

UpTimes

Mitgliederzeitschrift der
German Unix User Group

Industrie 4.0

Spaß mit Named Pipes

IT-Management

guug

2016-1

Inhaltsverzeichnis

Sehr geehrten guten Tag: Gruß aus der Redaktion <i>von Anika Kehrer</i>	3
/etc: Neues aus dem Verein <i>von Nils Magnus</i>	6
Ein Augenzeuge berichtet: GUUG-Frühjahrsfachgespräch 2016 <i>von Lenz Grimmer</i>	10
Erklimmen eines Babylonischen Turms: Industrie 4.0 und das Internet der Dinge <i>von Hartmut Streppel</i>	14
„Kleine Schritte, um anzufangen“: Aus der Praxis eines IT-Doku-Dienstleisters <i>Interview: Anika Kehrer</i>	24
Was lange währt ...: Markup für Entwicklungs-Dokumentation <i>von Patrick Koetter</i>	29
Einfacher als man denkt: Über das Schreiben von Fachtexten <i>von Jürgen Plate und Anika Kehrer</i>	34
Shellskripte mit Aha-Effekt VI: Named Pipe – das unbekannte Wesen <i>von Jürgen Plate</i>	40
Das hat doch alles einen Sinn: Musings on IT-Management <i>von Frank Benke</i>	43
Hilfreiches für alle Beteiligten: Autorenrichtlinien	50
Über die GUUG: German Unix User Group e.V.	53
Impressum	54

Sehr geehrten guten Tag Gruß aus der Redaktion

Ironie an: Dokumentation ist überbewertet; Ironie aus. Das Fatale ist: Wenn man sie braucht, ist es schon zu spät. Wissensvermittlung in verschiedensten Formen – davon handelt, könnte man sagen, auf verschiedenen Ebenen diese Ausgabe der *UpTimes*.

von Anika Kehrer

Was ich nicht weiß,
macht mich eines Tages ‚Ups‘ sagen.

Frei nach dem deutschen Sprichwort unbekannter Herkunft

Ihr habt das Inhaltsverzeichnis nicht etwa überblättert, oder? Frevler! Ein Blick dort hinein ist ein Blick in den Kosmos. Den Redaktions-Kosmos. Okay, okay, das ist vollkommen pathetisch. Die Sache ist nur die: Was sich hinter den Kulissen abspielt, ist der eigentliche Schatz eines Ereignisses.

Mit dieser philosophischen Behauptung als großartigem Schwenk zum Thema *Backstage* streue ich nun als kleinen Kontrast übergangslos Salz in die Wunde namens Dokumentation: Ein Thema, das seit der letzten *UpTimes* Dich, lieben Lesenden, gehäuft nervt, ganz so wie im wirklichen Leben. Und jetzt kommt's: Auch in der Redaktion fällt uns das auf die Füße. Nein, es ist noch nicht so, dass es großartig wehgetan hat. Ganz wie im wirklichen Leben. Aber die Nachtigall hört man trapsen. Wer es überhört, den blendet in absehbarer Zeit schmerzhaft das Licht des anbrechenden Tages – und Tageslicht ist bekanntlich eine Analogie für das Licht der Wahrheit.

Wie im wirklichen Leben

Im letzten Intro der Winter-Ausgabe 2015-2 war davon die Rede, dass wir die Entstehung der *UpTimes* als Redaktionsprozess eines verteilten Teams partizipativ gestalten. Zum gemeinsamen Tun von etwas gehört wissen, wie es geht. Im Augenblick geschieht das Eintüten eines Artikels mehrheitlich zentralistisch: Die Verfasserin dieser Zeilen übernimmt es. Zum Eintüten eines *UpTimes*-Artikels gehört – natürlich neben der inhaltlichen Arbeit – das Fertigmachen für den Satz, den wir bekanntlich mit $\LaTeX$ vornehmen. Das mit dem zentralistisch ist nicht unbedingt freiwillig, aber im Moment ist es eben so.

Nun ist es so, dass das für die *UpTimes* benötigte $\LaTeX$ -Markup Eigenheiten besitzt, die zu dem speziellen Aussehen der *UpTimes* führen. Das ist

übrigens der Grund, warum die Autorenrichtlinien dem interessierten Autoren ans Herz legen, die redaktionseigene Vorlage zu benutzen – oder gleich ganz auf die Formatierung zu verzichten, und sich auf den Text zu konzentrieren. Die Gestaltung des Textes und sein Inhalt sind nämlich zwei Paar Schuhe.

Im internen Redaktionsprozess wiederum gehört das so genannte Satzfertigmachen jedoch zum Job, und zwar egal, ob wir von der *UpTimes* sprechen oder von der *c't*. Nun muss ein Redakteur aber wissen, wie die hauseigenen Tags lauten, damit er sie anbringen kann. Und? Ist es dokumentiert? Nun ja!, lautet die Antwort. Räusper, lautet die Antwort. Hüstel, hüstel.

Langer Rede kurzer Sinn: Fehlende Dokumentation fällt erst dann auf, wenn jemand nach ihr fragt. Und es fragt erst dann jemand nach ihr, wenn dieser Jemand etwas tun möchte, was er bislang nicht getan hat. An der *UpTimes* mitwirken, zum Beispiel, indem er einen Artikel satzfertig macht. Oder, das lässt sich beliebig verallgemeinern, in seinem Unternehmen neue Tasks übernehmen. Wenn in einem Team mehr Leute etwas tun, was bisher nur einer getan hat, dann kommt es zu Wachstumsschmerzen.

Auf der anderen Seite: Wachstumsschmerz impliziert Wachstum. So klein das Wachstum der *UpTimes*-Redaktion auch sein mag – es ist toll. (Naja, bis auf die dann anfallenden Doku-Tasks. Wie im richtigen Leben.) Und es ist unverzichtbar. Die *UpTimes* entsteht mit gehörigem Aufwand, und ohne Team funktioniert das nicht.

Hast Du den Eindruck, einen versteckten Appell wahrzunehmen? Dein Eindruck trügt nicht. :-) Wer Lust darauf hat, nach wohlüberlegten Grundregeln ein tolles Heft zu machen, der braucht nicht zu zögern. Er wendet sich an [<redaktion@uptimes.de>](mailto:redaktion@uptimes.de).



Die Geschichte hinter dem Titelbild *Spanabhebende Systemadministration*

Ihr kennt es alle: Das alte Gerät ist über Kopfhöhe im Rack eingebaut, und da soll es jetzt raus. Eine der vier Schrauben hat sich festgefressen. Beim Käfig der Käfigmutter sind inzwischen die Nasen abgebrochen, deswegen dreht sich die Käfigmutter munter mit. Nach verschiedenen Versuchen der Art *einer versucht, mit der Zange dagegen zu halten, einer dreht mit dem Schraubenzieher* kommen die zwei Senior-Sysadmins zum Schluss: Da hilft nur noch *Spanabhebende Systemadministration* – hier in Form der Eisensäge meines treuen SwissTools.

Fotografiert mit einer Olympus Pen-F, Objektiv Yashica ML 50mm 1:1.9 mit 16mm Makro-Zwischenring. (Hella Breitkopf)

Geschichten hinter Geschichten

Die Aufgaben der Redaktion sind so vielfältig wie anspruchsvoll. Aber das klingt schlimmer, als es ist. Wen das Heftmachen interessiert, der bringt die Voraussetzungen sowieso bereits mit. Man kann zum Beispiel einen Blick dafür entwickeln, welche Inhalte interessant sind. So entstand in dieser Ausgabe der mutmachende Artikel *Das hat doch alles einen Sinn: Musings on IT-Management* von Frank Benke: Ein UpTimes-Redakteur hat ihn im Blog des Autoren gefunden.

Andere Artikel entstehen von vornherein schon durch Zusammenarbeit von Redakteur und Autor. Manchmal sind Nachfragen durch den Redakteur nötig, ob ein Autor zu einem Thema etwas schreiben möchte, und manchmal hat ein Autor schon ein Thema, das er gern bearbeiten möchte – so wie bei dem hochinteressanten Artikel *Erklimmen eines Babylonischen Turms: Industrie 4.0 und das Internet der Dinge* von Hartmut Streppel.

Als Redakteur arbeitest du in einem solchen Fall der Neu-Entstehung genau dann an einem guten Ergebnis mit, wenn du dem Autoren Feedback gibst, an welchen Stellen vielleicht noch etwas fehlt, damit das Ergebnis rund wird. Denn die Eigenwahrnehmung des Textes entspricht nicht unbedingt der Fremdwahrnehmung desselben Textes, zumal bei einem komplexen Thema. Wenn am Ende der Autor seinen Artikel mag, obwohl du als Redakteur in Vertretung des späteren Lesers darin herumgewühlt hast, dann weißt du, dass du alles richtig gemacht hast.

So eine eigene kleine Geschichte ließe sich zu jedem Eintrag im Inhaltsverzeichnis erzählen. Eine Manuela Reiss kommt nicht auf dich zu und bietet ein Interview an – genauso wenig wie jeder andere Experte. Wenn du Fragen an ihn hast, dann leitest du so ein Interview selbst in die Wege. Und so kleine Juwelen wie Jürgen Plates *Shellskripte mit Aha-Effekt* fallen dir auch nicht einfach in die Hand: Nachhaltige Zusammenarbeit und immer wieder Feinarbeit am Gelieferten ist hierfür nötig. Und auch das macht die Tätigkeit eines Redakteurs aus: Eigene Artikelprojekte übernehmen,

mitsamt Rückfragen an den Autoren und Einfrieren in die Redaktionsvorlage.

Wissen bedeutet tun können

Intern eine Linie finden, inwieweit etwas in einem Artikel erklärt werden muss, und dann nach außen hin verbindlich und höflich auftreten, gehört zu den komplexeren Anforderungen, die sich an ein mehrköpfiges Team stellen. Wenn als Unfall einmal interne Diskussions-Mails aus Versehen an die Adresse des Autoren gehen, dann wird klar, dass einiges hinter dem steckt, wie ein Artikel entsteht – das kann sogar für Schluckauf sorgen, zum Beispiel beim Autoren, aber auch im Team. Denn in der internen Auseinandersetzung mit einem Text herrscht unter Umständen ein anderer Ton: ein Redaktionston. Das bedeutet aber auch: Identität, bisweilen sogar Passion. Das ist nicht selbstverständlich. Auch und gerade in einer so kleinen, kostbaren Einheit wie der UpTimes-Redaktion.

Und wozu das alles? Als Sinnspruch für den motivierenden Artikel *Einfacher als man denkt: Über das Schreiben von Fachtexten* wollten wir als erste Wahl ein Zitat nehmen, das wir aufgrund ungeklärter Nutzungserlaubnis dann aber nicht nehmen konnten. (Warum ich es dort nicht, aber hier wohl ohne Nutzungserlaubnis nehmen kann, ist ein interessantes Detail des Publikationswesens.) Es stammt von Felix von Leitner (aus *Fefes Blog*) und lautet, etwas gekürzt:

Wenn es jemanden interessiert, ist
[Wissen] automatisch nicht nutzlos.

Diese Aussage passt nicht nur als Argument des genannten Artikels, sein Wissen als Fachtext zu verfassen. Diese Aussage beschreibt gewissermaßen die Content-Philosophie der UpTimes. Solches Wissen, das ein GUUG-Mitglied interessiert (okay, einzige Einschränkung ist ein gewisser Zusammenhang mit offenen technischen Systemen), solches Wissen also ordentlich zu verpacken, das vereinfacht seine Aufnahme durch den

Lesenden. Und wissen bedeutet tun können, egal ob beim Satzfertigmachen eines UpTimes-Artikels oder bei seinem Ergebnis, um zum Beispiel mit Hilfe des UpTimes-Artikels *Was lange währt: Mar-*

kup für Entwicklungs-Dokumentation Probleme zu lösen.

Ich wünsche viel Freude beim Anwenden neuen Wissens mit der Sommer-UpTimes 2016.

Über Anika



Anika Kehrer arbeitet seit rund zehn Jahren als Technikjournalistin und ist seit der Wiederauflage der UpTimes im Sommer 2012 von der GUUG als Chefredakteurin beauftragt. Ihre Arbeit ist auszugsweise auf der Webseite <https://www.torial.com/anika.kehrer> ersichtlich. Ihr Ziel für die UpTimes ist, sie als Vereinszeitschrift der GUUG in Zusammenarbeit mit GUUG-Mitgliedern als hochwertiges, transparentes und partizipatives Fachmagazin zu gestalten.

/etc**Neues aus dem Verein**

Die *GUUG* lebt von den Aktivitäten und Plänen ihrer Mitglieder. An dieser Stelle berichten wir von Neuigkeiten aus allen Bereichen des Vereins. Diesmal geht es um eigene und andere Veranstaltungen, die *UpTimes*, den *LinuxTag*-Vereinsmerger, Techniktreffen und eine Zukunftswerkstatt.

von Nils Magnus

Reden ist Silber,
handeln ist Gold,
gemeinsam handeln ist Dynamit.

Frei nach dem bekannten Sprichwort unbekannter Herkunft.

Mitgliederversammlung und FFG

Knapp 150 Teilnehmer hatten sich zu den zwei Tracks des FFG in Köln versammelt: ein positiv stabilisierender Trend für die Hausveranstaltung der GUUG. Die Slides zu Admin-Themen, Storage, Configuration Management, Dokumentation und einer Keynote von Johann-Peter Hartmann zur Stellung von (Unix-) IT in unserer Zeit sind zum Download hinterlegt auf der Veranstaltungswebseite <https://www.guug.de/veranstaltungen/ffg2016/slides/>.

An der MV beteiligten sich rund 35 Mitglieder. Der Vorstand berichtete von den Fortschritten in der Mitgliederverwaltung, den FFGs, Kommunikationsmaßnahmen und den Haushaltsansatz 2016, in dem explizites Budget für weitere Veranstaltungen eingeplant ist. Das vorläufige Protokoll findet sich auf der GUUG-Wikiseite <https://wiki.guug.de/orga/mv/2016-protokoll>.

OpenPGP.conf im September

Die GUUG veranstaltet auf Anregung von GUUG-Mitglied und *GnuPG*-Initiator Werner Koch am 8. und 9. September 2016 in Köln die erste *OpenPGP.conf*. Dazu hat der Verein Martin Schulte mit der Durchführung beauftragt. Die Konferenz befasst sich mit Protokollfragen, Anwendungen, kryptographischen Fragestellungen und Einsatzszenarien rund um den offenen Standard, sei es in seiner Implementation *GnuPG* oder in anderen Fassungen. Der Call for Papers läuft noch bis Ende Juni 2016 auf der Webseite <https://gnupg.org/conf/>

Event-Tipps für Unix-Interessierte

Neben den selbst organisierten und veranstalteten Events der GUUG finden in den kommenden Mo-

naten auch wieder eine Reihe weiterer Veranstaltungen von oder mit GUUG-Mitglieder statt:

GUUG-Mitglied *B1 Systems* veranstaltet am 21. und 22. Juni 2016 in Köln die *Deutschen OpenStack-Tage*. Neben Usecases von Unternehmen, die OpenStack praktisch einsetzen, erfahren Besucher technische Details der Cloud-Plattform sowie benachbarter Themen wie *Ceph* und *Docker*. Ausrichter *B1 Systems* gewährt GUUG-Mitgliedern mit dem Code „nl_guug“ zehn Prozent Rabatt auf den Eintrittspreis: <https://openstack-tage.de/programm/>

Vom 22. bis 26. Juni 2016 trifft sich die Chamäleon-Community zur *OpenSUSE Conference* in Nürnberg. Kernel, Konfigurationen und Cloud sind nur einige der Themen der Veranstaltung, die die GUUG dieses Jahr unterstützt: <https://events.opensuse.org/conference/oSC16/>

Die *GUADEC*, Hauskonferenz der *GNOME*-Entwickler, macht in diesem Jahr vom 12. bis 14. August 2016 in Deutschland Stopp, und zwar in Karlsruhe. Nicht nur Desktop-only-Themen, sondern auch gegenwärtig heiß diskutierte Fragen wie *xdg-app*, das nun *Flatpak* heißt, stehen auf dem Programm: <https://2016.guadec.org/>

Zum elften Mal öffnen die Tore der Hochschule Bonn-Rhein-Sieg am 20. und 21. August 2016 zur *FrOSCon*. Schwerpunkte umfassen das allgegenwärtige IoT, Kartenanwendungen und die Beziehung von Wissenschaft und Open Source: <https://froscon.de/>

Stiefel und Maßkrüge sind keine Voraussetzungen zur Teilnahme an der *Linuxbierwanderung*, die bereits seit 1999 jeweils im Sommer an wechselnden europäischen Orten mit entsprechend wechselnden lokalen Orga-Teams stattfindet. In diesem Jahr treffen sich Linux-Anwender und -Admins vom 30. Juli bis zum 7. August 2016 auf der britischen Insel Isle of Man. Auch Fußfaule, OpenBSD-Präferierende und Weinliebhaber

finden ihren Spaß und die nötige Erholung bei der LBW. Im Jahr 2016 organisiert sie sich auf der Webseite: <http://lbw.crye.me.uk/>

Viele GUUG-Mitglieder sind bei der Planung der *CommitterConf* 2016 am 17. und 18. Oktober 2016 im Essener Veranstaltungszentrum *Unperfekthaus* beteiligt. Das Besondere ist die dynamische Programmplanung: Wer früh bucht, kann aktiv mitbestimmen, welche Vorträge wie prominent ins Programm kommen. Bislang stehen LPIC-Themen und Kommandozeilen-Handwerkszeug auf dem Programm, aber weitere Themen lassen sich noch einreichen: <https://committerconf.de/2016/>

Wer weitere empfehlenswerte Veranstaltungen kennt, kann sich gern an die Aktiven oder den Vorstand wenden. Darüber hinaus können sich Veranstalter innovativer neuer Events um eine Anschubunterstützung beim Vorstand bewerben. Siehe dazu das Sponsoring-Konzept der GUUG: <https://guug.de/verein/sponsoring.html>

Call for Proposals: FFG und Linux-Kongress

Das *Frühjahrsfachgespräch* (FFG) ist die GUUG-Hauskonferenz und damit praktisch untrennbar mit dem Verein verbunden. Der *Linux-Kongress* war seinerzeit die erste überregionale Linux-Veranstaltung in Deutschland. Damit beide im kommenden Jahr wieder stattfinden können, werden Organisationsteams gesucht.

Da solche Events zu veranstalten zwar lösbar ist, aber auch eine Herausforderung darstellt, wünscht sich der Verein am liebsten viele Freiwillige, die sich die Arbeit aufteilen. Alternativ akzeptiert die GUUG aber auch Durchführungskonzepte, die manche Aufgaben finanziell kompensieren. Wer hier Verantwortung übernehmen möchte, kann sich für Rückfragen an die Aktivenliste <aktive@guug.de> wenden und ein knapp einseitiges Grobkonzept inklusive Budgetplan einreichen.

Vereinsmerger: LinuxTag will GUUG beitreten

Auf der Mitgliederversammlung wurde es bereits angekündigt: Der *LinuxTag e.V.* hat angefragt, sich mit der GUUG zu vereinigen. Dazu treten die Mitglieder des langjährigen Ausrichters der gleichnamigen Veranstaltung der GUUG bei und bringen ihr Vereinsvermögen sowie die Marken- und Domainrechte mit. Für die GUUG und ihre Mitglie-

der ändert sich sonst nichts, einzig die IT und einige seiner Dienste möchte der *LinuxTag e.V.* weiterbetrieben sehen. Fragen zum Merger werden auf der Aktivenliste <aktive@guug.de> diskutiert.

Techniktreffen: IT- und Web-Teams planen Workshops

Die GUUG ist ein Technikverein, sodass es selbstverständlich erscheint, dass wir unsere Server und Dienste selbst betreiben. Bisweilen ist jedoch eine Generalüberholung notwendig. Deshalb hat der Vorstand Budgets für Technik-Workshops jeweils zu IT und Web bereitgestellt, auf denen richtungsweisende Entscheidungen zum weiteren Betrieb getroffen und umgesetzt werden.

Wer also mit einer gewissen Verbindlichkeit Verantwortung für unsere Server, einen Dienst oder das CMS übernehmen möchte, den lädt der Vorstand zu einem Admin-Wochenende ein. Interessenten wenden sich auch hier an die Aktiven der Liste <aktive@guug.de>, um Termine und Inhalte abzustimmen.

Umfrage: UpTimes optional auf Papier

Die Mitgliederversammlung hat den Wunsch geäußert, das Interesse herauszufinden, ob zusätzlich und ohne Kosten für die Mitglieder die Mitgliederzeitschrift *UpTimes* wieder auf Papier zu drucken, um sie beispielsweise auf Konferenzen auszulegen. Ferner sollen diejenigen Mitglieder ein gedrucktes Exemplar zugesandt bekommen, die explizit danach fragen. Um ein Stimmungsbild abzugeben, kreuzen Mitglieder bis zum 27. Juni 2016 bitte hier unverbindlich ihre Präferenz an: <https://goo.gl/JJra1t>

Eine Reihe Mitglieder haben bereits Ihre Meinung kundgetan: Es sind in der Umfrage auch Kommentare und weitergehende Vorschläge möglich. Das Ergebnis wird in einer der kommenden Mitgliederinformationen bekanntgegeben. Wie aus der URL ersichtlich, hat das Mitglied, das die Umfrage betreut, diese bei *Google Forms* gehostet. Für die Teilnahme werden allerdings keine personenbezogenen Daten erhoben.

