank Sundermeyer ist Teamleiter im Dokumentationsteam bei Suse.

zielle Zielgruppe geschrieben werden.“ Alle erforderlichen Grundkenntnisse müssen vorab in einem Bereich „Voraussetzungen“ abgeklärt sein. „Damit ergibt sich die Flughöhe eigentlich automatisch.“

Eine vorherige Analyse der Zielgruppe und der Leserschaft hält als Technical Writer auch seine Kollegin Julia Faltenbacher für enorm wichtig: „Vor dem Schreiben muss eine Zielgruppenanalyse durchgeführt werden, die die Vorkenntnisse und Bedürfnisse der Leser festhält, damit diese stets auch beim Schreiben beachtet werden können. Erst daraus ergibt sich dann zum Beispiel auch das Ausgabeformat.“ Ein Techniker, der in der Werkstatt vor einer Maschine steht, benötigt zwangsläufig ein anderes Format als der, der vor dem Büro-PC sitzt. Im Internet gelten dann noch einmal andere Regeln: Viele Teams behandeln für Suchmaschinen optimierte Texte (SEO) als eigene Ausgabe-kategorie.

Ohne Doku kein Produkt

Meike Chabowski mahnt: „Dokumentieren ist nicht nur 'einfach was schreiben'. Softwaredokumentation besteht zunächst überwiegend aus Recherche und Planung. Das Schreiben an sich ist der vom Aufwand her geringste Teil.“ Zwei Wochen Recherche, Testen und Kommunizieren für einen hilfreichen Absatz, der ein großes Problem vieler Kunden löst, kann da schon einmal vorkommen.

Ideal ist es, wenn schon die Entwickler eine eigene, interne Vorabdoku erstellen, auf der man aufbauen kann. Das muss aber schon in der Aufwandsplanung eingepreist werden. Eine Checkliste, auf der „Dokumentieren und Kooperation mit dem Doku-Team“ als verpflichtende Aufgabe (Mandatory Task) eingestuft ist, hilft da sehr. Ein Produkt ist erst fertig, wenn es Dokumentation gibt.

Zusammen mit den Entwicklern nehmen moderne Dokumentationsteams Teil am agilen Prozess der Softwareentwicklung, machen Milestones mit und nehmen an Sprints teil. Separate Doc Sprints oder Doc Days – Tage, an denen nur an der Dokumentation gearbeitet wird – helfen sehr, auch beim Erzeugen der Awareness. Kein Wunder, dass es ein gutes Zeichen für ein funktionierendes Dokumentationsteam ist, wenn es sich nahtlos in die Tools und Sprachen integrieren kann, die die Entwickler nutzen. Derlei Kenntnisse beschleunigen den Dokumentationsprozess enorm und reduzieren den Overhead in anderen Teams.

Aber auch die Techniker mit Anwenderkontakt müssen Zeit einplanen – und das ist oft teuer. Wenn Consultants, Sales Engineers, Support und Presales Feedback geben sollen, müssen sie Stunden dafür reservieren – Stunden, in denen sie erst einmal keinen Umsatz für die Firma machen. Aber genau diese wertvolle Zeit, in der das Feedback aus der Zusammenarbeit vor Ort mit dem Kunden und bei der Implementierung von Lösungen in die Doku fließt, ist bares Geld wert. Derlei können kein Softwareentwickler und kein Technischer Redakteur liefern, weil ihnen die Real-Life-Erfahrung mit den Produkten oder Lösungen fehlen.

Gute Doc-Teams machen es den Zulieferern so einfach wie möglich – egal, ob diese in Sales oder Development arbeiten, egal, welche Expertise in Sachen Dokumentation vorliegt. Ein gutes Dokumentationsteam sollte nach außen werkzeug- und formatagnostisch sein. Jeder Input ist willkommen, egal in welcher Art von Datei.

Tools, Tools, Tools!

Thomas Schraitle, ebenfalls seit vielen Jahren im Dokumentationsteam von Suse tätig, legt den Fokus noch mehr auf

Sprache	Übersetzt	Nicht übersetzt	Nicht übersetzte Wörter	Qualitätsprüfungen	Vorschläge	Kommentare
Afrikaans	27%	21.596	167.398	3.063	1	
Albanisch	14%	14.508	126.016	232	6	
aln (generated) (aln)	0%	2.458	20.485			
Amharisch	1%	14.531	117.353			
Arabisch	22%	97.029	1.762.583	1.714	209	1
Armenisch	29%	1.342	12.331	9	1	
Aserbaidshianisch	5%	985	3.199	16		
Assamesisch	39%	1.484	13.690	190		
Asturisch	12%	16.390	141.976	599	7	
Baskisch	18%	23.505	176.746	570		

5 OpenSuse nutzt das Übersetzungsmanagementwerkzeug Weblate, um die diversen Übersetzungen und Lokalisierungen effektiv zu koordinieren.

die technischen Aspekte: Ihm sind Konsistenz, Zielgruppenbezug sowie eine stimmige, gute Struktur wichtig, all das in einem möglichst ansprechenden, benutzerfreundlichem Layout. „All diese Faktoren bestimmen, wie deine Dokumentation wahrgenommen wird.“ Bei der Integration in die Prozesse helfen Style Guides, Definitionen, Vorlagen und Templates, aber auch klare Prozessdefinitionen und Softwarearchitekturen wie Continuous Integration.

Dazu sagt Schraitle: „Alle erwarten aktuelle Dokumentation, jeder sollte zur Doku beitragen und jeder kann helfen. Aber man muss sich informieren und sich mit dem Projekt absprechen.“ Es hilft ungemein, einen guten Korrektor zu haben, der die Beiträge redigiert, auch weil das Dokumentieren am Ende eben doch nicht so einfach ist, wie es sich viele vorstellen. Auch Schraitle ist sich sicher: „Der Satz 'Aber schreiben kann doch jeder' stimmt halt so nicht. Trotzdem kann jeder zur Doku beitragen.“