Umfrage zur IT-Dokumentation

Auf dem FFG gab es bereits erste Ergebnisse, aber die Umfrage zur IT-Dokumentation nimmt weiter Erfahrungen von System Engineers und Administratoren zum Thema *Betriebliche IT-Dokumentation*

entgegen. Der Link darf auch an Kollegen und Freunde geraten: <http://goo.gl/forms/9E6ev2ICVi>

Mitmachen und GUUG gestalten: Zukunftswerkstatt

Zwar sehr grundsätzlich, aber gar nicht akademisch ist die fundamentale Frage: „Wo wollen wir eigentlich hin mit unserer GUUG?“ Soll der Verein mehr Service-Charakter haben, der verlässlich ein FFG organisiert, soll er mehr als technische Interessenvertretung agieren, oder soll er einfach nur als Markt- und Spielplatz für Ideen und technische Lösungen dienen? Mitglieder hatten die Idee geäußert, solche Fragen im Rahmen einer Zukunftswerkstatt zu erörtern. Vielleicht erwächst daraus ein Bericht oder Vorschlag für eine zukünftige Mitgliederversammlung? Wer sich hier beteiligen möchte, wendet sich an Felix Pfefferkorn, am besten über die Aktivenliste <aktive@guug.de>.

Reiserichtlinie beschlossen

Der Vorstand hat eine einheitliche Reisekostenrichtlinie beschlossen, die für alle gilt, die im Auftrag des Vereines reisen. Er legt den schonenden Umgang mit den Vereinsmitteln bei Reisen und Übernachtungen transparent fest.

Beiträge und Logo-Verwendung

Gute Nachricht für alle persönlichen Mitglieder: Entsprechend dem Beschluss der Mitgliederversammlung im März dieses Jahres sinkt deren Vereinsbeitrag um 20 Euro auf 70 Euro Jahresbeitrag. Der Beitrag für 2016 wird in den kommenden Wochen in Rechnung gestellt. Es hilft übrigens sehr, wenn Mitglieder dem Verein ein SEPA-Mandat erteilen. Wer den Verein damit unterstützen möchte, wendet sich bitte formlos an <kontakt@guug.de>.

Mitglieder können übrigens gern das GUUG-Logo verwenden, um etwa auf Briefköpfen oder Webseiten auf die Mitgliedschaft hinzuweisen, sofern die Verwendung den Text *Mitglied der GUUG* führt. Das Logo muss dabei in Farbe und Gestaltung unverändert bleiben. Das Nutzungsrecht steht nur GUUG-Mitgliedern zu und ist nicht übertragbar. Die Materialien finden sich in unserem Wiki unter <https://wiki.guug.de/offen/guug-logo/index>

Mailinglisten in der GUUG

Bei rund 500 Mitgliedern müssen wir die E-Mail-Kommunikation etwas strukturieren, sodass jedes Mitglied die Informationen bekommt, die es benötigt. Weil es bisweilen etwas Verwirrung über die richtigen Adressen gibt, hier eine Übersicht der aktuellen Listen:

- (1) Auf **mitglieder@guug.de** ist jedes Mitglied fest eingetragen. Wir nutzen die Liste nur für vereinsrelevante Ankündigungen (Wahlen, Versammlungen, Newsletter), weshalb sie moderiert ist und wenig Traffic aufweist. Haben Mitglieder Wichtiges anzukündigen, können Sie sich an den Vorstand wenden.
- (2) Auf **guug-members@guug.de** lässt sich diskutieren, was sich aus aktuellen Themen oder Mitteilungen ergibt. Der Listenname mutet etwas seltsam an – er ist historisch bedingt. Jedes Mitglied darf dort posten, sofern es die Liste abonniert hat. Das trifft gegenwärtig auf etwa 250 Mitglieder zu. In den vergangenen Jahren hat es der Verein teilweise etwas unterschiedlich gehandhabt, ob neue Mitglieder auf dieser Liste eingeschrieben werden oder nicht.
- (3) Die Liste **aktive@guug.de** haben wir zu Gunsten der Mitgliederpartizipation Mitte 2015 eingeführt. Dort sind gegenwärtig diejenigen 30 Mitglieder versammelt, die aktiv die GUUG gestalten. Wenn die GUUG Aktivitäten plant oder bespricht, zum Beispiel das nächste FFG oder eine Weihnachtsaktion, so passiert das auf dieser Liste.
- (4) Unter der Adressen **vorstand@guug.de** und **kontakt@guug.de** (aktuell sind das Aliase) ist der Vorstand für Anfragen sowohl von Mitgliedern als auch von Externen erreichbar. Diese Anfragen landen in einem Ticketsystem. Dort eingegangene Anfragen bearbeiten Vorstandsmitglieder entweder direkt, besprechen sie auf den Vorstandssitzungen (Agenden und Protokolle finden sich im Wiki) oder direkt mit den Aktiven.
- (5) Unter **gmv@guug.de** können Mitglieder direkte Fragen zur Mitgliederverwaltung stellen. In der Regel beantwortet diese dann Christian Lademann, der sich im Auftrag des Vorstands um die Beiträge und andere Mitgliederbelange kümmert.

Wer sich zu den beschriebenen Zwecken auf eine der Listen subscribieren möchte, erledigt dies unter <https://lists.guug.de/listinfo/>.

Die interne Liste **vs@guug.de** verwendet der Vorstand zur organisatorischen Abstimmung (etwa Terminabsprachen) oder für Fragen, die der Vertraulichkeit bedürfen (etwa Personalfragen,

die aber in den letzten zwei Jahren nicht vorgekommen sind). Diese Liste dürfen nur Subscriber lesen und beschicken. Wir bitten Mitglieder, Anfragen nicht hierhin, sondern an **<vorstand@guug.de>** zu richten – oder noch besser direkt an die Aktiven unter **<aktive@guug.de>**.

Über Nils



Nils Magnus ist seit 2014 Mitglied im Vorstand des *German Unix Users Group* e.V. (GUUG). Beruflich verdingt er sich als Engineer sicherer Systemarchitekturen und Fachautor. Er befasst sich mit Vorliebe mit Cloud-Infrastrukturen auf Basis von Open Source. In seiner Rolle als Veranstalter beim *LinuxTag* und der GUUG organisiert er seit über 15 Jahren Konferenzen und Workshops. Nils lebt sich gerade in Berlin ein.

Ein Augenzeuge berichtet GUUG-Frühjahrsfachgespräch 2016

Vom 23. bis 26. Februar 2016 fand das 15. Frühjahrsfachgespräch (FFG) der GUUG in Köln statt. Der Autor wirkte als Mitglied des Programmkomitees an der Zusammenstellung des Vortragsprogramms mit und graste anschließend auf der Veranstaltung seine Lieblingsthemen ab.

von Lenz Grimmer

Wer nicht neugierig ist,
erfährt nichts.

Hoffräulein Sora in *Der Triumph der Empfindsamkeit*
von J.W. v. Goethe

Die Auswahl war gar nicht so leicht: Es waren viele hochwertige und spannende Themen eingereicht worden, gab aber leider nicht genug Slots für alle (Abbildung 1). Wie üblich gab es an den ersten beiden Veranstaltungstagen ganztägige Tutorien: Die Teilnehmer eigneten sich zu Themen wie *OpenStack*, *Puppet4*, *SaltStack* oder *Ceph* praxisrelevantes Wissen an. Am Donnerstag und Freitag lief das Vortragsprogramm in zwei parallel verlaufenden Sessions. Die Vorträge waren eng getaktet.



Abbildung 1: Gut gelaunt und stolz aufs Ergebnis: Dirk Wetter (rechts), Chef des FFG-Programms, hat dieses Jahr einen Relaunch des Programmkomitees vollzogen. (Foto: Anika Kehrer)

Mein Vortrags-Highlight war die Keynote *Von Kutschern, Schriftsetzern, Managern und Systemadministratoren* von Johann-Peter Hartmann, der sich als *Chief Tailwind Officer* von *Mayflower* bezeichnet. In diesem sehr unterhaltsamen und auch provokanten Vortrag stellte Hartmann die in den meisten Firmen immer noch vorherrschenden hierarchischen Strukturen in Frage. Er zeigte, wie Kollaboration über das Internet und Open Source neue Wege öffnet, sich zu organisieren und effektiv miteinander zu arbeiten.

Die darauf folgenden Fachvorträge behandelten Themen wie Storage (*Ceph*, *LVM*, *Samba*, Replikation), Netzwerk (Security, IPv6, Firewalling) und Administration (*Ansible*, *Puppet*, *Consul*). Viele der Vorträge wurden von Entwicklern aus den jeweiligen Projekten gehalten, sodass die Themen fundierten technischen Tiefgang hatten. Ich selbst habe mir aus dem reichhaltigen Angebot unter anderem die folgenden Vorträge angehört.

Tag 1: Riesendaten-Replikation, *Ceph*, Shell-Geschichte und *gnuplot*

In *MARS: Langstrecken-Replikation von Petabytes* berichtete Thomas Schöbel-Theuer über sein Open-Source-Projekt bei 1&1, mit dem er dort die Rechenzentrums-übergreifende Replikation von Daten realisiert (über 800 Storage-Server mit insgesamt mehr als 12 PByte Datenvolumen). *MARS* ähnelt in seiner Funktionsweise dem *Distributed Replicated Block Device* (DRBD), arbeitet im Gegensatz dazu aber in einem asynchronen Modus, was es gerade für Langstrecken-Replikation sehr geeignet macht. Das Funktionsprinzip erinnerte mich stark an die asynchrone *MySQL*-Datenbankreplikation. Thomas würde sich über mehr Anwender und Tester freuen: Er arbeitet momentan an der Aufnahme seines Codes in den Mainline Linux-Kernel.

In seinem Vortrag *Ceph – Aufbau und Anwendung eines objektbasierten Speichersystems* gab Christian Schubert Einblick in das Cluster-Storagesystem *Ceph*, zum Beispiel dessen Aufbau und Funktionsweise, und wie sie einen Ceph-Cluster aus zum Teil ausgemusterten Servern aufgebaut haben. Christians Vortrag zeigte mir, dass die Speicherung von Daten in einem Ceph-Cluster sehr skalierbar und robust gegen Ausfälle ist, es aber im Handling und der Performance Verbesserungspotential gibt.



Social Event im Kölner Museum für Ostasiatische Kunst

Der gesellige Abend (Abbildung 2, 3 und 4) gastierte diesmal in einem Museum. Organisator und GUUG-Vorstandsvorsitzender Martin Schulte dachte sich: Wenn schon, denn schon – wenn man schon mal hier ist, kann man auch gleich eine Führung buchen. Diese Führung durch die zu der Zeit angebotene Sonderausstellung mit Leiko Ikemura (Abbildung 5) war für die Besucher des geselligen Abends eine freudige und gut angenommene Überraschung.



Abbildung 2: Man durfte die Führung auch schwänzen und sich so entspannen wie – von links – Willi, UpTimes-Redaktionsmitglied Robin und Andreas. (Foto: Anika Kehler)



Abbildung 3: Sitzen, quatschen und anstoßen mit – von links – Jörg, Dieter, Katharina und Henning. (Foto: Anika Kehler)

Ein Teilnehmer kam sogar ins Schwärmen: „Die Ausstellung war wunderbar. Ich kannte die Künstlerin nicht, bin froh dass ich sie kennengelernt habe und möchte nachlesen, was sie noch alles so gemacht hat. Ich finde, dass sehr schön gelungen ist, wie die Ausstellung diese vergleichsweise moderne Künstlerin der Kunst gegenüberstellt, die sich über Jahrtausende angesammelt hat. In der Gegenüberstellung schaut man dann, was passiert, ob die Sachen eher gegeneinander gestellt sind, oder in irgendeiner Tradition aneinander anschließen.“

Die Museumsführerin habe herausgearbeitet, schwärmt er weiter, wie die verschiedenen Orte, an denen die mit 20 Jahren ausgewanderte Künstlerin gewesen war, ihre Arbeit jeweils beeinflusst habe. Aber plötzlich wird sein Blick klar. „Was man auch sagen muss“, hebt er entschlossen an: „Die Museumsführerin hat das Publikum definitiv unterschätzt. Sie hat zum Beispiel gesagt, sie fände es schön, dass so Leute wie wir jetzt auch mal hier sind, denn uns würde man ja eher selten sehen. Das ist was, was ITlern bestimmt oft passiert. Die sind besser gebildet, als die Welt da draußen annimmt.“



Abbildung 4: Von Sebastian – ganz rechts – haben es nur Bauch, Hand und Nase aufs Bild geschafft. Von links nach rechts freuen sich im Übrigen des Lebens: Oliver, Steffen, Heinz, Dirk, Harald, Udo und Rainer. (Foto: Anika Kehler)



Abbildung 5: Blick in den Ausstellungsraum der Sonderausstellung von der japanischen Wahldeutschen Leiko Ikemura, *All About Girls and Tigers*. (Foto: © Museum für Ostasiatische Kunst, Alexandra Malinka)

Die Exponate der Künstlerin wurden im vorderen Teil der Ausstellung kontrastiert von historischen Exponaten, sodass auf ungefähr drei oder vier moderne Stücke ein historisches kam. Erst im hinteren Teil waren die klassischen Exponate zu finden. Man konnte sich also auch allein nach hinten begeben, wenn einen das mehr interessierte. In Führungen wird unter Umständen nämlich auch mal „viel reininterpretiert“, fand ein anderer Besucher: „Die Lebenssituation einer aus Japan in bewusster Abgrenzung zu Japan ausgewanderten Frau, die sich in den 1970ern von der damaligen Frauenbewegung emanzipiert und dann doch die Liebe zu ihrer Heimat wiederentdeckt hat – das kann man alles in einem Bild sehen, aber man muss es nicht unbedingt sehen.“ Fair enough. (Anika Kehler)

Jörg Schilling gab in seinem Beitrag *Von der Bourne- zur POSIX-Shell und weiter* einen umfangreichen Rundumschlag über die Geschichte der Unix-Shells. Er stellte die Unterschiede und Besonderheiten der prominentesten Vertreter vor (etwa *Bourne-* und *Korn-Shell*, *Bash*). Interessant fand ich besonders die Historie der Shell-Urahn, ihrer Abkömmlinge und Nachbauten, sowie die teilweise doch großen Unterschiede zwischen den einzelnen Implementierungen.

Der Vortrag *gnuplot – Malen mit Zahlen* von Harald König bildete einen gelungenen Abschluss des ersten Konferenztags. Nach einer kurzen Einführung bestand der Rest seines Beitrags aus einer Live-Hacksession an der Konsole, in der Harald interessante Graphen hervorzauberte, um die Möglichkeiten von Gnuplot zu demonstrieren. Es war eindrucksvoll zu sehen, wie Gnuplot aus trockenen Zahlen mit wenigen Kommandos ausdrucksstarke Graphen generiert.

Tag 2: Kanban, Doku und Docker

Am Freitag gefiel mir der Vortrag *Adminteam und Projekte mit Kanban organisieren* von Thorsten Kramm besonders gut. Thorsten vermittelte einen guten Überblick über diese Methode zur Arbeitsorganisation, und wie sie sich im Kontext der Systemadministration erfolgreich umsetzen lässt. Ich habe aus diesem Vortrag einige wertvolle Anregungen für die eigene Arbeit mitgenommen.

Auch der Vortrag *Lessons learned: IT-Dokumentation in der Praxis* von Anika Kehler sprach mir aus der Seele – das Thema Dokumentation wird leider viel zu oft immer noch stiefmütterlich behandelt. Das Thema ist ein heikles Thema, zeigten auch die während des Vortrags aufflammenden Diskussionen. Es war ein sehr lebhafter Austausch zwischen der Vortragenden und den Teilnehmern.

Erkan Yanars Vortrag *World of Docker* konnte ich leider nur noch zum Teil mitnehmen, da ich meinen Zug nach Hause erwischen musste. Was mich positiv beeindruckt hat, war Erkans Präsentationsmethode: Er malte einfach Bilder und Diagramme mit einem Grafiktablett, um Dinge zu veranschaulichen. Die teilweise eingestreuten animierten GIFs fand ich persönlich etwas störend, aber es hat den Vortrag aufgelockert. Zum Thema selbst hatte Erkan wie üblich einiges Neues zu berichten. Das Thema Linux-Container ist jedenfalls aus einer modernen IT-Landschaft nicht mehr wegzudenken und bietet viel Potential.

Ein Novum war dieses Jahr die Möglichkeit, Vorträge direkt online zu bewerten und individuelles Feedback zu Qualität und Inhalt abzugeben. Das stellt für das Programmteam nützlichen Input für die Organisation des Events im nächsten Jahr dar.

Der gesellige Teil des FFG fand dieses Mal in einer sehr exotischen Location statt: dem *Kölner Museum für Ostasiatische Kunst*. Der Abend bot neben einem reichhaltigen und schmackhaften Buffet den richtigen Rahmen für Gespräche und Networking. Wer wollte, konnte an einer interessanten Führung durch das Museum teilnehmen, wo zu diesem Zeitpunkt die Sonderausstellung *Leiko Ikemura. All About Girls and Tigers* zu sehen war (siehe Kasten *Social Event im Kölner Museum für Ostasiatische Kunst*).

Ich habe in diesen zwei Tagen viel Neues gelernt, alte Bekannte getroffen und wichtige neue Kontakte geknüpft. Die Veranstaltung stellte durch die interessanten und fundierten Vorträge Qualität unter Beweis, sowie durch die Möglichkeit, direkt mit den durchweg guten Referenten ins Gespräch zu kommen. Grund, um nächstes Jahr wieder mit dabei zu sein, vielleicht dieses Mal mit einem eigenen Vortrag.

Über Lenz



Lenz Grimmer ist bei der *it-novum* in Fulda für die Weiterentwicklung des quelloffenen Storage-Managementsystems *openATTIC* verantwortlich. Er befasst sich seit seinem Informatikstudium in den 1990er Jahren privat und beruflich mit Linux und Open-Source-Software. Lenz ist häufig auf Veranstaltungen wie dem GUUG FFG anzutreffen, wo er als Referent Vorträge zu den unterschiedlichsten OSS-Technologien hält (zum Beispiel Storage und Infrastruktur). Sein privates Blog und weitere Links sind unter <http://blog.lenzg.net/> zu finden.

HOW DO YOU KNOW IT'S AGILE?



Erklimmen eines Babylonischen Turms *Industrie 4.0 und das Internet der Dinge*

Die Medien sind derzeit voll von Informationen zum Thema Digitalisierung der Industrie. Konzerne werben für sich als Führer in diesem neuen Wirtschaftsbereich. Der Artikel beschreibt die derzeitigen Modell-Architekturen und Protokolle hinter den Schlagworten *Industrie 4.0* und *Internet der Dinge*.

von Hartmut Streppel

Beziehungen schaffen,
am liebsten
zwischen allen Dingen der Welt.

Kurt Schwitters über sein Kunstprinzip *Merz* (Kontext: Dadaismus)

Bei einem Besuch in Berlin fiel mir der Spruch von Kurt Schwitters weniger ein, als vielmehr ins Auge (Abbildung 1). Er erinnerte mich daran, dass ich über das Thema *Internet der Dinge* und *Industrie 4.0* gern mal einen Artikel schreiben wollte. Meinen Zugang dazu erhielt ich ursprünglich über einen Freund, der ein sehr informatives Buch darüber herausgegeben hat [1]. Schließlich lauschte ich im Februar einem Webcast der *Computerwoche* [2]. Darin waren sich die Experten einig, dass es im Umfeld dieser beiden Themen viele unterschiedliche Plattformen geben und sich der Stärkere durchsetzen werde.



Abbildung 1: Fundstück aus den Straßen Berlins. (Foto: Hartmut Streppel)

Nun ist die Existenz mehrerer konkurrierender Plattformen nichts Neues, dachte ich mir, und auch gar nicht so schlecht, solange standardisierte Schnittstellen existieren, über die andere die Plattformen nutzen können. Offene, standardisierte Schnittstellen sind ja eines der Lieblingsthemen der *GUUG*-Mitglieder. Offene Systeme leben von Standards und offenen Referenzimplementierungen. Deshalb liegt der Fokus dieses Artikels bei den grundlegenden Architekturen, die derzeit diskutiert und entwickelt werden, sowie bei den verwendeten Protokollen.

Industrie 4.0, IoT und Industrial Internet

Nachdem die diesjährige *Hannover Messe* „boomt[e] mit Industrie 4.0“ [3] und Präsident Obama als Vertreter des Gastlandes USA damit auch in den Nachrichten vertreten war, hat nun jeder den Begriff zumindest mal gehört. Mein Postkasten ächzt täglich unter der Last der Nachrichten zum Thema, heute vormittag zum Beispiel: „Digitalisierung im Fokus“ in der Betreffzeile einer E-Mail vom Bundesministerium für Bildung und Forschung, oder „xyz Launches Cloud Network for the Internet of Things“, und das war erst der Anfang.