Um das sicherzustellen, sollte sich ein Dokumentationsteam in einer größeren Firma offen aufstellen. Ohne Versionskontrollsystem wie Github geht da schon mal gar nichts mehr, meint Schraitle: „Welches System man wählt, ist weniger wichtig. Wichtig ist, dass es gemacht wird! Ohne das würde ich heute kein Projekt starten, egal ob Dokumentation

oder etwas anderes. Git/Github bietet sich an, da es weit verbreitet ist und die Option offenlässt öffentlich oder im Privaten zu arbeiten. Wenn sich ein großes Team mit Dokumentation beschäftigt, muss unbedingt ein CI-System wie Jenkins aufgesetzt werden. Das regelmäßige Bauen der Dokumentation sorgt auch dafür, dass Mitarbeiter mehr auf Qualität achten und Fehler frühzeitig entdeckt werden. Jenkins spuckt dann im Idealfall regelmäßig alle benötigten Output-Formate aus, vom PDF zu EPUB, HTML (Single File oder viele kleine Seiten) und alle anderen Formate.“

Beim Thema Dokumentationsformat gibt es mancherorts regelrechte Flame wars zwischen den Fans der einen oder anderen Lösung. Doch Schraitle weiß: „Ob du in Markdown, ReST, AsciiDoc, Docbook oder etwas anderem schreibst, ist weniger wichtig. Nutze das Format, das für dein Projekt am besten passt. Das Format legt auch die Tools fest und nicht umgekehrt. Kenne die Vor- und Nachteile jedes Formats, informiere dich vorher und sieh dir an, was deine Kollegen in der Entwicklung verwenden.“

Wenn all das geklärt ist, müssen die Verantwortlichkeiten entschieden werden: „Wer betreut ein Kapitel? Wer darf Änderungen einpflegen? Wer überprüft den Stil, Rechtschreibung und die technische Richtigkeit? Wer darf das Endpro-

dukt veröffentlichen? Je größer das Team wird, desto mehr muss man sich diesen Fragen stellen.“ Da kommen dann auch Übersetzungen ins Spiel: Gibt es nur eine Sprache? Oder soll das Dokument in diverse andere Sprachen übersetzt werden? Ist Letzteres der Fall, benötigt man weitere Tools, die Übersetzungen ermöglichen **5**. OpenSuse verwendet dafür beispielsweise Weblate [6](#). Dem stimmt auch seine Kollegin Julia Faltenbacher zu: „Wer bei der Lokalisierung ein paar Grundregeln beachtet, bekommt auch bessere Qualität in der Originalsprache. Wir haben dazu einige Abschnitte in unserem Styleguide.“ [7](#) Best Practices für Owncloud finden sich auf der Github-Seite des Projekts [8](#).

Viele Dokumentationsexperten betonen, dass die Werkzeuge und Toolchains immer nur Mittel zum Zweck sein sollten und nicht im Vordergrund stehen sollten. Suse nutzt DAPS [9](#), die Docbook Authoring and Publishing Suite **6**. In vielen Fällen kommt AsciiDoc zum Einsatz, weil das Markdown-Format niedrige Einstiegshürden stellt, oder DocbookXML, weil XML vielseitiger beim automatischen maschinellen Verarbeiten ist. Das wird umso wichtiger, je mehr automatisierte QA und Tests man einbindet: Styleguide, Rechtschreibung, Corporate Identity, sogar Anbindung an den OpenBuild-Service wären da denkbar. Auch das Suse Documentation Team muss da immer wieder nachforschen und verschiedene Frameworks vergleichen [10](#).

„Man kann durchaus im Kleinen anfangen, mit einem Etherpad oder anderen kollaborativen Tools.“ sagt Schraitle. Vor wenigen Jahren beschäftigte sich auch im Linux-Magazin ein Artikel mit Tools für kollaborative Dokumentations-Workflows [11](#). „Wie man dann die vielen Bausteine zusammenbaut, hängt von vielen Faktoren ab, nicht zuletzt vom Know-how, Budget und davon, was die Kollegen in der Entwicklung schon verwenden.“ Hauptsache, bestätigt auch Martin Mattel, ist doch, dass das Tool geeignet ist: „Das Framework, um die Doku zu schreiben, muss in der Lage sein, die Anforderungen zu bedienen, aber auch der Weiterentwicklung der Dokumentation standhalten können. Es muss auch zu künftigen Anforderungen gewachsen sein, Darstellungen wie Tabellen oder

Referenzierungen müssen einfach umsetzbar sein. Hier auf quick & dirty zu setzen, rächt sich und kostet beim Aufräumen viel Zeit und Geld.“

Von ChatGPT in der Dokumentation ist er nicht überzeugt: Die Kosteneinsparung sei zu gering, auch weil die KI-Tools viel Recherche im Nachgang erforderten. Viel wichtiger ist da, sagt auch Frank Sundermeyer, dass die gewählten Tools miteinander kooperieren können.

Der Umstieg von SVN auf Git vor bald zehn Jahren hat beispielsweise bei Suse zu einem fast exponentiellen Anstieg von konstruktiven Rückmeldungen und Beiträgen aus der Entwicklung geführt. Mit der Wahl eines Versionskontrollsystems scheiden allerdings binäre Formate wie die gängiger Office-Suiten definitiv aus.

Verständliche Sprache

Und nach all der Planung, Kommunikation und Bewusstseinsweiterung gibt es dann ja noch die sprachliche Ebene, die sich als nicht minder komplex erweist.



DAPS
DocBook Authoring and Publishing Suite

Documentation Forum Git Repository Issue Tracker

DAPS helps you author and publish documentation written in DocBook XML and AsciiDoc. DAPS handles everything from short articles by a single author to large documentation projects by many authors. It is a command-line tool and works on Linux.

New DAPS 3.1
Publish DocBook & AsciiDoc as HTML, PDF, EPUB
[Download](#)

News

2021-06-10: DAPS 3.2
DAPS 3.2 has been released! It includes a script for better validating DocBook tables, compatibility with xmlformat 1.9, and bug fixes. It removes several old, unused features ('online-docs', 'html --jsp', and SVN property reading for 'html --meta'). [Release Information](#).

2021-05-11: DAPS 3.1
DAPS 3.1 has been released! It includes much improved AsciiDoc support, extended validation options, and support for DITA images. [Release Information](#).

2018-11-08: DAPS 3.0
DAPS 3.0 has been released. In addition to DocBook, it also supports AsciiDoc! [Release Information](#).

6 Bei Suse kommt die Docbook Authoring and Publishing Suite DAPS zum Einsatz.

Gute Texte zu schreiben, ist eine Wissenschaft für sich. Chabowski rät, den Fokus auf Verständlichkeit zu legen. „Ein Tech

Writer schreibt keine Prosa, sondern eine Anleitung. Einfache Sprache, kurze Sätze, Wiederholungen von (Fach-)Begriffen,

LINUX

MAGAZIN

ONLINE

NEWSLETTER FÜR IT-PROFIS

Sie sind IT-Profi für Linux und Open Source? Bleiben Sie informiert mit dem werktäglichen Newsletter für IT-Profis vom Linux-Magazin!

Newsletter

LINUX
MAGAZIN

ONLINE

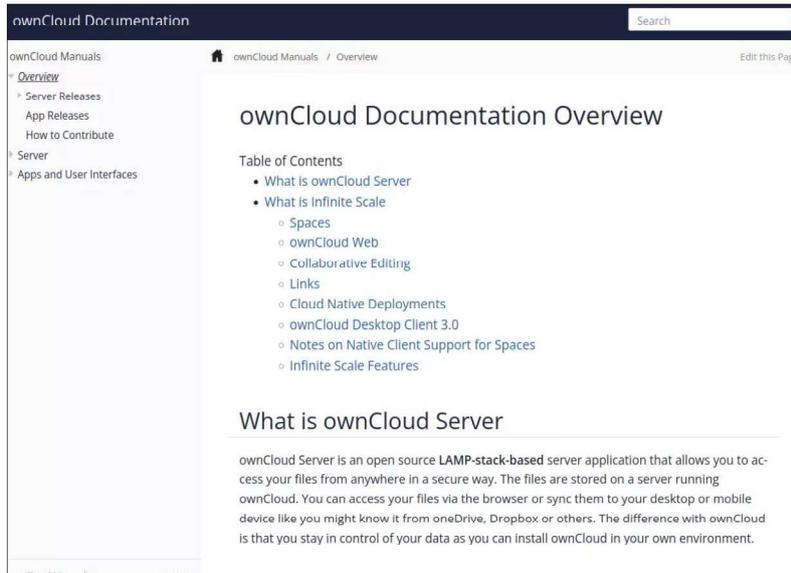
News

Stadt Dortmund prüft Einsatz freier Software und offener Standards
Die Stadt Dortmund hat das Projekt freie Software und offene Standards als Bestandteil ihres Masterplans für die digitale Stadtverwaltung aufgenommen. In den...

Mozilla veröffentlicht Internet Health Report
Health Report versucht die Mozilla-Stiftung, die Frage zu

- Tagesaktuelle IT-News
- Security-Infos des DFN-CERT
- Online-Stellenmarkt

Jetzt kostenfrei abonnieren! www.linux-magazin.de/subscribe



7 Egal, ob mit DocbookXML, AsciiDoc oder Markdown: Viel wichtiger als die Tools ist in jedem Fall das Ergebnis. Es muss klar, verständlich und ansprechend ausfallen. Die Owncloud-Dokumentation setzt zu diesem Zweck auf Antora und AsciiDoctor.

keine Ausschmückungen oder Füllwörter – das macht den Unterschied. Gute Dokumentation muss weder spannend noch abwechslungsreich sein, sondern auch bei komplexeren Themen nachvollziehbar und verständlich.“ Hier helfen Spell- und Stylechecker, Checklisten und automatische Qualitätssicherung, die das Einhalten von Regeln erzwingen.

„Unter Dokumentation erwarten Anwender heute aber auch viel mehr als nur geschriebene Texte. Ob die nächsten Generationen noch klassische Dokus lesen, ist durchaus nicht ausgemacht“, ist sich Chabowski sicher. Kurze Videos mit



Weitere Infos und interessante Links

www.lm-online.de/qr/49146

Der Autor

Markus Feilner nutzt seit 1994 Linux und Open Source. Er arbeitet als Consultant, Coach und Journalist, seit 2000 auch mit seiner Firma Feilner-IT, die sich auf die OSI-Layer 8, 9 und 10 konzentriert.

Schritt für Schritt vorexerzierten How-to-Anleitungen, ergänzende Podcasts, aufschlussreiche Grafiken – all das muss in die Dokumentation mit eingebaut werden, um den Konsum für die unterschiedlichsten Nutzergruppen zu ermöglichen. „Dafür braucht es wieder Planung, Zeit, entsprechende Werkzeuge und Experten. Dokumentationsfachleute dürfen Youtube, Tiktok und Instagram nicht unterschätzen“, mahnt Chabowski. Die Dokumentationsteams müssen hier eng mit dem Marketing zusammenarbeiten.

Aber auch die klassische Online-Hilfe ist eine notwendige Art der Dokumentation – eingebunden in die Software, ein Paradebeispiel für das, was auf Dokumentationsteams zukommt. Anwender erwarten keine dicken Wälzer, Manuals oder Guides mehr, so wie das Suse-Handbuch, das die Firma in den 1990ern groß gemacht hatte. Die Experten sind sich einig: Lösungs- und Themen-basierter Dokumentation gehört die Zukunft; agil erstellt und für fast beliebige Ausgabeformate strukturiert.

„Der Anwender will sich nicht durch ein umfangreiches Buch durcharbeiten, sondern sucht gezielt nach Informationen, die ihm dabei helfen, ein ganz bestimmtes Problem in den Griff zu bekom-

men oder eine bestimmte Softwareumgebung aufzubauen“, erklärt Chabowski. Das gelinge nur mit modularen Ansätzen in der Dokumentation: „Wer kann, sollte von vorneherein mit Bausteinen arbeiten. Softwaredokumentation wiederholt sich oft; viele Schritte, Prozeduren, Abläufe, die dort beschrieben sind, kann man wiederverwenden. Dafür ist ein Dokumentenmanagement-Tool hilfreich.“ Welches, das muss jedes Unternehmen selbst entscheiden, abhängig von der bereits verwendeten Toolchain.

Regeln und Best Practice

Bei den Interviews mit den Experten bemerkt man generell zwei Dinge: Erstens steht bei der Dokumentation keineswegs die Sprache oder die Grammatik im Vordergrund. Als wichtigste Komponenten für den Erfolg nennen die Doku-Profis Strukturen, Kommunikation und individuelle Fähigkeiten der Dokumentationsfachleute. Sie beschäftigt nicht die Frage, ob und wann man einen Fachbegriff übersetzt . Für sie spielen weder die im Deutschen und Englischen sehr unterschiedlichen Regelungen für Koppelungen eine Rolle noch die Ratschläge für die Maximallänge eines verständlichen Satzes. Vieles davon betrachten die Doku-Fachleute eher als Mythen, die Dritte über professionelle Dokumentationssteams im Kopf haben .