Die Begriffe sind etwas verwirrend und nicht klar voneinander abgegrenzt. Industrie 4.0 klingt nach neuartigen, digitalisierten Produktionsprozessen. Die *acatech* (Deutsche Akademie der Technikwissenschaften) unter ihrem Präsidenten Henning Kagermann hat diesen Begriff geprägt und ihn erstmals 2011 im Rahmen der Hannover Messe an die Öffentlichkeit getragen.

Er ist inzwischen zu einem Exportschlager geworden, wie früher ‚Made in Germany‘, und wird selbst im Ausland oft in deutscher Schreibweise genutzt. 4.0 steht für die vierte industrielle Revolution – nach der dritten, die mit der Einführung elektronischer Steuerungen (SPS, steht für speicherprogrammierbare Steuerung) in den Produktionsprozess die Industrie revolutionierte. Die Verbände, die an der Weiterentwicklung von Industrie 4.0 beteiligt sind, wollen die Führerschaft der deutschen Industrie auch im digitalen Zeitalter halten. Wer als erster Standards setzt und diese zum Beispiel durch Referenzimplementierung durchsetzt – *TPC/IP* steht hier Pate – und dann noch aus einer Position der Stärke heraus, der dürfte einen gewaltigen Vorteil haben.

Für amerikanische Firmen vor allem aus dem

Software-Bereich gelten Dienstleistungen auf der Basis von Daten als das Geschäft der Zukunft – Daten, die in großen Mengen und sehr einfach, da freiwillig und kostenlos, über das Internet gewonnen werden, etwa in sozialen Netzen. Google zum Beispiel verdient mit dem Verkauf von Werbung auf der Basis von Nutzerdaten ein Vermögen. Das *IoT* (Internet of Things and Services) verbindet die reale und die virtuelle Welt, indem es reale Objekte, vom Herzschrittmacher bis zum Kühlschrank, mit Hilfe von Sensoren und Aktoren digitalisiert und über das Internet mit netzbasiereten Diensten verbindet.

Völlig neue Geschäftsmodelle sind auf Basis vernetzter Dinge machbar, nämlich mit Hilfe der Daten, die sie liefern, und mit Hilfe der Algorithmen, die die autonome Steuerung der Dinge durch die Dinge selbst ermöglichen. Dass der Kühlschrank neue Milch bestellt, kurz bevor sie aufgebraucht ist, ist ja schon ein bekanntes Beispiel. Dass der Herzschrittmacher den Notarzt ruft, wenn der Patient ein ernstes Problem hat, ist auch eine naheliegende Lösung. Dass die Haustür dem frischgebackenen Ehemann verbietet, seine Liebste über die Schwelle zu tragen, weil er a) zu alt und zu schwach und b) nicht ausreichend krankenversichert ist, ist zwar heute noch lustig [4], aber in ein paar Jahren durchaus realistisch.

Ein dritter Begriff, das *Industrial Internet*, scheint mir die heutigen Möglichkeiten am besten zu fassen. Den Begriff hat der amerikanische Industriegigant *General Electric* (GE) vor ein paar Jahren geprägt: „Das [Industrial Internet] basiert auf der Konvergenz der globalen Industriesysteme und der Kompetenz der modernen Datenverarbeitung, auf analytischen Methoden, kostengünstiger Sensorik und der neuen Konnektivität, die das Internet ermöglicht hat.“ [5] Jeff Immelt, der CEO von GE, erklärt seine Vision in einem Youtube-Video recht drastisch [6]: Eine Firma, die heute als Industrieunternehmen zu Bett geht, wache am nächsten Morgen als Software- und Analytics-Unternehmen wieder auf.

Der Fokus geht also weg von den Maschinen und den damit erzeugten Produkten hin zur Software und der Verarbeitung von Massendaten mit Big-Data-Methoden und künstlicher Intelligenz. Die Intelligenz steckt aber nicht mehr nur in der Fertigung, sondern auch im Produkt und der das Produkt umgebenden Software. Übersehen wird häufig der Aspekt, dass die Daten, die ein Produkt liefert, Teil der Wertschöpfungskette werden. Heute hat der Produzent vielleicht noch einen Wartungsvertrag für das verkaufte Produkt. In Zu-

kunft hat er ein weiteres Geschäftsfeld mit den Daten, die seine Produkte auch nach dem Kauf noch liefern.

Dinge liefern Daten

Dass Sensoren in Produkten unabhängig davon, wo sie eingesetzt werden, auch über große Entfernungen weltweit sinnvolle Daten liefern können, ist mir das erste Mal bewusst geworden, als Flug MH370 im März 2014 im Indischen Ozean verschwand. Die Triebwerke des Flugzeugs lieferten zwar keine wichtigen Analysedaten – den dazu notwendigen Vertrag hatte die Fluggesellschaft nicht abgeschlossen. Aber die Triebwerke meldeten sich regelmäßig über Satellit bei ihrer Bodenstation. Aus diesen Kommunikationsdaten konnte ungefähr auf Absturzort und -zeit geschlossen werden.

Dieses Beispiel weiter gedacht zeigt ein recht neues Geschäftsmodell im Produktsupport. Anstatt zu warten, bis ein Kunde ein Problem hat, es entdeckt, dem Produzenten meldet und auf den Support wartet, überwachen heute die Produzenten ihre Produkte über das Internet. Sie informieren den Kunden über notwendige Wartungsarbeiten, noch bevor das Problem beim Kunden auftritt.

Das Gesundheitswesen erprobt ähnliche Verfahren. Heute sind Sensoren bei kranken Menschen im Einsatz, überwachen beispielsweise den Blutzucker und den Herzschrittmacher. In der Wohnung gebrechlicher Menschen sind Bewegungssensoren installiert. Die Sensoren lösen selbstständig Alarm aus, wenn sie einen Notfall detektieren, der etwa als Bewegungslosigkeit über einen bestimmten Zeitraum hinweg definiert ist.

Noch einen Schritt weiter geht das Beispiel eines Mountainbike-Produzenten. Wenn er im Wettkampf Sportler betreut, die seine Produkte nutzen, verfügt er über alle Produkt- und Produktionsdaten über das Netz: Das Fahrrad ist an allen sensiblen Stellen mit Sensoren ausgestattet. Beim Wettkampf kann nun ein Techniker in Echtzeit die vom Rad gesendeten Daten lesen und sogar sofort mit den Konstruktionsdaten verknüpfen. Wenn wir in der Lage sein werden, mit 3D-Druckern auch die kompliziertesten Teile zu erstellen, könnten nun direkt nach dem Wettkampf neue Teile mit neuen, verbesserten Charakteristika hergestellt und im Rad für das nächste Rennen eingebaut werden.

Wer phantasiebegabt ist, kann in der Industrie anheuern und sich die tollsten Dinge ausdenken. Die Konzepte und Technologie dafür sind heute bereits vorhanden.

Wirtschaft probt Referenzarchitekturen

Mehrere Organisationen bemühen sich, das Internet der Dinge und Industrie 4.0 voran zu treiben. In den USA ist dies das *Industrial Internet Consortium* (IIC), das die *Industrial Internet Reference Architecture* (IIRA) entwickelt [7]. Dort heißt es: „Das Industrial Internet ist ein Internet der Dinge, Maschinen, Computer und Menschen, das intelligente industrielle Abläufe durch fortgeschrittene Datenanalyse ermöglicht, um transformierende Geschäftsergebnisse zu erzielen. Es verkörpert die Konvergenz des globalen industriellen Ökosystems, der fortschrittlichen IT und Herstellung, der umfassenden Sensorenverwendung und der allgegenwärtigen Netzwerkanbindung.“ ([8], [9])

In Deutschland ist es vor allem die *Plattform Industrie 4.0* [10], die sich dem Thema Industrie 4.0 verschrieben hat. In ihr arbeiten die Wirtschaftsverbände BITKOM, ZVEI (Zentralverband Elektrotechnik- und Elektronikindustrie) und VDMA (Verband Deutscher Maschinen- und Anlagenbau) mit mehreren Bundesministerien gemeinsam an der Weiterentwicklung und Umsetzung von Industrie 4.0. Die Bundesregierung listet Industrie 4.0 als einen Teil ihrer Hightech-Strategie [11]. Im Statusreport zum deutschen *Referenzarchitekturmodell Industrie 4.0* (RAMI 4.0) [12] heißt es aus Seite 6 lapidar: „Industrie 4.0 ist eine Spezialisierung des *Internet of Things and Services*.“ Ein Strategiepapier derselben Urheber zur Umsetzung ergänzt: „Als neuer Aspekt kommt bei Industrie 4.0 die Einbeziehung des Produkts bzw. Werkstücks hinzu.“ [13]

Ein weiterer wichtiger Begriff bei der Diskussion über das Internet der Dinge ist *Cyber Physical System* (CPS). In diesem Begriff sind explizit die reale, physische mit der virtuellen Cyberwelt vereint. Von dieser Verbindung war zwar weiter oben schon die Rede. Aber dieser Begriff bringt es auf den Punkt, was weder der Begriff Industrie 4.0 noch der Begriff IoTS erreicht. Der VDI geht in seiner Stellungnahme *Cyber-Physical Systems: Chancen und Nutzen aus Sicht der Automation* von folgender Definition aus: „Cyber-Physical Systems sind gekennzeichnet durch eine Verknüpfung von realen (physischen) Objekten und Prozessen mit informationsverarbeitenden (virtuellen) Objekten und Prozessen über offene, teilweise globale und jederzeit miteinander verbundene Informationsnetze.“ [14] In dem Papier kommt er unter anderem zu folgenden wichtigen Schlussfolgerungen über so genannte *Cyber-Physical Production Systems* (CPPS):

- Der Automationsgrad wird mit CPS ansteigen.
- Security und Safety sind kritische Erfolgsfaktoren für die Realisierung von CPPS.
- Der Mensch muss bei der Einführung und beim Einsatz von CPS im Mittelpunkt stehen.
- Es werden neue Geschäftsmodelle durch CPS möglich.

Ziel: Praxistauglichkeit

Insbesondere die Geschäftsmodelle sind ein kritischer Punkt dieser digitalen Revolution. Schaut man nur mal auf die vollständige Digitalisierung eines Fertigungsunternehmens – wobei dies nicht nur die Fertigung, den *shop floor*, sondern auch den *office floor* einbezieht – und zieht man vielleicht noch eine Digitalisierung der gefertigten Produkte in Betracht, vergisst aber am Schluss, Geschäftsmodelle um das Produkt herum zu entwerfen, dann werden dieses Geschäft andere machen.

Diese Argumentation ist auch der Hintergrund der kritischen Diskussionen zur Zukunft der (deutschen) Automobilindustrie, die zu einem einfachen Produktlieferanten für die Software-Industrie werden könnte [15]. In dieser Diskussion gelten zwei Bereiche als Herausforderung für die Automobilindustrie: a) die Digitalisierung des Fahrzeugs, von der weitergehenden Automatisierung bis zum autonomen Fahren und vollständiger Vernetzung; und b) der Wandel vom Produkt zum Service durch Einführung neuer Geschäftsmodelle und Mobilitätskonzepte.

Offene Systeme und das Internet haben geholfen, dass „anyone anytime anywhere with any device“ kommunizieren kann (Sun-Microsystem-Werbung aus den 2000er Jahren). Die Internetpioniere haben mit der TCP/IP-Protokollsuite, die Unix-Entwickler mit offenen Schnittstellen und Referenzimplementierungen dabei geholfen. Das Internet der Dinge baut auf diesen Technologien auf. Referenzarchitekturen helfen, Plattformen zu entwickeln, auf denen dann Lösungen erarbeitet und ausgerollt werden. Das man sich dessen bewusst ist, zeigt zum Beispiel die jüngst gegründete Normungsinitiative für Industrie 4.0 in Deutschland [16]. Es gibt sogar eine Initiative, die Industrie-4.0-Interessierten den Zugang zu Testfeldern ermöglichen will und von beiden Seiten – Bundesregierung und Industrie – Unterstützung erfährt [17].

Die beiden Vereinigungen *IIC* und *Plattform Industrie 4.0* haben Anfang 2016 gemeinsam ihre Architekturen einem kritischen Blick unterzogen. In der Pressemitteilung zu diesem Treffen findet man ein Diagramm, das die Ähnlichkeiten, aber auch die Unterschiede deutlich macht. [18]

Das deutsche Referenzarchitekturmodell RAMI 4.0

Das deutsche Referenzarchitekturmodell Industrie 4.0 (*RAMI 4.0*, [12]) ist ein dreidimensionales Modell, das sich stark an existierende Modelle anlehnt, existierende und fehlende Standards benennt und Untermengen von Standards definiert, mit denen eine schnelle Umsetzung von Teilinhalten für Industrie 4.0 möglich ist (Abbildung 2). Die senkrechte Achse verwendet Schichten für die unterschiedlichen Sichtweisen von der Geschäftssebene bis zu den Assets. Auf der waagerechten Achse ist der Produktlebenszyklus von der Entwicklung bis zu Nutzung und Support dargestellt. Die dritte Achse benennt Funktionalitäten und Verantwortlichkeiten.

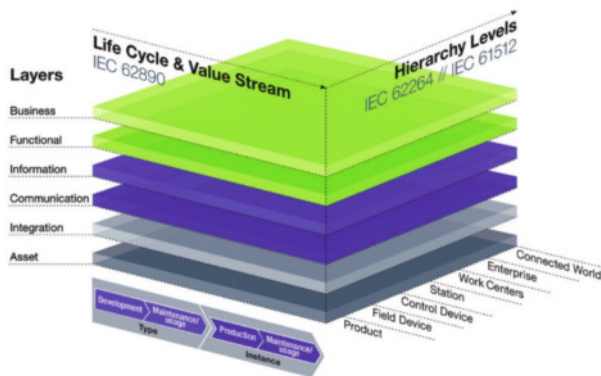


Abbildung 2: Das deutsche Referenzarchitekturmodell RAMI 4.0 von April 2015. (Quelle: © Zentralverband Elektrotechnik- und Elektronikindustrie e.V. (ZVEI), [19])

Dieses Modell verfeinern die Erfinder im weiteren auf die typischen Ebenen eines Fertigungsunternehmens. Sie definieren so genannte Verwaltungsschalen, die aus einem nicht-Industrie-4.0-konformen Ding eine Industrie-4.0-Komponente machen. Mit diesem Mechanismus können natürlich schnell auch große, alte Teile in eine Industrie-4.0-Umgebung integriert werden. Leider geht dieses Referenzarchitekturmodell kaum auf das produzierte (Industrie-)Produkt ein, das ja die wesentliche Erweiterung der bisherigen Sicht auf die Wertschöpfungskette eines Unternehmens darstellt.

Die amerikanische Referenzarchitektur IIRA

Aus Sicht der amerikanischen Referenzarchitektur *IIRA* [7] besteht ein *Industrial Internet System* (IIS) aus fünf so genannten *Functional Domains* (Abbildung 3): Control, Operations, Information, Applications und Business. In der *Control Domain* geschieht die gesamte Kommunikation mit Sensoren und Aktoren. Die *Operations Domain* verwaltet das Gesamtsystem von der Installation über die Überwachung bis zur Außerbetriebnahme. Die *Information Domain* sorgt für Verwaltung, Verarbeitung und Weiterleitung von Daten. Die *Applications Domain* stellt Schnittstellen nach außen zur Verfügung. Und in der *Business Domain* laufen die unternehmenskritischen Anwendungen, die auf die in der Control Domain von den Dingen gelieferten Daten zugreifen, daraus Schlüsse ziehen und Geschäftsentscheidungen treffen, eventuell sogar in Echtzeit.

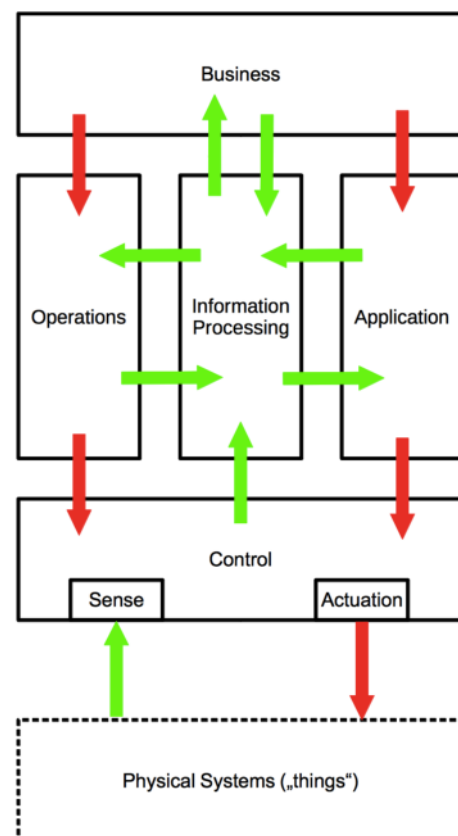


Abbildung 3: Die fünf *Functional Domains* der amerikanischen Referenzarchitektur *IIRA*. Grüne Pfeile stehen für Datenflüsse, rote Pfeile für Kontrollflüsse. (Leicht vereinfachte und variierte Darstellung. Das Original der Quelle © IIRA ist zu konsultieren unter [8], S. 28.)

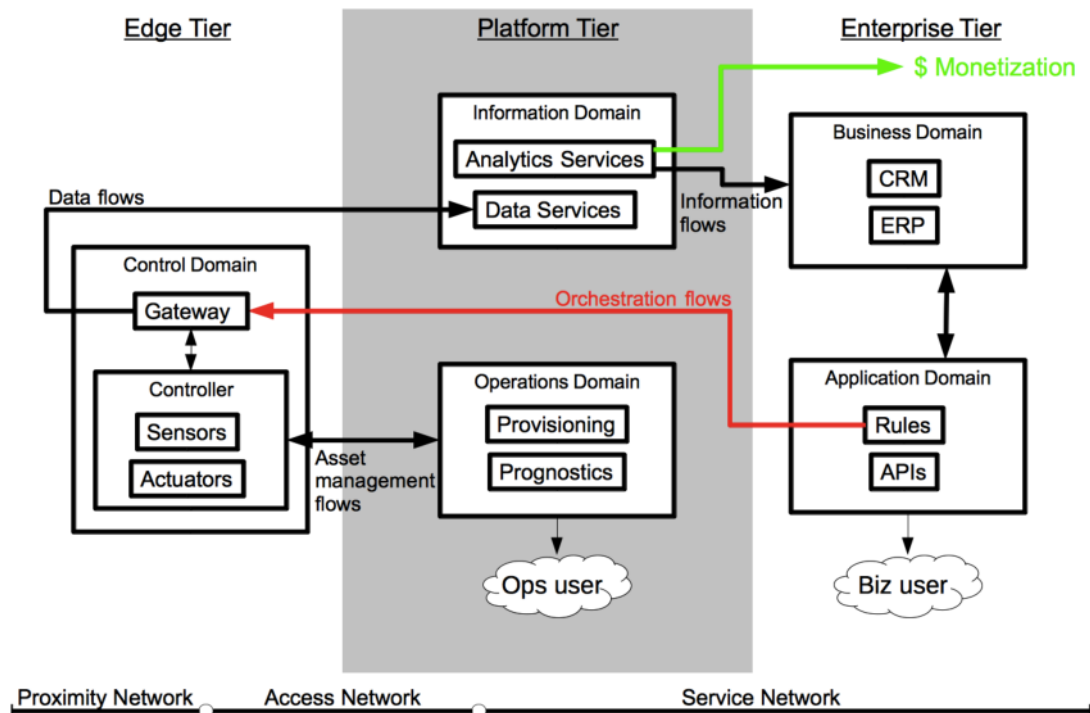


Abbildung 4: Die drei Ebenen der amerikanischen Referenzarchitektur IIRA mit ausgewählten Interaktionen. (Stark vereinfachte Darstellung. Das Original der Quelle © IIRA ist zu konsultieren unter [8], S. 39.)

Diese fünf Domains werden nun in eine IT-typische Drei-Ebenen-Architektur verteilt: Edge, Platform und Enterprise (Abbildung 4). Sensoren, Aktoren (Elemente, die etwas tun können) und zugehörige Elemente aus der Control Domain bilden die Edge-Schicht. In der Realität werden die Dinge dieser Schicht tatsächlich weit weg vom traditionellen Rechenzentrum angesiedelt sein. Die gesamte Verwaltung, also die Operations-Domain, und die Datenverarbeitung, also die Information-Domain, bilden die Platform-Schicht. Die Applications Domain und die Business Domain schließlich bilden die Enterprise-Schicht. In dieser Schicht werden ERP-, MES- und Analytics-Systeme betrieben, die zur Steuerung des Unternehmens notwendig sind.

Die europäische Referenzarchitektur IoT-A: Ein Ding im IoT

Um in die Implementierung eines einzelnen „Dings“ in der Edge-Schicht zu schauen, folgt ein Diagramm aus dem EU-geförderten Projekt IoT-A [20]. Dieses Projekt entwickelte ebenfalls eine Referenzarchitektur und beschrieb sie sehr detailliert und wissenschaftlich [21]. Das Diagramm in Ab-

bildung 5 verdeutlicht einige interessante Zusammenhänge, wie Dinge im Internet der Dinge implementiert sein können.