Moderne Dokumentationsteams – ganz besonders solche im Open-Source-Bereich – sind also keineswegs nur detailverliebte Sprachfanatiker. Sie arbeiten genauso agil wie die Entwickler, mit denen sie kooperieren. Hier kommt ein weiterer Punkt ins Spiel: Gute Dokumentationsteams nutzen agile Tools und Methoden und können sich mit (fast) jedem Entwicklungsmodell von Waterfall bis Scrum koordinieren.

Welche Tools dabei zum Einsatz kommen – ob ein Framework, eine Sprache, ein bestimmtes Format, eine Versionskontrolle oder gar ein einfacher Editor – das ist immer die allerletzte Frage, abhängig von den Umgebungsvariablen im Workflow der Entwickler und der Kunden. Wer seinen Workflow fest im Griff hat, der kann dann auch Input in jeder Form und in jedem Format annehmen – so wie DAPS bei Suse. (jcb) ■



Netbox: Mit Plugins nützliche Funktionen nachrüsten

Mit Überblick

Netbox hat sich als Werkzeug mit Zwitterfunktion für DCIM und IPAM längst etabliert. Wer jedes verbaute Teil in seinem Rechenzentrum dokumentieren will, findet in Netbox dafür die perfekte Lösung. Martin Loschwitz

Die **Dokumentation** des eigenen Netzwerks gehört üblicherweise nicht zu den Lieblingsaufgaben derer, die es bauen. Hilft aber nix: Wird der Administrator morgens um drei aus dem Bett geklingelt und sieht sich einem Problem im Netz gegenüber, muss er wissen, wer mit wem Daten austauscht, welcher Server an welcher Stelle verbaut ist, wie er an dessen BMC-Schnittstelle kommt und wo

er gegebenenfalls eine serielle Konsole anschließen muss.

Netbox **1** gehört zu den Werkzeugen, die Netzwerknern die Dokumentationsarbeit zumindest ein bisschen schmackhafter machen. Das Werkzeug entstand einst aus der Unzufriedenheit heraus, dass für Datacenter Inventory Management (DCIM) und IP Address Management (IPAM) zwar diverse Werkzeuge

existierten, die aber meistens nicht verzahnt waren. Logische Zusammenhänge zwischen einem System im Inventar und der IP-Adresse ließen sich so nicht darstellen. Wer in einer Firma arbeitet, in der noch die klassische Excel-Tabelle mit sämtlichen vergebenen IPs existiert, kennt die Mühsal, die diese Herangehensweise verursacht.

Netbox setzte dem Treiben vor etlichen Jahren ein Ende: Erstmals machte es eine Kombination aus DCIM und IPAM verfügbar, die unter einer freien Lizenz stand und nicht fest mit einem der großen Asset-Manager von HP und Konsorten verbunden war. Stattdessen wurde Netbox von Anfang an von Admins für Admins konzipiert. Unerreicht war seinerzeit beispielsweise die Detailtiefe, mit der sich der Inhalt von Racks erfassen ließ. Früh erkannten die Netbox-Entwickler, dass im modernen Rechenzentrum eben nicht mehr nur IP-Adressen und Netzwerke existieren: Darüber hinaus gilt es auch, Faktoren wie BGP-Verbindungen oder die Adressen virtueller Instanzen sinnvoll zu verwalten.

Heute präsentiert sich Netbox als Netzwerk-Tausendsassa und bildet all diese Faktoren und etliche weitere Parameter in seinen Daten ab. Strom- und Netzkabel lassen sich nicht nur einzeln benennen und inventarisieren, sondern auch exakt so beschreiben, dass man Verbindungen zwischen den einzelnen Servern und Switches sofort erkennt. Wo vielerorts heute noch immer mehr oder minder geraten wird, welches Kabel wohl wo entlangführt und an welchem Port es steckt, wissen Netbox-Nutzer im Detail Bescheid.

Mehrwert durch Plugins

Was viele Administratoren laufender Netbox-Instanzen allerdings nicht wissen: Netbox verfügt über eine Plugin-Schnittstelle, über die das Werkzeug sich verhältnismäßig leicht um weitere Funktionen erweitern lässt. Wie in der Open-Source-Welt üblich lieferte die Community prompt: Etliche freie Erweiterungen stehen zur Verfügung, mit denen das lokale Netz sich noch besser dokumentieren lässt.

Wer beispielsweise keine Lust hat, Netzwerkpläne zu zeichnen, der kann diese Aufgabe an ein Netbox-Plugin abtreten. Sofern die Daten in Netbox stimmen und die Verkabelung ordentlich erledigt und erfasst wurde, zaubert das Visualisierungs-Plugin für Netbox in wenigen Sekunden einen kompletten Netzplan. Der lässt sich dank mannigfaltiger Einstellungsoptionen beim Ausgabeformat auch an einen Plotter senden, der dann eine Netzwerktapete für die Bürowand druckt.

Zugegeben: Im Netz kursieren etliche Netbox-Erweiterungen, nicht alle leben auf Github. Die Entwickler von Netbox haben eigens für diesen Zweck eine Liste mit von ihnen für gut befundenen Plugins erstellt. Der folgende Artikel listet einige der besten Addons auf und zeigt, wie sie die Dokumentation des eigenen Netzwerks verbessern und manchmal sogar revolutionieren. Zuvor geht es aber kurz um das Setup von Netbox selbst.

Aller Anfang

Hinsichtlich des Ablaufs hat sich eine Netbox-Installation seit Jahren kaum ver-

The screenshot shows the Netbox web interface with a search bar at the top right and a user dropdown menu. The main content area is divided into several panels:

- Organization:** Sites (24), Tenants (11), Contacts (3).
- Circuits:** Providers (9), Circuits (29), Provider Networks (1).
- Virtualization:** Clusters (32), Virtual Machines (180).
- IPAM:** VRFs (6), Aggregates (4), Prefixes (90), IP Ranges (4), IP Addresses (180), VLANs (63).
- DCIM:** Sites (24), Racks (42), Device Types (14), Devices (72), Cables (108).
- Welcome!** A message about customizing the dashboard.
- NetBox News:** A list of recent releases including NetBox v3.4.8, v3.5, v3.4.7, and v3.4.6.
- Change Log:** A table showing recent system changes.