Das Diagramm zeigt unten rechts in den blauen Kästchen physische, reale Dinge, die aktiv oder passiv sein können (*Device*). Ein passives *Sensor*-Gerät kann nur Daten liefern, im einfachsten Fall über Auslesen von RFID-Chips. Ein aktives *Actuator*-Gerät kann zusätzlich Aktionen auslösen. *Devices* können natürlich auch viel komplexer als ein RFID-Chip sein, etwa wenn sie Daten speichern, ein eigenes Betriebssystem besitzen oder Daten anderer Sensoren weiterverarbeiten. Diese Hardware-Seite eines „Dings“ fasst das Diagramm zu einer *Physical Entity* zusammen.

Jede physische Geräte-Einheit wird durch eine entsprechende virtuelle Einheit repräsentiert – die Software-Seite, dargestellt in grün, genannt *Virtual Entity*. Beide zusammen bilden die Erweiterung *Augmented Entity*. Die virtuelle Einheit erhält durch zusätzliche Ressourcen (*Network Resource* und *On-Device Resource*) Metadaten über das Gerät, sodass sie auch ohne vorheriges Wissen über das Gerät dessen Eigenschaften und Daten interpretieren kann. Software-seitig stellt die virtuelle Einheit einen Dienst (*Service*) zur Verfügung, der für den Nutzer bereitsteht (*User*).

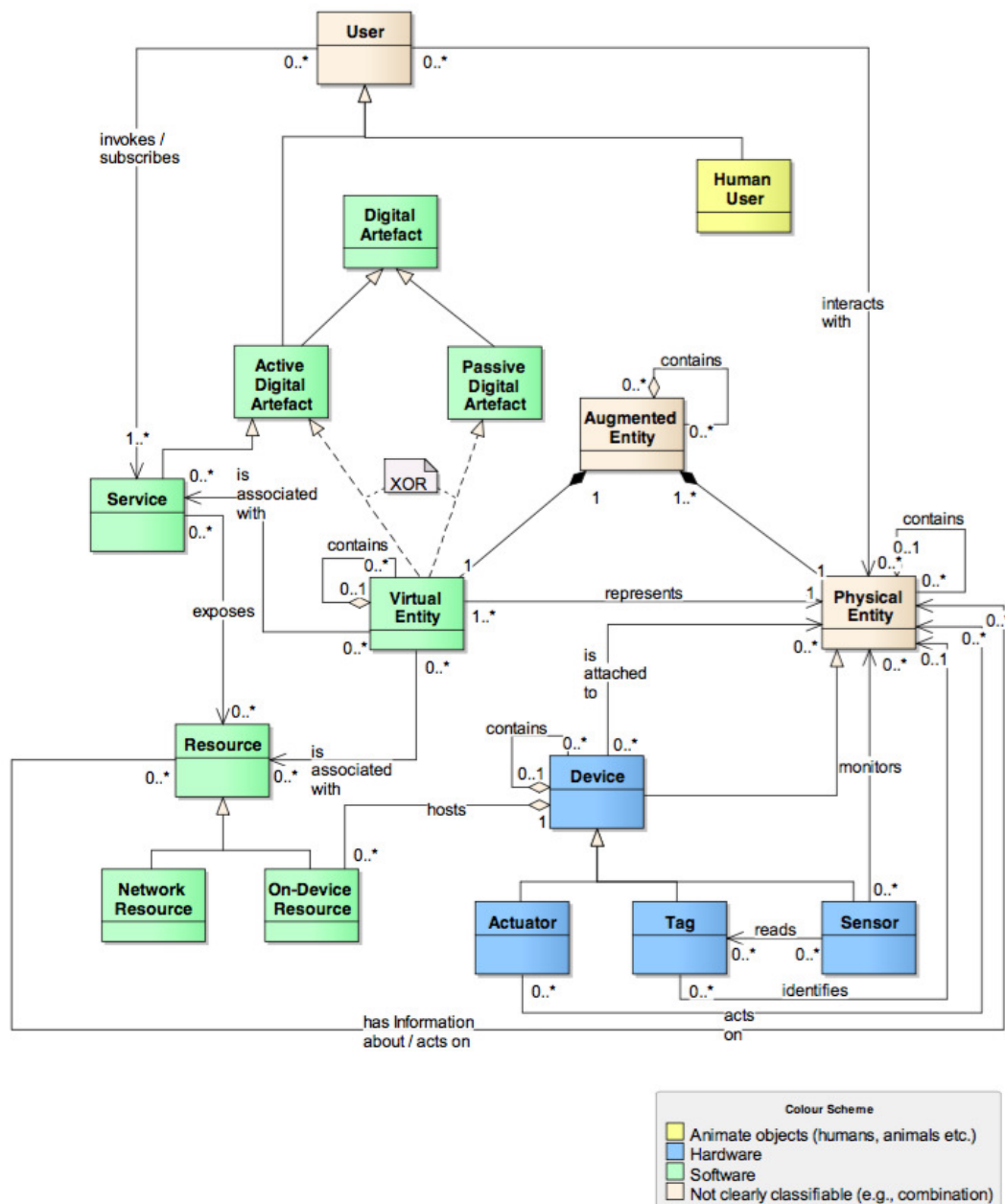


Abbildung 5: UML-Darstellung des europäischen IoT-Domain-Modells *IoT-A*. (Quelle: © EU-Projekt *IoT-A*, [21], S. 58.)

Der Nutzer kann ein Mensch sein (*Human User* in gelb), muss es aber nicht. Er kann auch selbst wieder ein Service sein, der beispielsweise autonom Daten abfragt, sie verarbeitet und daraufhin Aktionen auslöst. Wenn man sich in einer Maschine zum Beispiel einen Sensor vorstellt, der die Information liefert, wieviele Ressourcen der Maschine zur Verfügung stehen, so könnte eine andere Maschine diese Daten nutzen, um rechtzeitig Nachschub zu bestellen. Das wäre ein Beispiel für M2M-Kommunikation (Machine-to-Machine).

An dieser Stelle sei kurz auf die juristische Frage hingewiesen, ob ein von einer Maschine ausgelöster Bestellvorgang rechtliche Gültigkeit haben kann. Dies ist nur ein kleines Teilproblem der vie-

len juristischen Fragestellungen rund um das Internet der Dinge, vor allem um autonom handelnde Maschinen. Die Frage, wer bei einem von einem autonom fahrenden Auto verursachten Unfall haftet, ist eine weitere ungeklärte.

Standardisierte Protokolle

Ohne standardisierte Kommunikationsprotokolle kann das Internet der Dinge nur in homogenen Umgebungen funktionieren. Wer sich in der Geschichte der IT auskennt, weiß, dass das Internet und in Zukunft das Internet der Dinge nur funktionieren kann, weil einige sehr findige und weitsichtige Köpfe einen Satz von Protokollen entwor-

fen haben, der sich als außerordentlich robust und skalierbar erwies: die IP-Protokolle. Deren Entwickler Vint Cerf und Robert Kahn bekamen übrigens den Turing Award für ihren Beitrag im Jahr 2004 [22]. Auch wenn die Experten in den 1980-ern nicht mit der exponentiell wachsenden Zahl der Teilnehmer rechneten, die einmal vernetzt sein würden, ist ihr System flexibel genug, um ein neues IP-Adress-Schema (IPv6) nahtlos in die existierende IPv4-Welt zu integrieren. Mit 128-Bit langen Adressen kann eben jedes „Ding“ seine eigene Adresse bekommen und an der weltweiten Kommunikation über das Internet teilnehmen.

Diese neue vernetzte Welt benötigt Protokolle mit den unterschiedlichsten Eigenschaften: Geräte kommunizieren mit Geräten, Geräte mit Servern, Server mit der Cloud, Anwendungen verarbeiten Daten und schicken Ergebnisse wieder an Geräte. Entfernte Geräte in der Edge-Schicht des IIRA-Modells benötigen wegen ihrer speziellen Anforderungen kleine Datenmenge pro Übertragung, geringe Stromaufnahme, wenig Rechenleistung, also wieder besondere Protokolle.

Abbildung 6 zeigt den vom EU-Projekt BUTLER [23] zusammengetragenen Protokollstack mit unterschiedlichen Varianten auf den einzelnen Ebenen. In der Anwendungsschicht ist zum Beispiel das *Constrained Application Protocol* (CoAP) genau für einen solchen Anwendungsfall entstanden. Designziel war, sich leicht in HTTP-basierte Umgebungen zu integrieren, also in Web-Anwendungen. In der Netzwerkschicht wurde *IPv6 over Low power Wireless Personal Area Networks* (6LoWPAN) neu entwickelt, um die IPv6-Funktionalität über IEEE802.15.4-Verbindungen zur Verfügung zu stellen. Der IEEE802.15.4-Standard für *Wireless Personal Area Networks* beschreibt ein Funknetzwerk für drahtlose Sensornetze, die von am Körper getragene Sensoren und Aktoren gebildet werden.

Da es sich bei den Sensoren und Aktoren in der Edge-Schicht um möglichst einfache Geräte mit wenig Ressourcen handelt, sind die etablierten IP-Protokolle in der Regel zu mächtig und zu aufwändig. Sogar die Struktur der einzelnen Netzschichten wird als zu komplex angesehen. Also entstehen neue Protokolle für ganz bestimmte Anwendungszwecke, die über mehrere Schichten hinweg arbeiten. Ein Beispiel dafür ist *ZigBee*, eine Spezifikation für drahtlose Netze mit geringem Datenvolumen. ZigBee setzt auf den Standard IEEE802.15.4 auf. Eine 2002 gegründete Allianz von über 200 Firmen arbeitet daran mit [25].

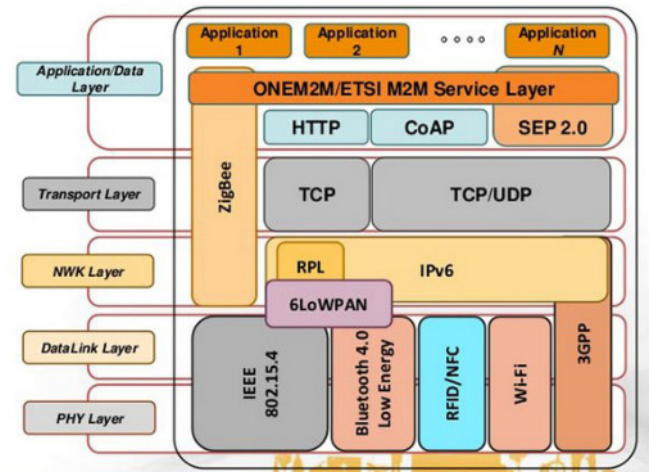


Abbildung 6: Die technologischen Herausforderungen des Internet der Dinge im Bereich Kommunikation. (Quelle: © EU-Projekt BUTLER [24])

Eine Reihe von Message-Queuing-Protokollen stehen für den Datenaustausch zur Verfügung: etwa MQTT (Message Queuing Telemetry Transport), XMPP (Extensible Messaging and Presence Protocol) – auch als Jabber bekannt – oder AMQP (Advanced Message Queuing Protocol). Auf der untersten Ebene kommen die bekannten Technologien von WPAN über Bluetooth, NFC, WLAN und GSM zum Einsatz.

Natürlich gibt es auch Entwicklungen, sehr ressourcensparende Betriebssysteme für Sensor- und Aktorennetze zu entwickeln. Aus der Linux-Community ist hier das Projekt *Zephyr* [26] zu nennen, das an einem kleinen, skalierbaren Realtime-Kern für unterschiedliche Architekturen arbeitet.

Big Data, KI und die Cloud

Das Internet der Dinge und Industrie 4.0 sind nicht einfach nur durch neue Architekturen, Protokolle und Prozesse umsetzbar. Auch die Werkzeuge, mit denen sich Daten speichern und verarbeiten lassen, erfahren Anpassung an die neuen Gegebenheiten, vor allem an die Masse der Daten. Heute analysieren Big-Data-Technologien große, unstrukturierte, nicht korrelierte Datenmengen; künstlichen Intelligenz erzielt mit neuen Lernmethoden neue Erfolge, um Wissen selbständiger zu erarbeiten; und Datenspeicher in der Cloud ermöglichen, diese großen Datenmengen von überall aus einfach abzulegen. Sensoren werden allgegenwärtig und liefern Daten über jeden und alles.

Allein standardisierte Architekturen und Protokolle gewährleisten hier weltweite Interoperabilität. Es wird unterschiedliche Plattformen geben, in den Markt gedrängt von den großen Playern der

Industrie. Solange offene Schnittstellen zur Verfügung stehen, ist der Markt für neue Anwendungen offen, die mit diesen Daten neue Ideen verwirklichen.

SIMPLY EXPLAINED



NO AJAX



AJAX

Allerdings ist das Verständnis davon uneinheitlich, was Industrie 4.0 und das Internet der Dinge einmal können sollen. Heute regiert Software die Welt – nicht die Dinge, die aus der Fertigung kommen. Die Diskussionen um die autonomen fahrenden Autos zeigen, wer am Ende seine Geschäftsmodelle umsetzen und damit Geld verdienen kann: Diejenigen, die das Software-Geschäft beherrschen. Der Mehrwert steckt nicht mehr nur in der noch einmal verbesserten Produktionsmethode, sondern in der durch Software realisierten Idee, was sich aus den Daten dieses tollen Geräts herausholen und zum Geschäft machen lässt.

Noch Fragen?

Für mich bleiben zwei Fragen. Wenn doch in naher Zukunft alles vernetzt und automatisiert ist; wenn die KI lernen kann und wichtige Entscheidungen trifft; wenn der Kühlschrank den Nachschub bestellt und der programmierbare Kochtopf das Essen kocht, das der Roboter-Butler serviert; welche Rolle haben wir Menschen dann noch? Werden wir endgültig zu auf Sofas liegenden, per Smartphone einkaufenden Konsumenten, zu *Shopping Couch Potatoes*?

Und: Wie baut man ein industrielles Internet, in dem der Mensch nicht zum überwachten Objekt wird?

Die einfachste Frage: „Was bringt das alles?“, beantwortet ein kurzer TED-Vortrag von Marco Annunziata, Chief Economist von General Electric [27]. Er ist sich sicher: Alles wird gut! Wenn er aber behauptet, dass es keine unerwarteten Stromausfälle mehr geben wird, dann muss ich an seiner Vision zweifeln. Auch hackbare Smart Meter (Stromzähler) sind Teil des Internet of Things. [28]

Links

- [1] Ulrich Sendler (Hg.): *Industrie 4.0; Beherrschung der industriellen Komplexität mit SysLM*. Springer Vieweg, Heidelberg 2013.
- [2] Webcast der *Computerwoche* zum Internet der Dinge:
http://www.idgevents.de/webcasts/319/internet_of_things_richtig_nutzen.html
- [3] Dr. Jochen Köckler, Vorstandsmitglied Deutsche Messe, im Messe-Abschlussbericht:
<http://www.hannovermesse.de/de/messe/daten-fakten/after-show-report/abschlussbericht>
- [4] Stunksitzung Kölner Karneval 2016, bei Minute 38:50:
https://www.youtube.com/watch?v=3_grSemg7JI
- [5] *General Electrics* über das Industrial Internet:
<http://www.ge.com/de/unternehmen/industrial-internet>
- [6] CEO Jeff Immelt von *General Electrics* über das Internet der Dinge:
<https://www.youtube.com/watch?list=PLxRhTjvLlyolpfSDoqMmnkNXP80sNh3tR&v=fHPmXxlzVnA>
- [7] Amerikanisches Projekt *Industrial Internet Reference Architecture* (IIRA) des *Industrial Internet Consortium* (IIC):
<http://www.iiconsortium.org/IIRA.htm>
- [8] Paper *Industrial Internet Reference Architecture* (IIRA) des *Industrial Internet Consortium* (IIC), S. 9:
<http://www.iiconsortium.org/IIRA-1-7-ajs.pdf>
- [9] Originaltext: „The Industrial Internet is an internet of things, machines, computers and people, enabling intelligent industrial operations using advanced data analytics for transformational business outcomes. It embodies the convergence of the global industrial ecosystem, advanced computing and manufacturing, pervasive sensing and ubiquitous network connectivity.“
- [10] Deutsches IoT-Konsortium *Plattform 4.0*:
<http://www.plattform-i40.de/I40/Navigation/DE/Home/home.html>
- [11] Hightech-Strategie des Bundesministerium für Bildung und Forschung:
<http://www.hightech-strategie.de>
- [12] *Statusreport; Referenzarchitekturmodell Industrie 4.0 (RAMI 4.0)*, hg. von VDI/VDE und ZVEI, April 2015:
<http://www.zvei.org/Downloads/Automation/Statusreport-Referenzmodelle-2015-v10.pdf>
- [13] *Umsetzungsstrategie Industrie 4.0; Ergebnisbericht der Plattform Industrie 4.0*. Hg. v. BITKOM, VDMA und ZVEI, April 2015, S. 40:
<https://www.bmwi.de/BMWi/Redaktion/PDF/I/industrie-40-verbaendeplattform-bericht.pdf>
- [14] Thesenpapier *Cyber-Physical Systems; Chancen und Nutzen aus Sicht der Automation*. Hg. v. VDI/VDE, April 2013. S.2:
https://www.vdi.de/uploads/media/Stellungnahme_Cyber-Physical_Systems.pdf
- [15] Whitepaper von 2015 über Digitalisierung im Auto:
http://mm1.de/fileadmin/content/Whitepaper/mm1_White_Paper_Digitalisierung_im_Auto_9_2015.pdf
- [16] Pressemitteilung *Deutsche Industrie startet Normungsinitiative für Industrie 4.0*, 20. April 2016:
<https://www.plattform-i40.de/I40/Redaktion/DE/Pressemitteilungen/2016/2016-04-21-deutsche-industrie-startet-normungsinitiative-fuer-i40.html>
- [17] Testfelder für Industrie 4.0 *Labs Network Industrie 4.0*, gefördert durch die Bundesregierung und *Plattform Industrie 4.0*:
<http://www.lni40.de>
- [18] Pressemitteilung zum Treffen von IIC und Plattform Industrie 4.0 zum Vergleich der Architekturen:
<http://www.plattform-i40.de/I40/Redaktion/EN/PressReleases/2016/2016-03-02-blog-iic.html>
- [19] Grafik des deutschen Referenzarchitekturmodells *RAMI 4.0* mit kurzer Erklärung:
<http://www.zvei.org/Themen/Industrie40/Seiten/Das-Referenzarchitekturmodell-RAMI-40-und-die-Industrie-40-Komponente.aspx>
- [20] Europäisches Konsortium *IoT-A*, gefördert von der EU:
<http://www.iot-a.eu/public>
- [21] Paper [*Final architectural reference model for the IoT v3.0*] des *IoT-A-Konsortiums*:
www.iot-a.eu/public/public-documents/d1.5/at_download/file
- [22] Die Turing-Award-Gewinner:
http://amturing.acm.org/award_winners

- [23] Europäisches Projekt *BUTLER* (uBiquitous, secUre inTernet-of-things with Location and contEx-awaReness), Laufzeit 2011 bis 2014:
<http://www.iot-butler.eu>
- [24] Projektpräsentation des EU-Projekts *BUTLER*, S. 10:
<http://de.slideshare.net/butler-iot/butler-project-overview-13603599>
- [25] *ZigBee Alliance*:
<http://www.zigbee.org>
- [26] Open-Source-Projekt *Zephyr*: <https://www.zephyrproject.org>
- [27] Chief Economist Marco Annunziata von *General Electric* über das Industrial Internet:
http://www.ted.com/talks/marco_annunziata_welcome_to_the_age_of_the_industrial_internet
- [28] Hartmut Streppel: Angriffsvektor Stromnetz; Buchbesprechung *Blackout*. In: UpTimes 2015-1, S. 34:
<https://www.guug.de/uptimes/2015-1/index.html>

Über Hartmut



Hartmut Streppel ist Spezialist für die Themen Hochverfügbarkeit und Business Continuity. Er arbeitete bis 2014 für die Firma *Oracle*, zu der er durch die Akquisition von *Sun Microsystems* gekommen war. In den 1990-er Jahren war er zwei Jahre Mitglied des Vorstands der GUUG.

„Kleine Schritte, um anzufangen“ Aus der Praxis eines IT-Doku-Dienstleisters

Probleme mit der Erstellung und Ordnung interner Betriebsdokumentation sind so alltäglich wie unerforscht. Die Dokumentationsexpertin **Manuela Reiss** zeigt Einblicke in die Auseinandersetzung mit dieser Art der technischen Dokumentation, deren Besonderheit ist, dass oft nicht ausgebildete Redakteure sie hauptberuflich erstellen, sondern Admins nebenbei im laufenden Betrieb.

Interview: **Anika Kehrer**

Beklage nicht,
was nicht zu ändern ist,
aber ändere,
was zu beklagen ist.

William Shakespeare

Frau Reiss, wie kann man sich Ihre Arbeit vorstellen?