ID	Time	Username	Full Name	Action	Type	Object	Request ID
9	2023-04-19 21:39	admin		Created	Saved Filter	Test1	d6d99219-51c9-4b79-bba0-6f96fd077082
8	2023-04-19 21:24	admin		Updated	Front Port	Port 46	70f58add-0ee6-4796-b24b-6f3632b4d963

1 Netbox gilt im Kontext von DCIM und IPAM als Standardlösung, lässt sich mit den passenden Plugins aber um viele Funktionen erweitern.

ändert. Das liegt vor allem daran, dass die Netbox-Entwickler vor ein paar Jahren offensichtlich so gute Designarbeit geleistet haben, dass das Resultat bis heute trägt. Noch immer benötigt das Werkzeug weitgehend dieselben Komponenten wie zu Beginn seiner Entwicklung: Eine PostgreSQL-Datenbank, Redis, Unicorn als WSGI-Umgebung sowie Apache oder Nginx, um Netbox an den Client auszuliefern.

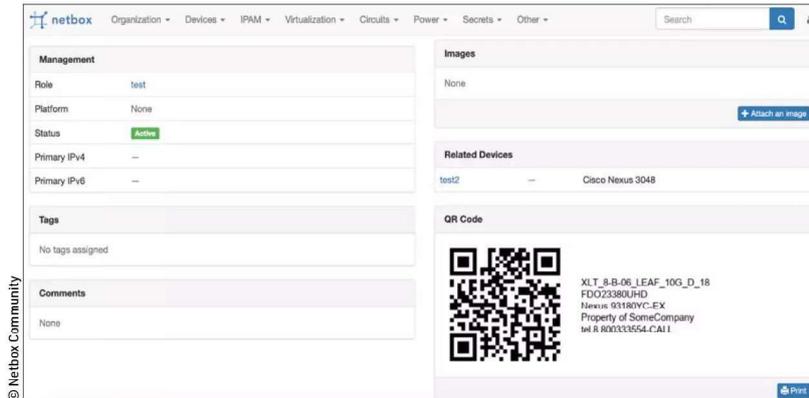
Um Netbox an den Start zu bringen, liefern die Entwickler zudem unverändert ein Setup-Werkzeug aus, das einen Teil der Installation übernimmt. Im Folgenden gehen wir davon aus, dass auf dem Zielsystem mit Netbox eine Redis-Instanz und PostgreSQL laufen; die Benutzerdaten beider Datenbanken müssen bekannt sein. Zudem muss es einen Systembenutzer *netbox* geben, mit dessen Rechten der HTTP-Dienst läuft und dem die zu Netbox gehörenden Dateien im System gehören. In `/opt/netbox/`, dem Home-Verzeichnis dieses Nutzers, landen auch die Dateien der Software.

Sie laden die jeweils aktuellste Netbox-Version [nach](#) `/opt/netbox/` herunter, ändern die Berechtigungen der Dateien auf den Systembenutzer *netbox* und ent-

packen danach als dieser den heruntergeladenen Tarball in `/opt/netbox/`. Alternativ dazu schlägt die Dokumentation vor, ihn nach `/opt` zu extrahieren und davon einen symbolischen Link auf `/opt/netbox/` anzulegen.

Dann geht es an die Netbox-Konfiguration. Mittels der Kommandos aus den Zeilen 2 und 3 von [Listing 1](#) übernehmen Sie die Beispielkonfiguration, in der Sie anschließend die Parameter `ALLOWED_HOSTS`, `DATABASE`, `REDIS` sowie `SECRET_KEY` anpassen. Ein Schlüssel für den letztgenannten Eintrag lässt sich mittels des Aufrufs `python3 ../generate_secret_key.py` generieren. Schließlich steht der Aufruf des Werkzeugs an, das einen anfänglichen Benutzer in Netbox anlegt ([Listing 1](#), Zeilen 4 bis 6). Danach gilt es noch, per Cronjob die Aufräumarbeiten von Netbox zu aktivieren (Zeile 7).

Anschließend kopieren Sie die Unicorn-Beispielkonfigurationsdatei `contrib/unicorn.py` im Netbox-Ordner nach `/opt/netbox/`. Über die Kommandos aus den Zeilen 9 und 10 bekommt Systemd Units, über die es Netbox steuert. Dann aktivieren Sie den Dienst (Zeile 11 und 12). Dazu muss freilich Unicorn bereits installiert sein; das Paket



© Netbox Community

2 Mittels des QR-Code-Plugins lassen sich Labels produzieren, die nach einem Scan per Smartphone auf die jeweilige Geräteseite in Netbox führen.

findet sich in den Repositories aller aktuellen Enterprise-Distributionen.

Im Anschluss erfolgt die Installation eines Webservers; im Beispiel kommt dafür Apache zum Zug. Sie sollten an dieser Stelle die Kombination aus SSL-Schlüssel und SSL-Zertifikat sowie ein eventuell benötigtes Intermediate-Zertifikat zur Hand haben, um Netbox von Anfang an adäquat abzusichern. Das Kommando aus Zeile 14 übernimmt im Anschluss die

Apache-Beispielkonfiguration aus Netbox. Darin verändern Sie dann die Pfade zu den SSL-Dateien, bevor Sie Netbox endgültig in Betrieb nehmen (Zeile 16ff).

Unter dem Host Namen des Systems, auf dem Netbox läuft, sollte sich nun der Port 443 von Netbox erreichen lassen. Die erste Anmeldung erfolgt mittels der zuvor auf der Shell festgelegten Benutzerdaten des Netbox-Superusers. Viel zu sehen gibt es nach dem ersten Login al-

Listing 1: Netbox-Konfiguration

```
01 ### Netbox konfigurieren
02 $ cd /opt/netbox/netbox/netbox/
03 $ sudo cp configuration_example.py configuration.py
04 $ source /opt/netbox/venv/bin/activate
05 $ cd /opt/netbox/netbox
06 $ python3 manage.py createsuperuser
07 $ sudo ln -s /opt/netbox/contrib/netbox-housekeeping.sh /etc/cron.
    daily/netbox-housekeeping
08 ### Systemd-Dienst
09 $ sudo cp -v /opt/netbox/contrib/*.service /etc/systemd/system/
10 $ sudo systemctl daemon-reload
11 $ sudo systemctl start netbox netbox-rq
12 $ sudo systemctl enable netbox netbox-rq
13 ### Apache-Konfiguration
14 $ sudo cp /opt/netbox/contrib/apache.conf /etc/apache2/
    sites-available/netbox.conf
15 ### Netbox in Betrieb nehmen
16 $ sudo a2enmod ssl proxy proxy_http headers rewrite
17 $ sudo a2ensite netbox
18 $ sudo systemctl restart apache2
```

lerdings noch nicht, schließlich fehlen sämtliche Inventar- und Netzwerkdaten.