Manuela Reiss: Ich halte offene und Inhouse-Seminare zu allen Aspekten der IT-Dokumentation und unterstütze auch bei der Implementation. Die Inhouse-Schulung ist häufig ein Einstieg für die längere Betreuung beim Kunden. Es hat sich absolut bewährt, damit einzusteigen: Der erste Tag dient dem gemeinsamen Verständnis, und der zweite der Analyse und Maßnahmenentwicklung. Daraus leiten sich Arbeitspakete ab, zum Beispiel für mich, dass ich bei Richtlinien oder Konzeptionen helfe, oder für den Kunden, dass er ein Tool anschafft und einrichtet. Manchmal ist meine Unterstützung sehr umfassend, manchmal auch nur ein, zwei Dokumente, oder ich übernehme das Review der Dokumente.

Zu welchem Anteil sind Sie hinterher auch an der Implementation beteiligt?

Manuela Reiss: Hm. (*Überlegt.*) Der Anteil steigt auf jeden Fall. Früher habe ich den Schwerpunkt auf Wissensvermittlung gelegt, heute bin ich meistens in irgendeiner Form auch danach noch tätig. Das ist dann zum Beispiel Hilfe bei der Erstellung von Konzepten, bei Tool-Evaluierung, bei Template-Erstellung oder Dokumenten-Review für die Qualitätssicherung. Aber auch Hilfe bei der Planung des Dokumentationsprojekts leiste ich. Wir kommen ja aus dem Workshop heraus und haben Arbeitspakete. Es geht dann darum, sie in konkrete Aufgaben herunterzubrechen. Vor allem ist immer wichtig, einen Zeitplan zu erstellen, damit das ganze Projekt nicht irgendwann einschläft.

Ergibt sich da ein bestimmtes Muster? Erst macht man das, dann macht man das ...

Manuela Reiss: Da gleicht wirklich kein Kunde dem anderen. Das Grundproblem ist zwar oft ähn-

lich, aber die Umsetzung ist individuell. Schließlich beginnt keiner auf der grünen Wiese. Die meisten haben bereits viele Dokumente, aber unstrukturiert. Die Kunden haben auch meistens schon Tools im Einsatz, zum Beispiel eine Inventarisierung, ein Ticketsystem oder ein Wiki, und die müssen Teil des Konzepts werden, damit jeder die Informationen erhält, die er benötigt. Außerdem müssen neue Tools dazu passen, damit keine Insellösung entsteht, die schwer zu pflegen ist.

Idealfall: Doku-Verantwortlicher

Wer genau nimmt an den Inhouse-Workshops teil – ist das die ganze Abteilung, oder so etwas wie einzelne Obleute?

Manuela Reiss: Wenn es um operative Einheiten geht, zum Beispiel User-Helpdesk oder IT-Operations, nehmen die Kollegen in der Regel nicht alle am Workshop teil, schon allein deshalb, weil der Betrieb ja weitergehen muss – das ist eine große Hürde. Mein Workshop ist darum auch aufgeteilt. Am ersten Tag geht es um das allgemeine Verständnis für IT-Dokumentation. Da nehmen dann gern ein paar mehr teil, auch der IT-Leiter oder ein Beauftragter. Wenn es dann um die Maßnahmen geht, ist eine kleinere Größe sinnvoller. Hier geht es zunächst darum, konzeptionell zu arbeiten, nämlich sich zu überlegen, was zum Beispiel in die Notfalldokumentation oder in das Inhaltsverzeichnis der Anwendungsdokumentation hinein gehört. Wer teilnimmt, das kommt darauf an, was für Dokumentationsbereiche wir betrachten: Software-Entwicklung, Helpdesk oder IT-Betrieb. Das Konzept für die IT-Notfalldokumentation zu entwickeln macht in großer Runde keinen Sinn, sondern das ist eher Sache der Leitungsebene. Da muss man sich dann im

Nachgang überlegen, wie man das weitervermittelt. Hilfreich sind dann Vorlagen und Einweisungen.

Wer meldet sich bei Ihnen – eher die Admins aus der Praxis oder die Manager aus der Geschäftsführungsebene?

Manuela Reiss: Meistens melden sich bei mir die operativen Einheiten, also Teamleiter oder die IT-Leitung. Top-Manager aus dem Compliance- oder Governance-Bereich melden sich eher selten. Im günstigsten Fall hat das Unternehmen schon einen Verantwortlichen für die IT-Dokumentation, und der ist es dann, der sich bei mir meldet.

Was für Probleme haben diejenigen so, die sich bei Ihnen melden?

Manuela Reiss: Das grundlegende Problem bei fast allen ist, dass die vorhandene IT-Dokumentation gesetzliche, aber auch interne Anforderungen nicht erfüllt. Es gibt zum Beispiel oft keine Notfalldokumentation. Dass man die braucht, leitet sich im Grunde zum Beispiel schon aus dem GmbH-Gesetz ab, wo drin steht, dass die Geschäftsleitung einen ordnungsgemäßen IT-Betrieb sicherstellen und Schaden vom Unternehmen abwenden muss. Bei der konkreten Problemstellung kommt es auch darauf an, wie stark das Unternehmen bereits prozess- oder auch serviceorientiert arbeitet. Die IT-Organisationen in Banken oder Konzernen sind in der Regel serviceorientiert aufgestellt – sie haben dann einen hohen Reifegrad, wenn es um Fragen der Dokumentation geht. Bei Banken geht es oft um die Anforderungen der BAFIN. Das ist eine andere Fragestellung, als wenn ein Unternehmen seine IT-Dokumentation erst einmal aufbauen muss.

Gibt es typische Eigenschaften der Organisationen, die sich bei Ihnen melden?

Manuela Reiss: Eher nicht. Die Anfragen sind branchenunabhängig und größenunabhängig. Sowohl Behörden als auch Industrieunternehmen, kleine Ämter genauso wie Konzerne melden sich.

ITSM schiebt Doku an

Hängt eigentlich Ihrer Beobachtung nach die Beschäftigung mit Organisationsfragen – etwa ITSM – und die Inangriffnahme der Doku-Situation zusammen?

Manuela Reiss: Ich beobachte weniger, dass die Dokumentation Treiber für bessere Organisation ist. Andersherum ist IT-Service-Management oder

Prozessmanagement ohne Dokumentation nicht möglich. Man beschäftigt sich dann automatisch intensiver damit. ITSM heißt ja zum Beispiel, die SLAs aufzuschreiben, oder einen Service-Katalog zu haben. Und es geht dann auch um Prozesse, die die Pflege der Dokumentation unterstützen. Habe ich einen Change-Prozess, an dessen Ende die Aktualisierung der Dokumentation steht, bleibt diese auch aktuell.

Stellen Sie Unterschiede fest, wie große und kleine Teams ihre Dokumentation handhaben?

Manuela Reiss: Je kleiner das Team ist, desto weniger wird in der Regel dokumentiert. Denn hier herrscht die Einstellung: „Ich mache es ja, für wen sollte ich es aufschreiben?“ Außerdem bleibt den Mitarbeitern in einem kleinen Team oft weniger Zeit dafür. Und in dem Moment, wo es um Ablaufbeschreibungen geht, wird die Dokumentation zwangsläufig aufwändig. Von einem Team mit zwei, drei Mitarbeitern eine umfassende, von ihnen selbst erstellte Sicherheits- und Notfalldokumentation zu erwarten, ist illusorisch. Allerdings hilft das leider nichts – die Verpflichtung ist ja da. Im Grunde ist man dann bei *Make or Buy*. Was man nicht selbst machen kann, muss man eben einkaufen.



Abbildung 1: Manuela Reiss und ihr Ehemann Georg, der die IT-Revision eines Verkehrs- und Energieversorgungskonzerns leitet, haben ihre Kompetenz in dem erstmals 2008 von Addison Wesley verlegten Buch über IT-Dokumentation in der Praxis gebündelt (seit 2014 wird das Buch vom Hanser Verlag verlegt). Die erweiterte und aktualisierte Auflage erschien im Januar 2016. Es ist bis heute das einzige seiner Art auf dem Markt.

Gibt es irgendwelche Standards, an denen man sich orientieren kann?

Manuela Reiss: Nein. Anders als etwa für die Software-Entwicklungsdokumentation gibt es für die betriebliche IT-Dokumentation keine übergeordneten Standards oder Vereinbarungen. Wenn überhaupt, dann haben Firmen sich selbst welche geschaffen. Es gibt zwar vereinzelte Anforderungskataloge, etwa die *GoBD* zur Verfahrensdokumentation [2] oder Prüfungsstandards wie der *IDW PS 330* [3], der Wirtschaftsprüfern als Leitfa-den zur Prüfung rechnungslegungsrelevanter IT-Systeme dient. Aber das ist reine Compliance-Sicht. Das ist nichts, woraus ich Konkretes für meine IT-Dokumentation ableiten kann. Als Admin habe ich keine Regeln, die mir sagen, was ich hinschreiben muss in eine Betriebsdokumentation, in Arbeitsanleitungen oder ein Backup-Konzept. Genau das war damals die Motivation für unser Buch (Abbildung 1, [1]). Damit wollten wir zumindest eine Richtschnur geben.

Management unter Doku-Druck

Sie beschäftigen sich gezielt mit IT-Betriebsdokumentation jetzt seit rund 15 Jahren. Was ist seit damals gleich geblieben?

Manuela Reiss: Schon vor 15 Jahren war es so, dass Dokumente zwar erstellt, aber dann nicht gepflegt wurden. Die Nachhaltigkeit ist damals wie heute ein Problem.

Und gibt es etwas, das sich verändert hat?

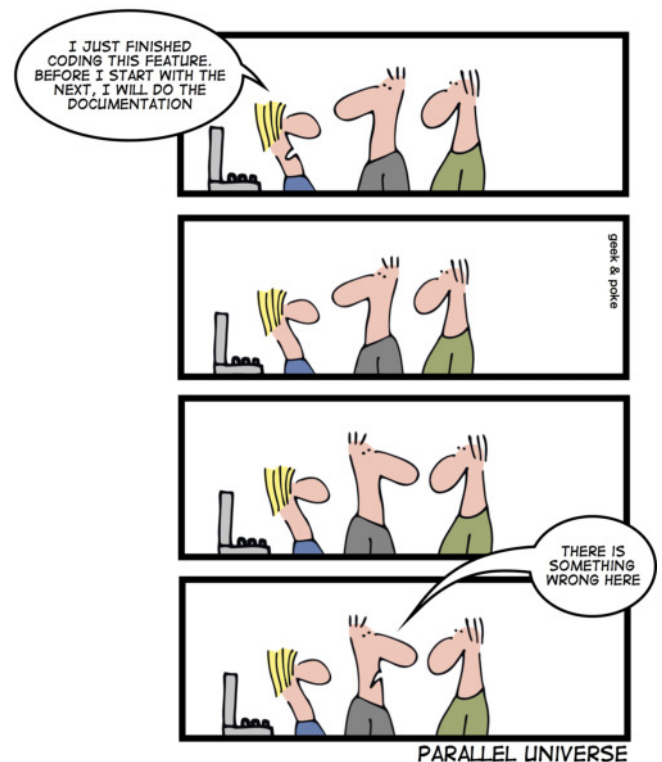
Manuela Reiss: Damals hat sich das Management weniger um die operative Dokumentation gekümmert. Heute steht das Management unter höherem Nachweisdruck, auch wegen der gestiegenen Abhängigkeit der Geschäftstätigkeiten von der IT. Heute nehmen zum Beispiel Wirtschaftsprüfungen die IT in den Fokus, was früher nicht so war. Heute hat das Management also sehr viel mehr Compliance-Anforderungen zu stemmen und muss mittels Dokumentation Nachweise über die Ordnungsmäßigkeit seines IT-Betriebes erbringen. Der Einstieg für Unternehmen, sich ernsthaft mit ihrer Dokumentation zu beschäftigen, ist zum Beispiel oft ein Audit, bei dem Dokumentationsmängel offensichtlich wurden, oder eine angestrebte Zertifizierung.

Das sind alles extrinsische Motivationen, also Anschubser, die von außen kommen. Sind die Anfragen, die Sie erhalten, manchmal auch int-

rinsisch motiviert, also von den Leuten selbst gewünscht?

Manuela Reiss: Eine rein intrinsische Motivation ist bei den Anfragen an mich eher selten. Meistens wird die IT-Dokumentation als Muss und als notwendiges Übel betrachtet, eben auch gerade weil sie von oben verordnet wird. Es sind eher Compliance-getriebene Themen, die dazu führen, dass die IT-Dokumentation angefasst wird. Dass eine IT-Organisation sagt: „Wir wollen von uns aus für uns selbst unsere Dokumentation verbessern“, das kommt zwar vor, aber nur sehr selten.

SIMPLY EXPLAINED



Nebenbei-Doku funktioniert nicht

Was ist Ihrer Meinung nach die allererste Voraussetzung, um eine nützliche IT-Betriebsdokumentation zu haben?

Manuela Reiss: Als Erstes ist das für mich das Verständnis und das Commitment des Managements, dass also die Geschäftsführung dahinter steht. Da sind wir dann nämlich beim Thema Zeit: Dokumentation erfordert Zeit und Ressourcen. Häufig wird aber erwartet, dass die Dokumentation nebenbei erstellt wird. Und das kann nicht funktionieren. Die Dokumentation muss sowohl bei der Planung des eigenen Betriebs, als auch bei Aufträgen an Dritte kalkuliert werden, sie muss zum Beispiel mitbeauftragt werden. Die zweite Voraussetzung ist die Einführung von Regeln für die

Dokumentation. Das heißt zum Beispiel, Verantwortlichkeiten festzulegen, Dokumentationsverfahren aufzustellen, Qualitätsmanagement für Dokumente und ein Änderungsmanagement einzuführen, damit die Dokumente aktuell bleiben.

Was sind Ihrer Erfahrung nach die Hauptgründe dafür, dass eine IT-Dokumentation nicht funktioniert?

Manuela Reiss: Zum einen natürlich, dass die genannten Voraussetzungen eben nicht vorliegen. Viel Dokumentation wird zum anderen in IT-Projekten erstellt, also von einem externen Dienstleister. Die ist dann nicht eingebunden in ein Gesamtkonzept. Wichtig wäre hier eine geregelte Übergabe an den IT-Betrieb, also Festlegungen, was der Dienstleister an Dokumentation liefern muss und dass man daran vielleicht sogar die Abnahme koppelt. Ein weiteres Problem ist, dass die Dokumentation auf verschiedene Tools verteilt ist. Zum Beispiel im Ticketsystem, ein Wiki, ein Inventarisierungs-Tool und natürlich Dokumente im File-System. Bei meinen Kunden versuche ich dann immer, das führende Tool zu identifizieren und die Dokumentation an ein oder zwei Stellen zusammenzuführen.

Was denken Sie, wenn jemand sagt: „Ich habe einfach keine Zeit, mich um Doku zu kümmern“?

Manuela Reiss: Realistischerweise haben viele Admins kaum Freiraum für die Dokumentation. Gerade dann ist es wichtig, die Zeit dafür gut zu nutzen. Das funktioniert nur, wenn es Prioritäten und klare Vorgaben gibt. Wenn jeder weiß, wie, wo und was er dokumentieren muss, wird es meistens einfacher. Häufig empfinden aber Unternehmen die Doku auch als einen Riesenberg,

bei dem sie nicht wissen, wie sie anfangen können. Und dann fangen sie gar nicht erst an. Ich gehe dann so vor, kleine Arbeitspakete zu packen und zu priorisieren – kleine Schritte, um anzufangen. Zum Beispiel: Wie kann ich die Systemdokumentation mit Automatisierung erleichtern, also inventarisieren? Was ist wichtig, zum Beispiel Sicherheit und Backup? Ich muss nicht als Erstes alle Abläufe im Betrieb beschreiben. Sondern ich muss kleine Päckchen packen, die über einen langen Zeitraum stemmbar sind. Die größte Hürde ist wirklich, dass ein Berg gesehen wird, der unüberwindlich ist. Dann fängt keiner an.

Zum Abschluss: Geben Sie bitte allen Admins da draußen eine Handreichung mit auf den Weg.

Manuela Reiss: Kurz gesagt besteht die Dokumentation für den täglichen IT-Betrieb aus den beiden Bereichen Was und Wie, nämlich welche Systeme habe ich im Einsatz, und wie muss ich diese administrieren. Das Erste ist: **Automatisieren** Sie die Systemdokumentation so weit wie möglich. Dokumente müssen ja nicht nur Prosa-Dokumente sein. Auch ein Datenblatt oder ein Bericht ist ein Dokument. Wichtig ist dann nur, dass es an passender Stelle abgelegt oder einem übergeordneten Dokument zugeordnet wird, damit die Zuordnung nicht verloren geht. Zweitens: Wenn es um die Dokumentation des Wie, also um Abläufe geht, sind **Vorlagen** hilfreich. Setzen Sie sich gemeinsam hin und überlegen Sie, wie der tägliche Betrieb der Systeme beschrieben werden soll: Wer tut wann was in welcher Form in welchem System in welcher Reihenfolge? Und wieder muss es nicht immer Prosa sein. Auch Checklisten und grafische Abläufe sind nützlich. Drittens: Definieren Sie **Regeln** für die Dokumentation und halten Sie diese auch ein.

Links

[1] Manuela und Georg Reiss: Praxisbuch IT-Dokumentation; Vom Betriebshandbuch bis zum Dokumentationsmanagement – die Dokumentation im Griff. München, Hanser 2016 (2., aktualisierte und erweiterte Auflage): <http://www.hanser-fachbuch.de/buch/Praxisbuch+IT+Dokumentation/9783446445994>

[2] GoBD = Grundsätze zur ordnungsmäßigen Führung und Aufbewahrung von Büchern, Aufzeichnungen und Unterlagen in elektronischer Form sowie zum Datenzugriff.

http://www.bundesfinanzministerium.de/Content/DE/Downloads/BMF_Schreiben/Weitere_Steuerthemen/Abgabenordnung/Datenzugriff_GDPdU/2014-11-14-GoBD.html

[3] IDW PS 330 = IDW Prüfungsstandard: Abschlussprüfung bei Einsatz von Informationstechnologie, herausgegeben vom Institut der Wirtschaftsprüfer in Deutschland e.V. (IDW).

<http://www.kompass-sicherheitsstandards.de/43722.aspx>

Über Manuela Reiss



Manuela Reiss unterstützt Unternehmen bei der Konzeption, Analyse, Erstellung und Qualitätssicherung ihrer IT-Dokumentation. Die studierte Geografin machte sich als Trainerin und Autorin 1993 selbständig (<http://www.manuela-reiss.de>). Später kamen Consulting-Tätigkeiten dazu. Bei ihren Tätigkeiten in IT-Projekten und im operativen IT-Betrieb hat sie immer wieder gesehen, dass die Dokumentation in vielen IT-Betrieben ein großes Problem darstellt. Ihr Blog zum Thema heißt <http://itdoku-kompakt.de>

Was lange währt ... Markup für Entwicklungs-Dokumentation

Gründe für das gering ausgeprägte Bedürfnis Software zu dokumentieren gibt es viele. Dieser Artikel konzentriert sich auf die damit verbundenen Arbeitsabläufe und Werkzeuge – wie sie waren, wie sie heute sind und warum es heute sogar Spaß machen kann, Dokumentation zu erstellen.

von Patrick Koetter

Der Ausgang krönt das Vollbrachte.

Ovid

Software ist ein Produkt. Ein Produkt ohne Dokumentation ist unvollständig. Das ist nichts Neues und auch die Tatsache, dass Entwickler – Ausnahmen ausgenommen – Dokumentation scheuen wie der Teufel das Weihwasser, überrascht sicher niemanden, der in der IT tätig ist.

Aber Dokumentation ist wie Code: eine Ausführungsanweisung. Sie wird geschrieben, um zum Beispiel die Anwender zu mobilisieren. Mit Hilfe der Dokumentation sollen diese in der Lage sein, Software erfolgreich anzuwenden. Das dafür erforderliche Wissen muss dafür erfasst, niedergeschrieben, in Form gebracht und mediengerecht verteilt werden.

Guten Code schreiben ist bekanntlich schwierig. Es braucht beispielsweise eine Architektur, die der Aufgabe gewachsen ist. Der Code muss auch seine Funktionen eindeutig und fehlerfrei erfüllen. Er muss auf den Punkt kommen, damit er schnell und effizient ist. Redundanzen sind dafür zu vermeiden. Kurz: Wer programmiert und dokumentiert, weiß, wie ähnlich und fordernd beide Domänen sind.

Für die Programmierung haben Programmierer sich lange schon Abläufe und Werkzeuge entwickelt, die sie bestmöglich unterstützen. Sie standardisieren, versionieren und automatisieren wo immer möglich, damit sie sich auf die eigentliche Arbeit konzentrieren können – das Schreiben guten Codes. In der Welt der Dokumentation war das jedoch lange nicht so.

Broken Workflow

Ein typischer Workflow für Print-Dokumentation sah lange in etwa so aus: Der Autor, meist der Software-Entwickler selbst, musste seinen Text in einer Textverarbeitung verfassen, weil die Designer damit am besten umgehen konnten. In dem

Dokument musste er nicht nur Inhalt und Struktur entwickeln und kenntlich machen. Er musste auch durch korrekte Formatierung optisch kenntlich machen, welche Worte Code, Kommandos, Kommando-Optionen und ähnliche unterschiedliche Worttypen repräsentieren.

Ein Gestalter übernahm dann das Dokument, kopierte den Inhalt und fügte ihn in ein *Adobe InDesign*- oder *QuarkXPress*-Dokument ein. Dann setzte er Absatz für Absatz, Wort für Wort von Hand. Das Ergebnis wanderte dann als PDF an den Autor zurück, mit der Aufforderung zur Korrektur. Korrekturen landeten als Kommentare in der PDF-Datei.

Auf diese Weise ging das Dokument solange hin und her, bis es fertig war. Stünden größere Veränderungen an, wurden sie entweder Stück für Stück von Hand eingefügt, oder das Dokument gleich von neuem begonnen. Nur in Ausnahmefällen gehörte es zum Workflow, das Quelldokument in einem SGML- oder XML-Schema zu verfassen und dann nahezu automatisch etwa mit *Adobe Illustrator* auf Grundlage vorgefertigter Transformationsregeln zu setzen.

Dokumentation wird Code

Diese Art der Dokumentation war aufwendig, fehleranfällig und ineffizient. Die Autoren mussten nicht nur die zentrale Aufgabe meistern, inhaltlich sinnvolle Dokumentation zu schreiben. Sie mussten den Inhalt auch gestalterisch korrekt für die Designer vorgeben. Damit waren sie gefordert, in einer Wissensdomäne zu arbeiten, für die sie nicht ausgebildet waren. Fehler und Missverständnisse waren die Folge.

Die Abkehr von Binär-Formaten und die Zuwendung hin zu Plaintext markierte einen wichtigen Wendepunkt. Jetzt wurde es möglich, Doku-

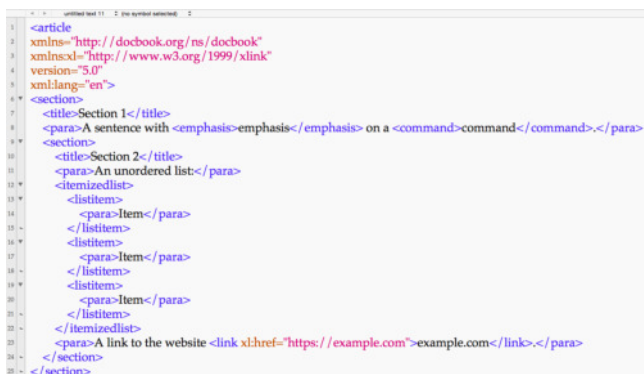
mentation wie Programmcode zu behandeln. Der Text wurde versioniert, verteilt bearbeitet, Änderungen miteinander verknüpft und mit *diff* auf Änderungen untersucht.

Die wichtigste Änderung war jedoch die Abkehr von einem Text, der ein bestimmtes Aussehen trägt, hin zu einem Text, der eine bestimmte Bedeutung trägt. Autoren formatierten ein Kommando nun nicht mehr als Zeichenfolge in – zum Beispiel – *Monotype, regular, 12pt*, sondern sie zeichneten die Zeichenfolge stattdessen einfach als *command* aus. Das fertige Dokument mit seinen Auszeichnungen übergaben sie an die Designer. Und diese wiesen den Auszeichnungen dann Formatierungen zu.

DocBook DTD

Möglich wurde dies durch die Erfindung von *DocBook* Anfang der 1990er Jahre, einer *Document Type Definition* (DTD) aus dem Umfeld des O'Reilly-Verlages. Ihr Erfinder Norman Walsh entwickelte mit *DocBook* eine format-agnostische Auszeichnungssprache, die ein Vokabular (Elemente) speziell für technische Dokumentation zur Verfügung stellt. Mit ihr war es möglich, Zeichen, Worten und Absätzen Eigenschaften wie *Kommando*, *Datei* oder *Code-Beispiel* zuzuweisen.

Anfänglich noch in reinem SGML formuliert, folgte bald ein XML-Schema, mit dem der Quelltext von *DocBook*-Dokumenten ausgezeichnet wird (Abbildung 1). Von diesem Rohformat aus wird der Text in verschiedenste Zielformate überführt. Die Regeln des Zielformates stehen in einem XSL-Stylesheet. Mit Hilfe eines XSLT-Prozessors wird das XML-Dokument dann unter Einhaltung der Regeln des XSL-Stylesheets transformiert.



```

1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!DOCTYPE article SYSTEM "http://docbook.org/ns/docbook"
3 <?xml:base href="http://www.w3.org/1999/xhtml" ?>
4 <?xml:lang="en" ?>
5 <article>
6   <section>
7     <title>Section 1</title>
8     <para>A sentence with <emphasis>emphasis</emphasis> on a <command>command</command>.</para>
9   </section>
10  <section>
11    <title>Section 2</title>
12    <para>An unordered list:</para>
13    <list>
14      <listitem>
15        <para>Item</para>
16      </listitem>
17      <listitem>
18        <para>Item</para>
19      </listitem>
20      <listitem>
21        <para>Item</para>
22      </listitem>
23    </list>
24    <para>A link to the website <link href="https://example.com">example.com</link>.</para>
25  </section>
26 </article>

```

Abbildung 1: Beispiel für das Markup von *DocBook*.

Für eine Man-Page (*troff*, ASCII-Text) genügte eine einfache Transformation. Für PDF, RTF und

andere Formate nutzte das Dokument ein Zwischenformat, ein *FO-Dokument* (Formatting Objects). Ein *FOP* (FO-Prozessor) wandelte es von dort aus in das endgültige Zielformat um.

Dann stagnierte die Entwicklung eine Weile: Das Handling der Quelltexte war einfacher, und die Produktion von Zieldokumenten automatisierbar geworden. Alles war gut.

Dokumentation 2.x

Mit der Ankunft agiler Methoden stieg der Leidensdruck wieder. Dem Motto *release early, release often* konnte die Dokumentation nicht im erforderlichen Tempo folgen und bremste die neue, agile Arbeitsweise aus. Die Kritik am bestehenden Modell konzentrierte sich auf drei Aspekte der *DocBook DTD*: Komplexität, Konvention und Werkzeug.

Der Segen der DTD – ihre Fähigkeiten und ihr Umfang – ist zugleich ihr Fluch. Wer mit *DocBook* arbeitet, muss sich auf ein komplexes Regelwerk einstellen, das es über längere Zeit zu erlernen gilt. Als XML-Sprache ist sie zudem strukturorientiert und erwartet von Anfang an eine strukturierte Vorgehensweise. Damit setzt sie voraus, dass der Autor bereits beim Entwickeln des Textes eine klare Vorstellung von Inhalt und Struktur hat.

Genau das ist aber so gut wie nie der Fall. Wer sich selbst einmal beim Schreiben beobachtet, sieht, dass Form und Inhalt zu Beginn nicht feststehen. Wesentlicher Teil der Schreibarbeit ist, Form und Inhalt iterativ zu erarbeiten und in eine der Argumentation dienende Abfolge zu bringen. Die Konvention der DTD behindert also die Freiheit, die ein Autor zu Beginn benötigt.

Und schließlich waren die Autoren auf einige wenige Editoren festgelegt, wenn sie XML nicht nativ in einem Texteditor bearbeiten wollten. Gerade ihre textorientierten Editoren wollten die Entwickler aber nicht verlassen.

reStructuredText

reStructuredText (kurz: *rst*) ist das Hausformat der Python-Community (Abbildung 2): Mitglieder der Community sind angehalten, APIs, Code, Manuals und so weiter in diesem Format zu dokumentieren. Das populäre Buildsystem *Sphinx* geht nativ mit *rst* um und generiert daraus HTML. *Sphinx* lässt sich sogar in automatisierte Buildprozesse einbinden (Continuous Integration). Das Paket *Pygments* sorgt für praktisches Highlighting von Source-Code-Beispielen.

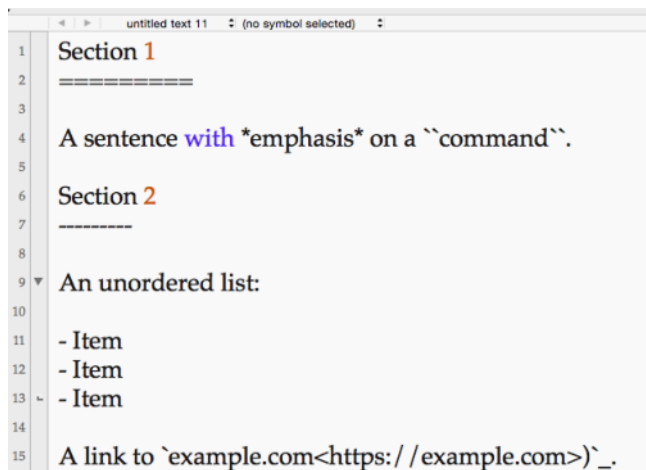


Abbildung 2: Beispiel für das Markup von *reStructuredText*.

Aufgrund der engen Verknüpfung mit Python ist *rst* auch ähnlich populär. Außerhalb der Python-Community ist es gefühlt genauso verbreitet wie das Versionsverwaltungssystem *bazaar*. Es fungierte als VCS der Python Community, bevor diese letztlich doch auf *git* zur Sourcecode-Versionierung einschwenkte.

Wer seinen *rst*-Quelltext in andere Zielformate umwandeln möchte, kann auf eine Werkzeugsammlung von Kommandos zurückgreifen, die alle mit *rst2* beginnen. Sie stellen die gängigsten Formate *man* (troff), HTML, XML, $\text{\LaTeX}$ und PDF zur Verfügung. Speziell sind Ausgaben für ODT- und S5-(X)HTML-Präsentationen.

Markdown

Markdown ist die zur Zeit am weitesten verbreitete Variante einer so genannten *Low Markup Language* (LML). Seine Popularität hat es den Machern von *GitHub* zu verdanken, die es zum Standardformat etwa für die Readmes in ihrem VCS-Portal erhoben haben. Es vereint einfaches Markup mit den am häufigsten verwendeten Strukturelementen und inhaltlichen Auszeichnungen (Abbildung 3). Überschriften, Absätze, Listen, Fettung und Kursivsatz beherrscht es, und auch Tabellen sowie Bilder baut es angenehm einfach ein.

Die in Markdown zu Verfügung stehenden Sprachelemente sind allesamt generisch. Sie lassen sich auf beliebige Textformen anwenden, sei es Brief, Dokumentation, Artikel oder Blog. Das ist – nach Auffassung des Autors – zugleich seine größte Schwäche.

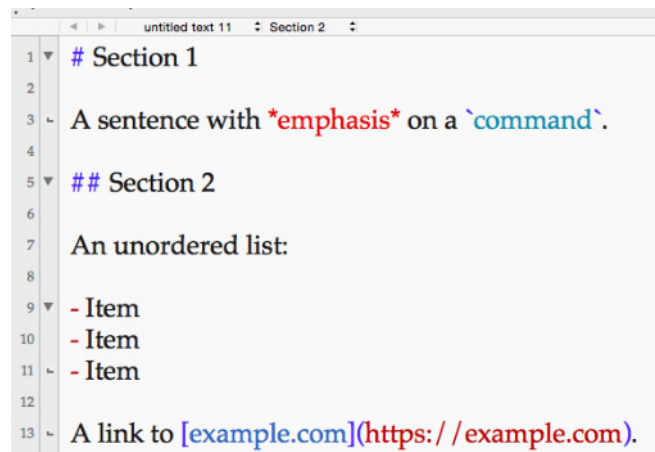


Abbildung 3: Beispiel für das Markup von *Markdown*.

AsciiDoc

In der Welt der Dokumentation und besonders in der technischen Software-Dokumentation existieren besondere Auszeichnungsformen, die den Leser auf beachtenswerte Inhalte hinweisen. Gemeint sind Passagen, die mit Icons für *Info*, *Wichtig*, *Vorsicht* oder *Gefahr* im Textfluss besonders hervorgehoben werden. Die besondere Gestaltung dient als visueller Anker, der dem Leser helfen soll, den Gesamttext im Überblick schnell zu erfassen.

Derartige Passagen und noch viele andere Sprachelemente beherrscht *AsciiDoc* (Abbildung 4). Unter den heute gebräuchlichen Low Markup Languages ist *AsciiDoc* das eloquenteste. Es weiß am präzisesten zu beschreiben, was einen Text alles auszeichnen kann. Seine Beredsamkeit kommt nicht von ungefähr: Es steht auf den Schultern eines alten Bekannten – *AsciiDoc* ist eine LML-Adaption von *DocBook*.

Wer *AsciiDoc*-Quelltext in andere Zielformate umwandeln will, kann auf zwei Kommandozeilenprogramme zurückgreifen: *asciidoc* und *a2x*. Das erste generiert HTML nach verschiedenen Standards entweder in eine einzelne Ausgabedatei, oder nach Kapiteln aufgeteilt in mehrere Dateien. Das zweite kennt unterschiedlichste Ausgabeformate: Von RTF über PDF und Man-Pages (*troff*-Format) bis hin zu E-Pub bedient es auch gehobene Ansprüche.

a2x bedient sich je nach Zielformat unterschiedlicher Werkzeuge. Die Schnittstelle ist generisch gehalten, man ist also nicht auf ein bestimmtes Werkzeug pro Zielformat festgelegt. Zur PDF-Ausgabe können beispielsweise entweder *Apache FOP* (Open Source), kommerzielle Renderer oder auch $\text{\LaTeX}$ (per *dblatex*) eingebunden werden.

Möglich wird das bei PDF, weil a2x den AsciiDoc-Quelltext zuerst in das XML-Format von DocBook umwandelt und dann – mit Hilfe von DocBook-XSL – eine FO-Datei erstellt. Von dort aus kann die .fo-Datei dann an beliebige FOPs übergeben werden, die aus der FO-Datei ein PDF rendern.

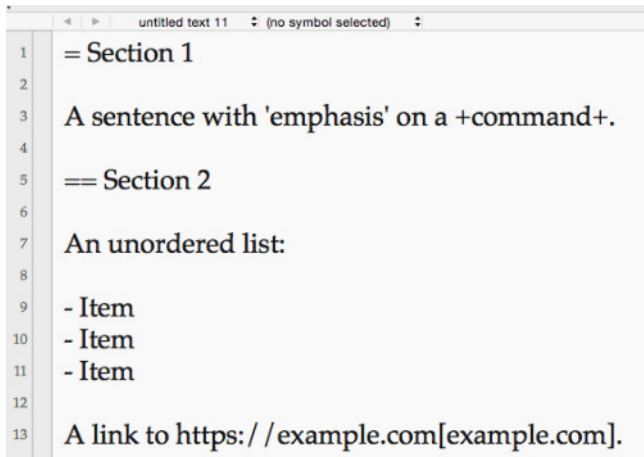


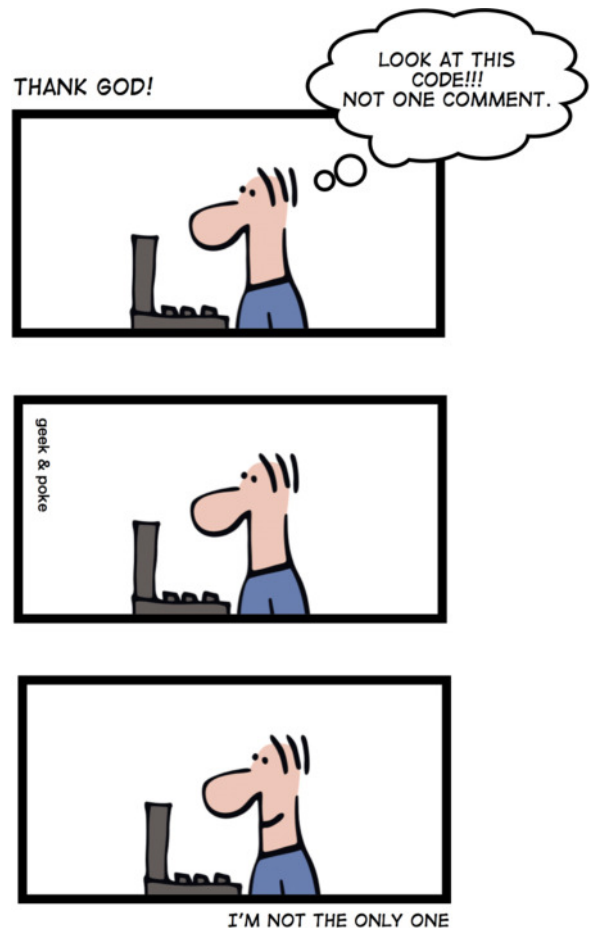
Abbildung 4: Beispiel für das Markup von *AsciiDoc*.

Klingt kompliziert? Ist es anfänglich auch. Die Lernkurve bei AsciiDoc ist flach. Es dauert eine Weile, bis man zielsicher das richtige Werkzeug in der Kette mit passenden Optionen versieht, um die PDF-Ausgabe den eigenen Vorstellungen anzupassen. Dafür steht am Ende des Build-Prozesses aber auch ein Dokument mit Titelseite, Inhalts-, Tabellen- und Bildverzeichnis, Index, Glossar und allem, was man sich wünschen kann. Oben drauf kann auch gern ein hauseigenen Font kommen – eingebettet in das PDF-Dokument.

Buildchain

Wer seine Buildchain auf LML umgestellt hat, will in der Regel noch weiter automatisieren, indem er die Dokumentation zentral verteilt oder zum Beispiel Abonnenten der Dokumentation bei Aktualisierungen automatisch benachrichtigt.

Hier bieten sich je nach Markup-Sprache Portale wie <https://readthedocs.org> oder <https://www.gitbook.com> an. Letzteres integriert sich in das eigene Git-Ökosystem. Nach einem Push ins Repository löst Git einen Webhook im passenden Portal aus. Das aktualisiert per Pull-Befehl die lokale Kopie des Repositories und stößt anschließend einen automatischen Build-Prozess für die Dokumentation an. Je nach Lizenzmodell – frei oder MietSERVICE – benachrichtigen die Portale im Anschluss die Abonnenten der Dokumentation, oder sie stellen den Abonnenten die Dokumentation direkt per E-Mail zu.



Beyond the Docs

Der Ansatz von Low Markup Languages eignet sich auch für Blogs oder Websites. Die Inhalte werden lokal mit einem Editor erstellt und bearbeitet. Je nach Website-Generator wird dann lokal eine Website gebaut, oder die Daten wandern zuerst auf den veröffentlichenden Server, der sie zur einer Website oder einem Blog kompiliert. Wer Kommentare in sein statisches Blog einbinden will, kann dazu die Online-Dienste *Disqus* oder *Isso* nutzen.

Markupsprachen haben einen Wandel hin zur Reduktion vollzogen. Einfache und schnelle Bearbeitung stehen im Vordergrund. Jeder der genannten LML-Dialekte hat seine ganz eigenen Stärken. Das Notwendige beherrschen sie alle. Eine Lösung, die auf einfache Weise alle Aspekte abdeckt, existiert noch nicht.

Zeitgleich hat sich eine Community formiert, die Dokumentation zum Thema hat. Sie ist offen und widmet sich den unterschiedlichsten Facetten wie Tools, Strategien oder Standarddokumente. Nächstes großes Treffen der Community ist vom 18. bis 20. September die Konferenz *Write the Docs* in Prag.

Über Patrick



Patrick Koetter ist heute CEO der *sys4* AG. Dort leitet er die Geschicke der AG und arbeitet als E-Mail-Experte für ISPs und Unternehmen, die mehr brauchen, als nur E-Mail von der Stange. Begonnen hat seine Liebe zu E-Mail, als er seinen ersten eigenen Server einrichtete und dabei auf *Sendmail* stieß. Das wäre fast das Ende gewesen, wenn er nicht auch noch *Postfix* entdeckt hätte. Danach führte ein Howto über SMTP-Authentifizierung zu dem gemeinsam mit Ralf Hildebrandt geschriebenen *The Book of Postfix*. Das Buch wurde Postfix-Weltbestseller. Patrick engagiert sich weiterhin in vielen Open-Source-Projekten mit E-Mail-Bezug wie *amavisd-new*, *opendkim* und *opendmarc*. Zuletzt lud er nach einem ungewöhnlich fordernden Projekt, das er mit einem eigens zusammengestellten Dreamteam ins Ziel gebracht hatte, dasselbe Team zu einer Firmenidee ein. Alle kamen – und sie gründeten *sys4*.

Einfacher als man denkt Über das Schreiben von Fachtexten

Dieser Artikel gibt schreibenden Technikern Rüstzeug an die Hand, das Schreiben als weniger schwer zu erleben. Er versammelt Argumente für das Erstellen von Fachaufsätzen sowie praktische Hinweise, damit die Ergebnisse auch dem Leser Freude machen.

von Jürgen Plate und Anika Kehrer

Wer will, dass ihm die anderen sagen,
was sie wissen, der muss ihnen sagen,
was er selbst weiß.
Das beste Mittel, Informationen zu erhalten,
ist Informationen zu geben.