Für Sie beginnt an dieser Stelle viel Fleißarbeit: Es kann einige Zeit in Anspruch nehmen, die komplette IT des eigenen Unternehmens in Netbox zu erfassen. Sie sollten hier allerdings keine Abkürzungen nehmen, bloß um Zeit zu sparen. Denn je genauer die in Netbox erfassten Daten ausfallen, desto besser ist die Dokumentation der eigenen Infrastruktur und umso erfolgreicher lassen sich die Plugins nutzen, die wir im Folgenden vorstellen.

Hardware beschriften

Während es früher üblich war, lediglich Geräte im Rechenzentrum zu beschriften, kennzeichnet man mittlerweile auch Kabel. Ein logischer Ansatz, da sich ein Kabel, das man anhand eines entsprechenden Etiketts an Ursprung und Ziel eindeutig identifizieren kann, relativ leicht austauschen lässt. Dazu genügt es, die jeweiligen Stecker auf beiden Seiten zu trennen, und schon lässt sich das Kabel aus dem Rack fischen. Hat man sich beim Bau des Racks entsprechend Mühe gegeben, gerät so auch ein eventuell vorhandenes Kabelmanagement nicht aus den Fugen.

Die verfügbaren Tagging-Lösungen der Hersteller sind allerdings nicht sonderlich digital. Sie können zwar ein Tag mit einem Namen produzieren, oft genug war es das aber auch schon. Für den Administrator vor Ort ist das nur bedingt hilfreich. Jeder, der schon einmal in einem Rechenzentrum gearbeitet hat, kennt das Problem wahrscheinlich: Rack-Bezeichnungen fallen noch intuitiv aus, bei einem sinnvollen Namensschema möglicherweise auch die Server-Namen. Informationen zu einem beliebigen Kabel im Rack lassen sich ohne Hilfe eines Computers allerdings nicht herausfinden. Wer vor einer Stromverteilereinheit (Power Distribution Unit, PDU) steht und sich darauf einloggen möchte, kann dagegen mit dem Host-Namen nur wenig anfangen, wenn er nicht zufällig die IP-Adresse des Management-Interfaces des Geräts im Kopf hat.

Mit dem entsprechenden Plugin bietet Netbox hier erweiterte Möglichkeiten: Beschriftungen lassen sich bis hinunter

auf einzelne Kabel oder virtuelle Instanzen anlegen. Doch wäre es nicht viel hilfreicher, könnte man vor dem Rack stehend alle Details sofort sehen, die in Netbox über die Hardware hinterlegt sind? Genau diese Möglichkeit bietet das Netbox-QR-Code-Plugin [2](#).

Möglich wird das, weil Netbox zum einen über eine grafische Oberfläche verfügt und zum anderen im Hintergrund eine API anbietet. Die ist so angelegt, dass sich jedes Gerät in Netbox über eine eigene URL eindeutig erreichen lässt. Das QR-Code-Plugin [2](#) vollbringt insofern gar kein Hexenwerk: Es generiert aus dieser URL einen QR-Code, den es mitsamt grundlegenden Details zum Gerät ausgibt. Den größten Teil der zu druckenden Felder bestimmen Sie, sodass das Label neben dem QR-Code sämtliche Details enthält, die Sie sich wünschen.

Noch praktischer: Ist das Plugin aktiv, blendet Netbox das Label samt QR-Code auf jeder Geräteseite ein und aktiviert zu sätzlich einen Print-Button. Wie es weitergeht, hängt von der lokalen Hardware ab, die für das Tagging zum Einsatz kommt. Realisieren Sie Ihr Tagging per Label-Printer, können Sie die Labels aus Netbox gleich ausdrucken und die Geräte und Kabel beschriften. So sind Sie für den nächsten Besuch im RZ perfekt vorbereitet. Stehen Sie vor der jeweiligen Hardware, genügt es, den QR-Code mit dem Smartphone zu scannen – flugs öffnet sich die passende Seite in Netbox.

Die Installation des Plugins ist denkbar einfach. Haben Sie Netbox wie beschrieben ausgerollt, lassen sich Erweiterungen ganz einfach mittels Pip in der Netbox-eigenen virtuellen Python-Umgebung installieren. Die beiden Befehle `source /opt/netbox/venv/bin/activate` sowie `pip install netbox-qrcode` sowie der Eintrag `PLUGINS = ['netbox_qrcode']` in `opt/netbox/netbox/netbox/configuration.py` schließen die Konfiguration ab. Die Änderungen greifen nach einem Neustart von Netbox. Die Geräte-QR-Codes tauchen nun von alleine auf, das Beschriften kann losgehen.

Damit das Plugin auch nach Netbox-Updates weiterhin zur Verfügung steht, empfiehlt es sich, in der Datei `~/netbox/local_requirements.txt` den Eintrag `netbox_qrcode` zu hinterlegen. Dann installiert Netbox das Plugin automatisch

Name	View	SOA Serial	Default TTL	Status
1.0.10.in-addr.arpa	—	1651013881	86400	Active
2.0.10.in-addr.arpa	—	1651013881	86400	Active
zone1.example.com	—	1651010898	86400	Active
zone2.example.com	—	1651010898	86400	Active
zone3.example.com	—	1651000650	86400	Active
zone4.example.com	—	1651000650	86400	Active
zone5.example.com	—	1651000650	86400	Active
zone6.example.com	—	1651000650	86400	Active
zone7.example.com	—	1651000650	86400	Active
zone8.example.com	—	1651000650	86400	Active

3 DNS-Einträge lassen sich mittels des DNS-Plugins verwalten, und zwar so, dass die Daten gleich auch beispielsweise in Ansible zum Einsatz kommen können.

erneut, wenn es seine virtuelle Python-Umgebung im Rahmen eines Updates frisch erstellt.