Niccolò Machiavelli

Für einige Techniker ist die Aufgabe, einen Text zu schreiben, mehr Qual als Freude. Solche Techniker mögen argumentieren, dass ihre Befähigung zum Beispiel in der Entwicklung elektronischer Schaltungen oder im Programmieren liegt, und nicht in literarischen Ergüssen. Ihre Schaffenskraft werde also besser mit der Entwicklung von etwas Neuem genutzt, als mit dem Schreiben von etwas Früherem, muffeln sie.

Diese Auffassung ist verständlich, würdigt aber nicht den Wert technischer Berichte. Andererseits erklärt sie das Problem, dass Software-Entwickler oft auf Kriegsfuß mit Dokumentation stehen, oder dass Handbücher gelegentlich zu Schnappatmung beim Leser führen. Dieser Beitrag versammelt daher Argumente für das freudigere Erstellen von Fachaufsätzen sowie praktische Hinweise, damit sie auch dem Leser hinterher mehr Freude machen.

Stilfragen

Sachliteratur erfüllt einen nützlichen Zweck – nämlich möglichst klare und präzise technische Information zu vermitteln. Der Inhalt ist das Entscheidende. In der Weltliteratur ist dagegen die Ausdrucksweise das Entscheidende. Zum Beispiel könnte der Inhalt von *Macbeth* in wenigen Seiten wiedergegeben werden, aber ohne Shakespeares Stil wäre es nur die halbe Miete. Solche Literatur ist zur Unterhaltung geschrieben; Fachaufsätze dagegen zur Information.

Obwohl also in der technischen Literatur der sachliche Inhalt im Vordergrund steht, ist der Stil nicht völlig unwichtig. Von ihm hängen die Lesbarkeit und damit die Verständlichkeit des technischen Inhaltes ab. Wer kurzweilig schreibt, begeistert; aber das ist nicht der Primärzweck von Fachtexten und absolut nicht die Anforderung an

Techniker. Es ist also nicht das Ziel, einen schönen Text zu schreiben. Aber ein klarer Text ist das Ziel. Für den Leser bedeutet ein klarer Text, dass er den Inhalt leichter aufnehmen kann. Selbst das ist – zugegeben – nicht trivial. Aber es gibt Grundregeln, die helfen. Zusätzlich zu den folgenden Erläuterungen hilft der Kasten *Aushang: Typische Fallen und ihre Alternativen*, die häufigsten Fallen zu meiden.

Inhaltsfragen

Eine zweite Schwierigkeit für Techniker ist die Definition des Inhalts. Ein Ingenieur, der einen Verstärker bauen soll, kann nicht anfangen, solange er nicht weiß, was für ein Verstärker genau gebraucht wird. Er muss zum Beispiel wissen, wie groß Verstärkung, Ausgangsleistung, Lastwiderstand, Frequenzbereich und zulässiger Klirrfaktor sein sollen. Eine solche Definition ist auch beim Erstellen von Fachaufsätzen nicht trivial. Beispielsweise steht manchmal eine Frage am Anfang, der man nachgehen möchte. Und wie soll man dann die Grenzen der Antwort definieren, wenn man sie noch nicht kennt?

Eine gewisse Spezifikation ist aber beim Schreiben von Fachtexten genauso wichtig, wie beim Ersinnen eines technischen Gerätes. Sonst tritt schnell ein, was Autoren als Phänomen des Verzettels kennen: Man kommt vom eigentlichen Thema ab. Das erschwert nicht nur das Schreiben, sondern verwirrt am Ende auch den Leser. Es gilt also, einen Rahmen zu stecken, was der Text können soll, und was er nicht können braucht. Entgegen den Gepflogenheiten in wissenschaftlicher Literatur gehört die Abgrenzung, was der Text nicht leistet, allerdings nicht in den Text hinein. Sondern die Abgrenzung ist zunächst nur für den Autoren da

und bewirkt, dass er in den Text nur solche Inhalte aufnimmt, die keine Fragen offen lassen.

Wer nur für sich selbst schreibt, dem fehlt manchmal die Außensicht, um Grenzen zu erkennen. Hier können Kollegen als Sparringpartner dienen, um abzuklopfen, was im Text stehen sollte und was nicht. Wer aber für andere schreibt – zum Beispiel für die Leser eines bestimmten Mediums – sollte sich nicht scheuen, bei der entsprechenden Redaktion Briefinggespräche oder die Besprechung von Zwischenständen zu fordern. Denn nichts ist frustrierender, als wenn ein Text mühsam fertig gestellt ist, nur um hinterher zu erfahren, dass etwas aber doch lieber so oder so gemacht werden sollte.

Zweckfragen

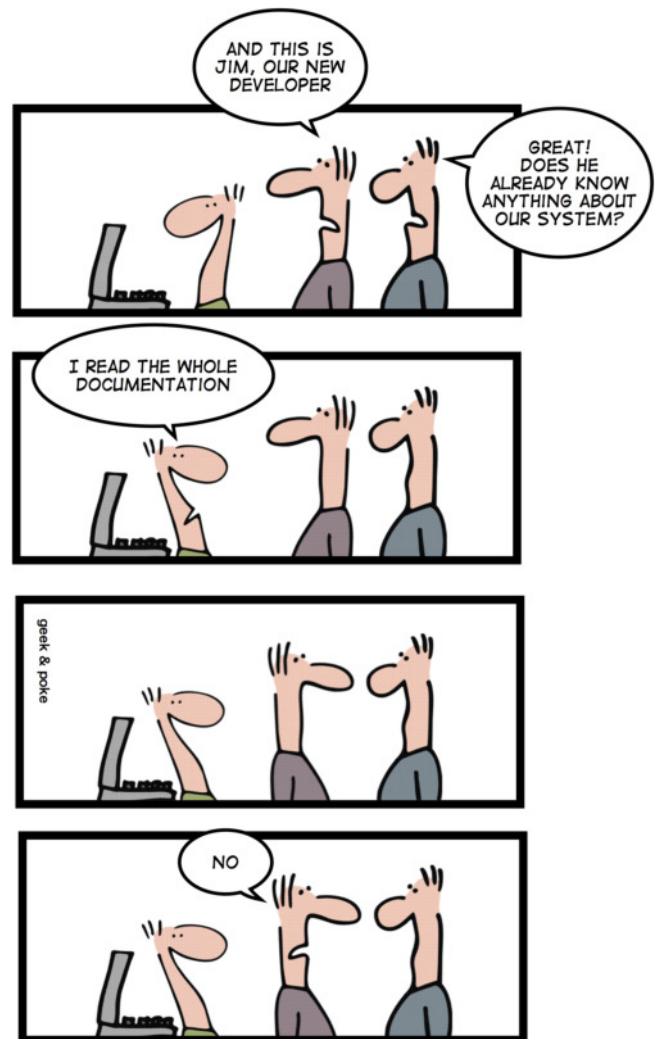
Ebenso wichtig ist es, sich zu überlegen, für wen der Fachtext eigentlich geschrieben werden soll. Wer braucht die so sorgsam erarbeitete Information? Wie sieht es mit der technischen Vorbildung der Leser aus? Das hat großen Einfluss auf das technische Niveau des Fachaufsatzes. Auch sollte sich der Autor Gedanken über den Zweck des Aufsatzes machen: Dient er nur dazu, technische Information zu vermitteln? Oder vergleicht er zwei Arten einer Problemlösung und sagt am Ende, welche für einen gegebenen Fall die bessere ist? Der Autor sollte also immer eine Antwort auf die Frage wissen: „Wer möchte zu welchen Zweck diesen Text lesen?“

Insbesondere das Schreiben von Fachartikeln etwa für Zeitschriften ist äußerst schwierig, weil das Alter, der Wissensstand und die Erfahrung der Leser viel breiter gestreut sind, als zum Beispiel beim Schreiben von Forschungsberichten für Wissenschaftspublikationen. Das einzige, was wir über den Leser von Fachartikeln in Zeitschriften oder auf Online-Plattformen wissen, ist, dass er sich zumindest grob für dasjenige Gebiet interessiert, dem sich das Medium zuordnen lässt. Da wir also nicht wissen, welcher Leser genau den Text lesen wird, ist es wichtig, ganz explizit am Beginn zu sagen, wovon der Text handelt, und wer ihn darum aus welchem Grund lesen sollte. Meistens haben Texte einen Vorspann, in dem sich diese Information wenigstens schon einmal zum Teil unterbringen lässt.

Strukturfragen

Wenn das Thema steht, kann der Autor beginnen, Material zusammenzutragen. Das kann unter Um-

ständen langwierig sein, etwa wenn Fachliteratur zu konsultieren ist oder Experten zu Wort kommen sollen, die zunächst akquiriert und dann interviewt werden müssen. Auch gehört manchmal zum Fachtext, Versuchsschaltungen aufzubauen und Messungen durchzuführen, oder die Evolution einer selbst geschriebenen Software nachzuvollziehen.



Im zweiten Schritt müssen die Fakten irgendwie geordnet werden: Alles, was zu einem bestimmten Teilaspekt gehört, gehört in eine Gruppe, und die einzelnen Gruppen sind so zu kombinieren, dass sie eine sinnvolle Reihenfolge ergeben. Es gibt viele Wege, das zu tun. Manche schreiben gern in einem Rutsch. Allerdings setzt das voraus, dass sie vorher alles Wichtige im Kopf zwischenspeichern, oder Notizen bereit liegen haben. Andere finden es naheliegender, die Inhalte einzeln aufzuschreiben, und dann die beste Reihenfolge durch Einfügen, Streichen und Verschieben von Textteilen zu erreichen. Allerdings wird der Text dadurch holprig – da sind wir wieder bei den Stilfragen. Doch um den Textstil, der bewirkt,

dass das Ganze flüssig wird, kann man sich später noch kümmern.

Zunächst gilt es, den Inhalt zu strukturieren. Grundsätzlich gibt es zwei Herangehensweisen: Bin ich Richter, oder bin ich Gutachter? Der so genannte Urteilsstil zum Strukturieren besteht darin, zuerst das Ergebnis darzustellen, und anschließend die Gründe, Sachverhalte und sonstigen Gedankengänge, die zu diesem Ergebnis geführt haben. Der Autor ordnet jede einzelne Erwägung ein, weshalb sie für das (Zwischen-)Ergebnis wichtig ist. Der Gutachterstil hingegen nennt zuerst Sachverhalte, Gründe, Fakten und andere relevante Tatbestände. Bei jeder einzelnen begründet er ihre Relevanz für das Untersuchungsziel. Am Ende kommt er zu einem aus allen Tatbeständen abgeleiteten Urteil. Gutachter- und Urteilsstil sind einander entgegengesetzt und damit inkompatibel.

Egal, welche Herangehensweise Sie wählen: Die zufrieden stellendste Struktur ist gewöhnlich die, die eine logische Folge der einzelnen Inhalte zeigt. Einer solche Ordnung kritisch im eigenen Text zu suchen, hat qualitätssichernde Konsequenzen:

- Lücken werden offensichtlich. Der Autor ist dann zwar gezwungen, nachzulegen, gar neue Recherchen anzustellen. Aber er stellt so sicher, dass auch der Leser später keine Lücken findet. (Abgesehen von Sorgfalt beweist der Autor so auch einfach Kompetenz.)
- Die Aufteilung des Themas in einzelne Abschnitte wird von selbst deutlich und hilft beim Erstellen geeigneter, den Leser führenden Zwischenüberschriften.
- Diese ganze Mühe führt zu einem angenehmen Nebeneffekt: Die Ausleuchtung dunkler Ecken unseres eigenen Wissens ist eine der Belohnungen für das Schreiben von Fachaufsätzen.

Fachautoren, die technisches Schreiben vor allem aus Büchern oder wissenschaftlichen Aufsätzen kennen, neigen dazu, ihr Thema in fünf oder sechs Unterebenen zu schachteln. Das ist zwar für den Autor nützlich, der damit seine Gedanken ordnet. In der endgültigen Ausarbeitung sind mehrere Überschriften-Ebenen aber ungünstig, weil sie den Leser verwirren. Insbesondere in Fachartikeln für Publikationen der öffentlichen

Presse ist mehr als eine Überschriftenebene meistens verpönt. Hintergrund ist, dass der streng serielle Lesefluss hier eine größere Rolle spielt, weil die Konsumption des Artikels als Ganzes die Regel ist. Anders ist das bei den oft viel längeren Buch- oder Wissenschaftstexten, in denen öfter nur einzelne Abschnitte interessieren.

Der Trick ist also, Kategorien und Subkategorien nicht in geschachtelten Überschriften abzubilden, sondern zu serialisieren. So kann man etwa die folgenden geschachtelten Überschriften und Unterüberschriften leicht durch zwei anders formulierte ersetzen:

- Rechnerarchitektur
 - RISC (Reduced Instruction Set Computer)
 - CISC (Complex Instruction Set Computer)

... wird zu ...

- RISC-Architektur (Reduced Instruction Set Computer)
- CISC-Architektur (Complex Instruction Set Computer)

Optikfragen

Ist alles endlich beisammen und geordnet, könnte der Autor denken, der größte Teil der Arbeit sei getan. Die strukturierten Informationen so in Worte umzusetzen (sprich: zu formulieren), dass man sie auch gut versteht, wird oft nicht als eigener Schritt betrachtet. Schließlich muss sich derjenige Leser, der den Text so nicht versteht, lediglich besser konzentrieren, denn alle notwendigen Informationen sind ja schwarz auf weiss vorhanden, nicht wahr? Aber: Es ist Sache des Autors, so zu schreiben, dass man das alles auch versteht. Nicht was man schreibt, sondern *wie* man schreibt, das nennt man den Stil.

Der Stil umfasst normalerweise solche Dinge wie Wortwahl und Satzlänge. Sicher sind das zwei für die Lesbarkeit eines Textes wichtige Dinge. Aber ein Fachartikel ist mehr als die Aneinanderreihung von Worten; auch visuelle Hilfsmittel wie Illustrationen oder Tabellen gehören dazu. Denn es ist auch eine Stilfrage, wie man Informationen präsentiert, nicht nur, wie man sie beschreibt.

Welche Darstellungsart am besten geeignet ist, muss der Autor von Fall zu Fall entscheiden. Manchmal stellt es sich zum Beispiel auch für den

Schreibenden als ausgesprochen nützlich heraus, Zusammenhänge einfach mal zu zeichnen. Diesen Erkenntnisgewinn braucht er dann nicht umständlich wieder in Worte zu verpacken – man kann dem Leser auch die entsprechende Grafik zeigen. Dasselbe gilt für Tabellen: Den Zusammenhang etwa von Messwerten zeigt eine Tabelle viel deutlicher, als wenn sie wortreich ein Blatt füllen.

Allerdings ist es dann auch die Pflicht des Autoren, darauf zu achten, die gewählte Abbildung – egal ob Bild, Tabelle oder Diagramm – in den Text einzubinden. Das bedeutet, die Besonderheit der gewählten Grafik ist im Text zu erwähnen. Auch ist es eine Sache guten Stils, jeder Abbildung eine Bildunterschrift beizugeben, die sagt, was man hier sieht. Es ist hingegen verschwendete Zeit von Leser und Autor, den Inhalt einer Zeichnung eins zu eins in Worten zu duplizieren. Stattdessen geht es darum, den Leser zu dem Punkt zu führen, an dem er die Grafik versteht.

Bei manchen Grafiken stellt man vielleicht fest, dass sie einfach zu komplex ist, als dass man sie elegant einbinden könnte. In so einem Fall ist es besser, die Zeichnung zu ändern und auf das Wesentliche zu reduzieren, oder zwei oder mehr thematisch geschlossene (Teil-)Zeichnungen daraus zu machen. Man sollte den Leser auf keinen Fall mit sehr komplexen Abbildungen allein lassen und darauf vertrauen, dass er das Ganze schon selbst versteht. Man läuft so nämlich auch Gefahr, das Abgebildete selbst nicht wirklich in das Thema einordnen zu können.

Und noch ein Wort zu manchmal seltsam erscheinenden redaktionellen Vorgaben. „Eine gute Illustration ersetzt 500 Worte“; „Wir möchten mindestens ein Bild pro Seite haben“: Solche Ratschläge scheinen wenig wert. In welchem Ausmaß und welche Abbildungen Sinn ergeben, hängt schließlich vom Thema und vom Leserkreis ab. Es gibt tatsächlich Themen, wo sich nicht allzu viel visualisieren lässt. Im anderen Extrem machen zwei Screenshots pro Seite mit fünf Zeilen Text dazu noch keinen guten Workshop-Artikel. Es gilt, die Balance zu finden. Und es gilt auch, neben der psychologischen auch die physiologische Lesesituation nicht aus dem Blick zu verlieren: Wer Text liest, und sei dieser noch so kurzweilig, dessen Auge ist dankbar für ein Wasserloch in der Buchstabenwüste, und dessen Gehirn reagiert positiv auf eine Abwechslung der Inanspruchnahme seiner Gehirnhälften.

Jetzt wird's ernst: Formulierungsfragen

Irrelevant in einem Text ist nicht nur das, was nicht zum Thema gehört. Sondern irrelevant ist auch das, was schlicht zu wenig aussagt – entweder, weil es zu verschwurbelt ist, um verständlich zu sein, oder weil es ein Allgemeinplatz ist („IT-Sicherheit wird immer wichtiger“). Es ist auch ein weitverbreiteter Irrtum zu glauben, dass Artikel mit fachlichem Anspruch pompöse, nach Gesetzestext klingende, möglichst aus dem Lateinischen stammende Wörter enthalten müssen. Oder ganz viele Substantive mit „-ierung“, „-ung“, „-heit“ oder „-tion“ am Ende, und natürlich – am allerwichtigsten – Passivkonstruktionen, weil die so schön absolut klingen.

Ganz im Gegenteil: Kein Mensch liest so etwas wirklich gern. Es kann sogar auch auf Seiten des Schreibenden, also nicht nur des Lesenden, das Verständnis dafür erschweren, was das Geschriebene eigentlich sagt. Und es kann schlechterdings ein Mitgrund sein, warum sich Ingenieure vor dem Schreiben scheuen: Sie sehen sich nicht in der Lage, ein solch gestelztes Deutsch zu verfassen. Solchen Technikern rufen wir zu: Juhu! Umso besser! Und keine Angst: Um glaubhaft zu sein, muss ein Text nicht kompliziert sein, sondern klar und direkt. Das hängt unmittelbar mit der Aufgabe von Fachaufsätzen zusammen – nämlich Information zu vermitteln.

Hier wird Passiv zur Sprache gebracht

Passivkonstruktionen sind ähnlich beliebt wie Substantivierungen. Oft schleppt das Passiv sogar die so genannte Verwaltungs- oder Behördensprache hinter sich her. Diese zeichnet sich dadurch aus, dass Substantive vor Verben den Vorzug erhalten. Man baut sogar Substantive ein, wo eigentlich nur ein Verb vorliegt („zur Sprache bringen“ statt „besprechen“). Das klingt nämlich hoheitsvoller und irgendwie amtlicher – als Autor spiegelt man sich so unbewusst gern selbst. Dafür gerät der Text aber anstrengender, und in der Folge schlechter verständlich. So etwas hat in einem Fachartikel nichts zu suchen.

Nun sind Passivkonstruktionen für die Techniksprache typisch: Da „wird“ angeschlossen, installiert, auf OK geklickt und so weiter. Wieso eigentlich?

Aktivkonstruktionen benennen die handelnde Person oder Gruppe. Passiv entpersonalisiert einen Vorgang. Das bedeutet, die handelnde Person kommt nicht zur Sprache. Passivaussagen wir-

ken dadurch objektiver. Und in der Tat: Manchmal sind die etwas bewirkenden Personen (oder andere Wirkungsursachen) nicht nennenswert. Wenn jemand vollgerechnet wird, dann ist klar, womit; wenn jemand nassgemacht wird, dann ist die Ursache (überwiegend) zweitrangig. Aber: Passivkonstruktionen bewirken Distanzierung, so als ob der Autor sich von dem geschriebenen fernhalten möchte. Und, noch schlimmer, Passivkonstruktionen lassen oft wichtige Informationen einfach aus. Es gibt Fälle, in denen es eine Wissenslücke ist, nicht zu benennen, wer oder was etwas tut. Das stellt man nur dann fest, wenn man die Passivkonstruktion hinterfragt.

Es empfiehlt sich nicht, Passivkonstruktionen als Tabu zu betrachten (es sei denn, Sie schreiben für eine Redaktion, die eine entsprechende Policy hat). Manchmal ist es sogar richtiggehend umständlich, einen Passivsatz aktivisch umzuformen. Geübte Schreiber setzen Passiva darum sinnvoll, nicht inflationär und vor allem nicht informationsverhüllend ein. Und geübte Schreiber achten sowieso darauf, verstärkt mit Verben zu arbeiten, auch wenn der im Kopf festsitzende Behördenstilvertreter empört mit den Fäusten wedelt.