Alles klar im DNS

Ein weiteres Plugin beseitigt – wie Netbox selbst – eine Baustelle, die sich in vielen Firmen findet: die korrekte Dokumentation und Umsetzung von DNS-Einträgen. In vielen, wenn nicht sogar den meisten Unternehmen sind die Datensätze für Hardwareinfos und DNS strikt voneinander getrennt. Ein Admin schraubt also einen Server ins Rack, legt ihn in Netbox an und erstellt danach im DNS-Server den für die Maschine benötigten Eintrag. Das ist doppelt gemoppelt und der Administrator muss künftig die Daten zum jeweiligen Gerät an zwei Stellen pflegen. Verändert sich der Server durch Wegfall oder Austausch, gilt es, anschließend möglicherweise die A- wie auch die PTR-Einträge für das System zu bearbeiten.

Muss man logisch zueinander gehörende Daten unabhängig voneinander pflegen, kommt es bekanntlich regelmäßig zu Unstimmigkeiten. Entfernt etwa die Truppe, die im Rechenzentrum hantiert, ein Gerät aus einem Schrank und vergisst danach, die DNS-Einträge zu löschen, resultiert daraus mit einiger Wahrscheinlichkeit früher oder später Verwirrung. Begünstigt wird das durch den

Umstand, dass in vielen Firmen die Verantwortung für die Wartung und die für den Betrieb der physischen Infrastruktur bei unterschiedlichen Teams liegen.

Auch dieses Problem lässt sich mit Netbox lösen, genauer gesagt mit dessen DNS-Plugin [3](#). Es verwaltet zunächst Domänen und die DNS-Einträge für IP-Netzwerke inklusive A- und PTR-Einträgen. Diese löscht es zudem automatisch oder legt die entsprechenden Einträge an, falls sich an den Netbox-Daten etwas ändert. Quasi im Vorübergehen wird Netbox damit zur Single Source of Truth in Sachen DNS, was Arbeit spart und die Effizienz erhöht. Kommt das Plugin zum Einsatz, muss die RZ-Truppe nach dem Einbau eines Servers künftig kein Ticket mehr an Ops schreiben, um DNS-Einträge für das neue Gerät zu produzieren. Netbox erledigt das stattdessen ganz von allein.

Die Installation des Plugins [3](#) verläuft analog zu jener für das QR-Code-Plugin (Listing 2, erste zwei Zeilen). Der Eintrag aus Listing 3 aktiviert die Erweiterung und zeigt sogar die korrekte Syntax für mehrere Plugins gleichzeitig, denn diese lassen sich mit einer Zeile pro Instanz in der Datei angeben. Ein Echo-Befehl (Listing 2, Zeile 3) aktiviert das Plugin dauerhaft, ein kurzes Kommando im Netbox-Ordner (Zeile 4) fügt den Netbox-Tabellen die für DNS-Einträge benötigten

4 Daten zu BGP-Sitzungen sind eigentlich nicht vorgesehen. Das BGP-Plugin erweitert die Lösung darum.

Spalten hinzu. Danach taucht in Netbox der neue Menüpunkt *Zones* auf, über den Sie bei Bedarf DNS-Zonen samt aller Einträge anlegen.

BGP-Infos in Netbox

Das nächste Plugin rüstet eine Funktion nach, bei der man sich fragt, wieso Netbox sie nicht bereits nativ enthält: das Speichern von Informationen zu bestehenden BGP-Sessions der eigenen Router. Gut möglich, dass die Entwickler sich seinerzeit gedacht haben, entsprechende Informationen würden ohnehin im zentralen Konfigurationsmanagement und eben nicht in Netbox verwaltet. Seiner Aufgabe als Single Source of Truth im Netzkontext kann Netbox so aber nur eingeschränkt nachkommen. Aber es gibt eine Lösung dafür: Mittels des Plugins `bgp` lässt sich die entsprechende Funktionalität nachrüsten [4](#).

Bei aktiver Erweiterung lassen sich BGP-Communities in Netbox ebenso pflegen wie aufzubauende BGP-Sessions,

definierte BGP Routing Policies sowie die Listen der zu empfangenden und zu sendenden IP-Prefixes für einzelne Sitzungen. Obendrein lassen sich all diese Details an den passenden Stellen konkret mit einzelnen Geräten verbinden. Die Informationsvielfalt ist dabei enorm: Für einzelne Sessions können Sie in Netbox die meisten BGP-Eigenschaften ebenso anlegen wie die dazugehörige Verbindung mit IP-Netzwerken.

Die Installation des Plugins folgt den schon bekannten Regeln: Via Python-venv installieren Sie das Plugin `net-box-bgp`, aktivieren es in der Konfiguration und fügen der `local_requirements.txt` den Eintrag `netbox_bgp` hinzu. Danach erreichen Sie die BGP-Funktionen sowohl über den Punkt *Plugins* im Netbox-Hauptmenü als auch über entsprechende Endpunkte in der API.

Das BGP-Plugin ist zudem ein hervorragendes Beispiel dafür, dass Netbox sich mit anderen Diensten für den Einsatz als Single Source of Truth eignet. Würden Sie nur das BGP-Plugin aktivieren, müssten

Sie im weiteren Verlauf zumindest in der Theorie ja die Details zu den BGP-Verbindungen an zwei Orten pflegen, falls Sie Ihre Router aus der Automation heraus mit Konfiguration versorgen. Es gilt, was für Dokumentation immer gilt: Doppelt hält nicht besser, sondern ist großer Mist.

Wegen seiner vielseitigen API lässt Netbox sich aber selbstverständlich auch aus Automationslösungen heraus als umfassende Informationsquelle nutzen. In Ansible beispielsweise existiert eigens für diesen Zweck ein Lookup-Modul (`nb_lookup`), das spezifische Werte

aus Netbox auslesen kann. Bauen Sie Ihre Ansible-Information nun so, dass sie die Details über BGP-Verbindungen aus Netbox bezieht, statt aus den eigenen Variablen, ist die Brücke geschlagen: Änderungen in Netbox samt einem erneuten Ansible-Lauf führen dann automatisch dazu, dass der Automatisierer die aktualisierten Details auch auf den Ziel-Router übernimmt.

Netzwerkpläne zeichnen

Der letzte Kandidat in der Liste der extrem hilfreichen Netbox-Plugins ist `netbox-ui-plugin` [5](#). Das UI im Namen stammt von NextUI, und Netzwerkgurus wissen damit vermutlich bereits, wo die Reise hingehet: `netbox-ui-plugin` kann mit den in Netbox vorgefundenen Daten automatisch komplette Netzwerkpläne zeichnen. Wofür sich Netzwerk-Admins ansonsten die Finger wund klicken, benötigt Netbox mit dieser Erweiterung nur wenige Sekunden, um eine Hochglanztapete des Netzwerks für die Büro-

Listing 2: DNS-Plugin

```
01 $ source /opt/netbox/venv/bin/activate
02 $ pip install netbox-plugin-dns
03 $ echo netbox-plugin-dns >> ~/netbox/local_requirements.txt
04 $ ./manage.py migrate
```

Listing 3: DNS-Eintrag

```
<pre>
PLUGINS = [
    "netbox_dns",
] </pre>
```

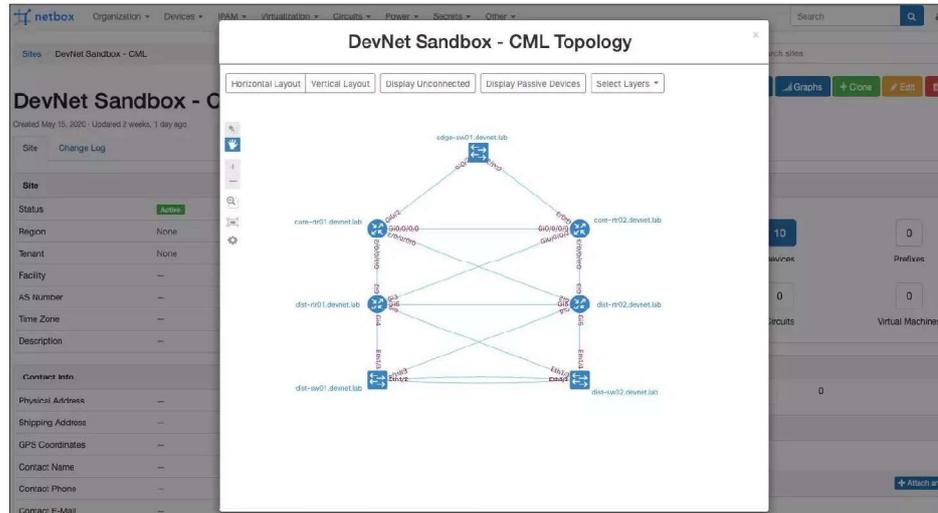
wand zu erstellen. Einen Haken hat die Sache allerdings: Damit Netbox Netzwerklane zeichnet, die die Realität tatsächlich abbildet, müssen die in Netbox hinterlegten Daten möglichst genau und richtig sein. Das umfasst Kabel, Netzwerke und die Hardware selbst. Haben Sie hier anfangs nicht exakt genug gearbeitet, rächt sich das spätestens beim Zeichnen der Netzwerklane, und es fällt Nacharbeit an.

Die Installation der Erweiterung unterscheidet sich von jener der anderen Plugins, jedoch nur in einem kleinen Detail: Das UI-Plugin  ist nicht per Pip zu beziehen. Stattdessen holen die Befehle aus Listing 4 das Addon auf das System. Als Nächstes steht der Eintrag `nextbox-ui-plugin` in der `local_requirements.txt` an. Der passende Eintrag zum Laden des Plugins in der Netbox-Konfiguration lautet dann `netbox-ui-plugin`.

Das Plugin hat eine ganze Reihe von Konfigurationsmöglichkeiten, die die Github-Seite umfassend dokumentiert. Dabei spielt der Parameter `layers_sort_order` eine ganz besondere Rolle: Dieser Parameter bestimmt die Reihenfolge der dargestellten Geräte in der fertigen Abbildung. Danach stehen nur noch ein paar vorbereitende Aufgaben an: In der Python-venv-Umgebung richten die beiden letzten Befehle aus Listing 4 das Plugin auf der Netbox-Seite endgültig ein. Ein Neustart schaltet die Konfiguration am Ende wie gewohnt scharf.

Listing 4: Netzplan-Plugin

```
$ git clone https://github.com/iDebugAll/nextbox-ui-plugin
$ cd nextbox-ui-plugin
$ source /opt/netbox/venv/bin/activate
$ pip3 install .
[...]
$ python3 manage.py collectstatic
$ python3 manage.py migrate
```



5 Das UI-Plugin zeichnet aus den in Netbox vorgehaltenen Daten Netzpläne. Das klappt freilich nicht ohne einen möglichst genauen, gut gepflegten Netbox-Datensatz.

Satte Auswahl

Die hier gezeigten Plugins stellen nur eine kleine Auswahl der sinnvollen Erweiterungen dar.

Liebäugeln Sie mit noch mehr Addons, dann sehen Sie sich am besten auf der Website von Awesome Netbox  um: Dort findet sich eine kuratierte Liste der Netbox-Entwickler mit vielen nützlichen Plugins. Dazu zählen unter anderem eine Telefonbuchverwaltung, eine Anbindung an OIDC, ein IP-Rechner für Netzwerke, Schnittstellen zu RouterOS sowie ein komplettes Lifecycle-Management, in dem sich Details zum Support einzelner Geräte hinterlegen lassen.

Fazit

Netbox lässt sich bereits allein vielseitig einsetzen, manche Plugins erhöhen die

Nützlichkeit der Lösung aber nochmals erheblich. Längst hat es sich im modernen Rechenzentrum eingebürgert, Wissen möglichst zuverlässig und zentral vorzuhalten. Über entsprechende Erweiterungen kann Netbox diese Rolle umfassend erfüllen. Es macht dann separat gepflegte BGP- und DNS-Daten ebenso überflüssig wie mit Visio & Co. gezeichnete Netzwerklane.

Obendrein bietet Netbox eine ganze Reihe von Plugins für Statusdienste: Möchten Sie etwa die in Netbox hinterlegten Geräte automatisch überwachen, liefert der Prometheus-Exporter sämtliche Nutzdaten in einem Format, die das Monitoring-System nativ verwenden kann. Zumindest für Netzwerkhardware wird Netbox damit zusätzlich zur maßgeblichen Konfigurationsquelle für das Monitoring. Kurz gesagt: Wer Netbox klug einsetzt, der erschlägt damit viele Probleme auf einen Streich. (jcb/jlu) ■

Dateien zum Artikel
herunterladen unter

www.lm-online.de/dl/49236



Weitere Infos und
interessante Links

www.lm-online.de/qr/49236