Ich, Du, Er, Sie: Leseransprache

Wer versucht, Passivkonstruktionen auszumergen, steht automatisch vor der Herausforderung, den Handelnden nicht nur zu identifizieren, sondern auch zu benennen. Manchen Autoren widerstrebt es, „Ich“ zu sagen, auch wenn das Ich im Text – also der Autor selbst – wahrheitsgemäß der Handelnde ist („Ich füge diese Zeile Code ein, um das Skript anzupassen“). Andere Autoren bringen gern ihre eigene Person in den Text, zum Beispiel, um ihm eine persönliche Note zu geben. Wieder andere sagen lieber „Ich“, als von sich selbst in der dritten Person zu sprechen, weil sie das zu sperrig oder schlichtweg albern finden („Der Autor fügte diese Zeile Code ein, um das Skript anzupassen“).

Es gibt noch eine dritte Möglichkeit, die aber etwas Übung erfordert. Sie besteht darin, das Subjekt zu übertragen. Ziel der Übertragung kann entweder etwas sein, das im Satz bereits vorkommt und dem Handelnden als Helfer dient, etwas zu bewirken („Diese Zeile Code passt das Skript an“). Oder Ziel der Übertragung ist eine Bezeichnung, die von der individuellen Person des Handelnden absieht, aber seine Funktion beibehält („Der Admin/Programmierer/Bastler fügte diese Zeile Code ein, um das Skript anzupassen“).

Es ist ein gewisses Dilemma, das nicht nur beim Passivausmerzen aufkommt: Wer das Ich in einen

Text bringt, nimmt in Kauf, dass er entneutralisiert. Wer immer nur „der Autor“ sagt, nimmt in Kauf, dass das mitunter merkwürdig klingt. Und wer beides nicht möchte, leistet Mehrarbeit dafür, sich passende Subjekte zu überlegen.

Wenn keine Policies vorliegen, bleibt es dem Autoren überlassen, wie er sich entscheidet. Er kann dann sowohl das Ich-Tabu als auch die Verwendung der dritten Person über Bord werfen. Schreiben Sie also ruhig: „Stecken Sie danach den USB-Dongle an“, wenn Sie das Passiv vermeiden, sich nicht krummlegen und den Leser lieber einfach direkt ansprechen möchten. Wobei. Sollten Sie den Leser eigentlich siezen, oder duzen?

Ikea duzt notorisch und überall. Das ist Teil der Corporate Identity: Weil Ikea aus Schweden kommt und Schweden sich duzen, wird auch der deutsche Kunde geduzt. Man duzt sich zum Beispiel auch in manchen Teilen Süddeutschlands generell häufiger, auch unter Fremden. Wanderer duzen sich oberhalb von 1500 Höhenmetern generell, im Tal fällt dieses Wanderer-Du wieder weg. In den 1980er und 90er Jahren duzten sich wildfremde Menschen in Newsgruppen und E-Mails – weil man sozusagen eine eingeschworene Gemeinschaft war. Auch Blogger duzen ihre Leser gern, und inzwischen findet man das Du in etlichen Stellenanzeigen.

Sollte auch ein Fachautor seine also Leser also duzen? Der Autor Erik Bartmann duzt in seinem Buch *Die elektronische Welt mit Raspberry Pi entdecken* den Leser von Anfang an hemmungslos. Das bewirkt aber bei manchem unangenehme Assoziationen: Möchte sich der Autor anbiedern („Wir sind doch beide Hacker/Maker/Admins“)? Oder, was schlimmer ist, schaut der Autor auf den Leser herab („Ich weiß bescheid und erkläre Dir die Welt“)?

Man kann mit Fug und Recht der Ansicht sein, dass man als Autor das Duzen auf die Zielgruppe Kinder beschränken sollte. Denn Nähe herstellen, indem man den Leser duzt, ist nicht jedem Leser angenehm. Allerdings stellt auch das Ansprechen per Sie Nähe her (wenn auch eine respektvollere), weil es eine direkte Beziehung aufbaut. Fachtexte gelingen dadurch gelegentlich inhaltlich weniger überzeugend.

Hier wird sichtbar: Der Stil wirkt sich auf den Inhalt aus, auch wenn der Inhalt an sich vollkommen in Ordnung ist. Aber wenn Sie sich an eine Handvoll Grundsätze halten, wie sie dieser Beitrag erläutert hat, brauchen Sie sich davor nicht fürchten.



Aushang: Typische Fallen und ihre Alternativen

- **Lange Sätze** gehören aufgeteilt in mehrere kurze. Ab und zu ein langer Satz ist in Ordnung, wenn er nicht zu verschachtelt ist. Abwechslung belebt sogar den Lesevorgang – mit etwas Übung stellen Sie fest, dass auch das Schreiben so mehr Spaß macht, wenn Sie mal mehr und mal weniger auf die Tube drücken.
- **Komplizierte Worte**, wie man sie von Politikern und Richtern erwartet, passen nicht in einen Fachaufsatz, der etwas erklären möchte. Suchen Sie das einfachste funktionale Wort, und probieren Sie auch, ob sich etwas vielleicht besser als Handlung beschreiben lässt.
- **Hilfs- und Modalverben** wie *können*, *werden*, *sollen* sind so weit wie möglich zu vermeiden. Diese Wörter relativieren Sachverhalte, und zwar mit Absicht, sei den Autoren mal unterstellt. Aber am Ende wirkt es schlicht so, als sei der Autor nicht davon überzeugt, was er schreibt, und das verunsichert den Leser. Dinge, die man tun „kann“, „wird“ oder „soll“, **tut** man also stattdessen ganz beherzt in der kleinen heilen Welt dieses Artikels. Wenn Sie sich absichern möchten, machen Sie lieber eine Fallunterscheidung, als auszuweichen.
- Einen flüssigen Text zeichnen **Zusammenhänge** aus. Absatz- und Satzanfänge bewirken Zusammenhang, wenn sie mit „aber“, „trotzdem“, „ebenso“ oder „daher“ beginnen.
- Idealerweise trägt jedes Wort zur Information des Lesers bei. Alles, was keine Information trägt, ist **überflüssig** und fliegt raus.
- Hüten Sie sich vor **allgemeinen Aussagen**. Dem Leser nützen sie nichts. Sie sagen ihm lediglich, dass Sie selbst nicht so ganz genau wissen, wovon Sie reden. Auch wenn Sie das in Wahrheit tun: Für den Leser zählt nur, was im Text steht.
- Der **Einstieg** birgt das größte Risiko für Allgemeinaussagen. Auch beliebt: Viele Aufsätze fangen bei Adam und Eva an. Wenige versuchen zum Beispiel, direkt in eine Situation einzusteigen, oder mit einem konkreten Fakt anzufangen und daraus eine Frage abzuleiten. Dabei beschert das sehr willkommene Dynamik, nicht nur dem Schreibenden bei der Arbeit, sondern auch hinterher dem Lesenden.
- Wer laut ist, hat noch lange nicht recht: **Ausrufezeichen** sind sehr, sehr, sehr spärlich zu verwenden, erst recht in Titeln (fast nie passend) oder Zwischenüberschriften: „Zurückhaltung üben!“ sagt nicht mehr als „Zurückhaltung üben“, sondern wirkt nur quengeliger.
- So manche **Zusammenfassung** wäre besser weggeblieben. Haben Sie sich verzettelt, oder warum brauchen Sie eine Zusammenfassung? Zumal eine, die bloß kürzer nochmal hinschreibt, was sowieso schon beschrieben war? Wenn wesentliche Zwischenergebnisse noch einmal zusammenzutragen sind, dann sollte das gleich zur nächsten Aussage führen. Abschnitte, die „Zusammenfassung“ heißen, sind in der Regel überflüssig.
- Ähnliches gilt für das **Fazit**. Wenn Sie zu einem Ergebnis kommen, dann ist das genau der Job, den Sie machen sollen, wenn Sie einen Fachartikel schreiben. Sie zeigen nur Unbeholfenheit, wenn Sie Ihr Ergebnis mit „Fazit“ betiteln, anstatt zum Beispiel eine abschließende Zwischenüberschrift zu wählen, die das Fazit auch benennt.
- Ab welchem Punkt lohnt sich **Polieren** nicht mehr? Wenn Sie unsicher sind, lassen Sie einen Kollegen oder die Redaktion entscheiden. Grundsätzlich gilt – und das ist sehr wichtig, weil die Welt sonst womöglich ohne Ihre Artikel auskommen muss, mit denen sie aber besser dran wäre: Zwei Artikel mit 70 Prozent Perfektion sind besser als einer mit 90 Prozent Perfektion. Am schlimmsten ist gar kein Artikel mit 100 Prozent Perfektion.

Shellskripte mit Aha-Effekt VI

Named Pipe – das unbekannte Wesen

Ein selten verwendetes Konzept steht mit der *Named Pipe* zur Verfügung. Sie ähnelt der normalen Pipe, nur lebt sie dauerhaft und kann von beliebigen Prozessen genutzt werden. So bietet sie eine weitere Möglichkeit zum Datentransfer zwischen Skripten, sogar zwischen unterschiedlichen Usern.

von Jürgen Plate

„Ceci n'est pas une pipe.“

René Magritte in *La trahison des images*

Eine Eigenheit der *Named Pipe* (auch *FIFO* genannt – First In First Out) ist, dass der Kernel einen Prozess, der etwas in eine Pipe schreibt, so lange blockiert, bis auf der anderen Seite ein Prozess etwas daraus liest. Für beide Prozesse erscheint die Named Pipe wie eine Datei. Der Sendeprozess leitet seine Ausgabe in die Pipe, der Empfängerprozess erhält seine Eingabe aus der Pipe. Für eine bidirektionale Kommunikation werden zwei Named Pipes benötigt.

Zum Anlegen einer Named Pipe dient das Kommando `mknod`. Da es verschiedene Arten von Dateien anlegen kann, muss hier der Buchstabe `p` die Named Pipe kennzeichnen:

```
mknod name p
```

Alternativ erfüllt das Kommando `mkfifo` den gleichen Zweck:

```
mkfifo name
```

Existiert eine Named Pipe namens `roehre`, kann ein Prozess in diese Pipe schreiben, zum Beispiel:

```
cat > roehre
```

Ein zweiter Prozess liest aus der Pipe:

```
cat < roehre
```

Oder auch:

```
tail -f roehre
```

Statt einer Ausgabe auf dem Bildschirm können die Daten auch anderweitig verarbeitet werden. Ein Nachteil ist, dass beide Prozesse blockieren, bis die Daten übertragen wurden: der Sendeprozess hängt, bis der Empfängerprozess die Pipe zum Lesen öffnet. Will man gleich weiter arbeiten,

schiebt man den Sendeprozess mittels Ampersand also in den Hintergrund.

Endet einer der beiden Prozesse, endet im Regelfall auch der andere. Ein Gegenbeispiel dafür ist `tail -f`. Das Programm läuft so lange, bis es explizit beendet wird. Hier könnten sogar etliche Prozesse Meldungen loswerden, etwa mittels `echo`.

Die Named Pipe existiert natürlich weiter, bis sie gelöscht wird – wie eine gewöhnliche Datei. Damit auch Prozesse anderer User auf eine Named Pipe lesend und schreibend zugreifen können, müssen die Zugriffsrechte angepasst werden, denn in der Regel verfügt nur der Eigentümer über das Schreibrecht.

Hin und Her

Eine Named Pipe funktioniert immer nur in einer Richtung. Daten, die ein Prozess in die Pipe schreibt, können nur von einem einzigen Prozess gelesen werden (Abbildung 1).

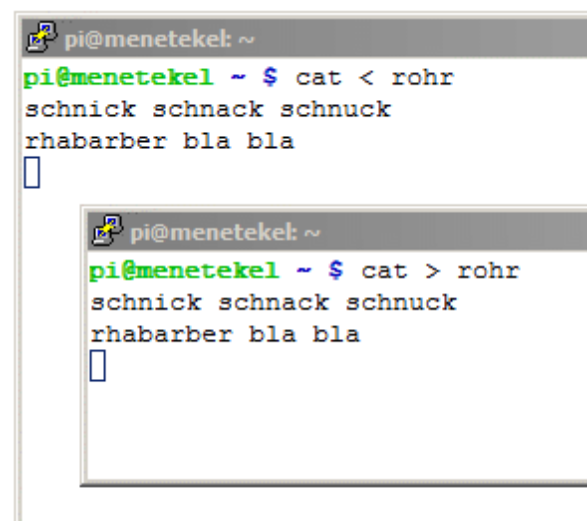


Abbildung 1: Zwei Shells kommunizieren über eine Named Pipe.

Will man Daten in beiden Richtungen austauschen, benötigt man zwei Pipes. Im folgenden Beispiel interagieren die zwei Pipes *hin* und *her*. Damit kann man auf Shell-Ebene chatten, ganz ohne Schlautelefon und *WhatsApp*. Abbildung 2 zeigt, wie es geht.

```

pi@menetekel: ~
pi@menetekel ~ $ cat < her &
[1] 2774
pi@menetekel ~ $ cat > hin
hallo
was gibt's?
oben: laber laber
unten: bla bla bla

```

```

pi@menetekel: ~
pi@menetekel ~ $ cat < hin &
[1] 2776
pi@menetekel ~ $ cat > her
hallo
was gibt's?
oben: laber laber
unten: bla bla bla

```

Abbildung 2: Zwei Named Pipes ermöglichen chatten auf Shell-Ebene.

In beiden Shells schiebt ein Ampersand den empfangenden Prozess in den Hintergrund – was den Prozess nicht daran hindert, seine Ausgaben auf den Bildschirm zu schreiben. Dann startet der jeweilige Sendeprozess im Vordergrund, und schon steht dem Text-Zwiesgespräch nichts mehr im Wege.

Daten verarbeiten

Will man die – gegebenenfalls von verschiedenen Prozessen stammenden Meldungen – weiterverarbeiten, ist *cat* ungünstig. Es terminiert, wenn der schreibende Prozess die Pipe schließt. *tail -f* macht aber auch nicht ganz glücklich. Also verlagern wir die Weiterverarbeitung auf die Shell.

Die folgende Programmsequenz öffnet in Zeile 03 immer wieder unsere Pipe *roehre* zum Lesen, bis Zeile 06 die Schleife und damit das Programm beendet. Das erlaubt die kontinuierliche Verarbeitung der ankommenden Meldungen in Zeile 04,

denn das *read*-Kommando blockiert den Ablauf, bis Daten eintreffen. Steht in einer Zeile nur das Wort *ende*, verzweigt das Skript in Zeile 05 zum Abbruch der Schleife, was die darauf folgende Zeile bewerkstelligt:

```

01 while true
02 do
03   read LINE < roehre
04   echo $LINE
05   if [[ "$LINE" == 'ende' ]]; then
06     break
07   fi
08 done

```

Auf diese Weise kann man eine Named Pipe nutzen, um mehrere Prozesse Log-Messages zu einem gemeinsamen Ziel senden zu lassen.

Will man dagegen mehrere Zeilen von einem einzigen Sender verarbeiten, muss man die Pipe nicht in das *read*-Kommando leiten, sondern in die *while*-Schleife. Dann öffnet nicht jeder Lesevorgang die Pipe und schließt sie wieder. Vielmehr öffnet der Schleifenstart die Pipe (Zeile 04), und sie schließt erst dann, wenn sie leer ist. Hier ist der Rückgabewert des Lesebefehls in Zeile 01 die Abbruchbedingung für die Schleife. Diese endet, wenn *read* Dateiende oder einen Timeout signalisiert:

```

01 while read LINE
02 do
03   echo $LINE
04 done < roehre

```

Zum Schluss noch ein etwas verblüffendes Beispiel mit folgender Kommandosequenz:

```

pi@menetekel: ~
pi@menetekel ~ $ cal > hin &
[1] 2431
pi@menetekel ~ $ date > hin &
[2] 2432
pi@menetekel ~ $ cat < hin
Mo 9. Mai 17:50:18 CEST 2016
      Mai 2016
So Mo Di Mi Do Fr Sa
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

[1]-  Fertig                                cal > hin
[2]+  Fertig                                date > hin
pi@menetekel ~ $

```

Abbildung 3: Überraschung.

Überraschung! Das Beispiel zeigt ein unerwartetes Verhalten: Wir starten zunächst `cat` und danach `date`. Lassen wir dann `cat` den Text aus der Pipe lesen, erscheint die Ausgabe nicht in der Reihenfolge, in der sie versendet wurde.

Das Problem liegt darin, dass die beiden Sendeprozesse anfangs blockiert sind. Sobald der Le-

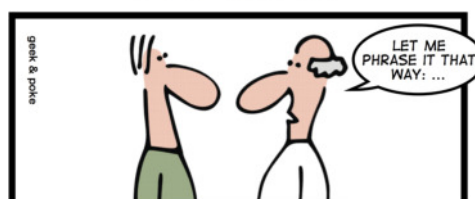
seprozess mit der Named Pipe verbunden ist, wachen alle Sendeprozesse auf. Doch der Scheduler im Kernel aktiviert die Prozesse nicht in einer bestimmten Reihenfolge. Daher gibt es keine Garantie für das Verhalten der Pipe bei mehreren Sendern. Es kann sogar sein, dass längere Ausgaben von mehreren Sendeprozessen gemixt werden.

Über Jürgen

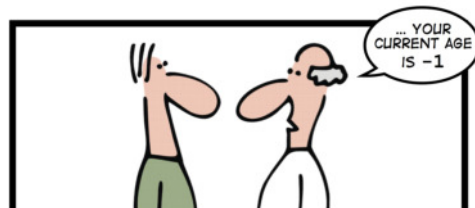
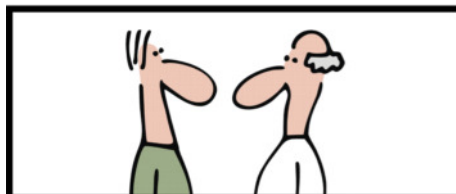


Jürgen Plate ist Professor für Elektro- und Informationstechnik an der *Hochschule für angewandte Wissenschaften München*. Er beschäftigt sich seit 1980 mit Datenfernübertragung und war, bevor der Internetanschluss für Privatpersonen möglich wurde, in der Mailboxszene aktiv. Unter anderem hat er eine der ersten öffentlichen Mailboxen – TEDAS der *mc*-Redaktion – programmiert und 1984 in Betrieb genommen.

DR. GEEK



IMAGINE YOU'RE A PHYSICIAN AND YOU HAVE TO REVEAL YOUR PATIENT THAT HE HAS REACHED THE LAST YEAR OF HIS LIFE



WHY DON'T DO IT THE GROOVY WAY?

Das hat doch alles einen Sinn Musings on IT-Management

Glücklich kann sich preisen, wer in der eigenen Berufspraxis Managementmethoden mit guten Erfahrungen verbindet. Dem Autor ist genau dies gelungen. Den Schatz seiner Einsichten hat die Redaktion aus seinem Blog geborgen.

von **Frank Benke**

Win-Win Situation:

Die einen super Geld verdient,
die anderen tolles Projekt gemacht.
Nur: Mangels Transparenz weiß niemand,
dass es netto nichts gebracht hat.

Frank Benke

Seit etwas mehr als zwei Jahren darf ich als IT-Leiter bei einem gesunden Maschinenbauer respektabler Größe wirken. Die Stelle wurde mir angeboten, um tatsächlich IT-Management einzuführen und das Unternehmen IT-seitig auf kontinuierliches Wachstum und robusten Betrieb zu trimmen. Die Gelegenheit war reizvoll, moderne Verfahren zum IT-Management anzuwenden, und tatsächlich all die Dinge auch in vollem Umfang selbst umzusetzen, die mir als Consultant im täglichen Leben begegnet waren – und die vielerorts verpuffen oder nicht gehört werden wollen. Sollte ich eine Schlüsselaussage treffen, wäre sie:

Der wesentliche Erfolgsfaktor für technische Erneuerung und Innovation in einer operativen IT-Umgebung ist die Organisation, keinesfalls die Technik.

Oder mit anderen Worten: Wenn Innovationsprojekte scheitern, nicht fristgerecht enden, zu teuer werden oder keinen Effekt bringen, hat das in der Regel organisatorische Gründe.

Hierfür finden sich nach eigener Beobachtung in meinem aktuellen und ehemaligen Arbeitsumfeld, im Bekanntenkreis und nicht zuletzt in der Fachpresse unzählige Beispiele. Die meisten Fälle zeigen auf, wie es nicht geht. Nur sehr wenige zeigen, wie es tatsächlich funktioniert. Warum ist das so?

Denke ich eine Weile darüber nach, kommen mir tatsächlich und ausschließlich die folgenden sieben Aspekte einer modernen IT-Organisation in den Sinn.

1. Prozessorientierung

Auch wenn die einen es belächeln und die anderen übertreiben: Es geht nicht ohne. *ITIL*, die *IT*

Infrastructure Library, ist ein mächtiges Werkzeug. Dabei ist es egal, wie groß oder klein man daran geht – wichtig sind die hier definierten Kategorien. Die Kunst ist es, diese Kategorien mit Augenmaß auf die jeweilige Organisationsgröße anzupassen und eine Auslegung zu finden, die sich von den Mitarbeitern mit Leben füllen lässt. Hat man erst einmal ein solches Arbeitsmodell optimiert, ergibt sich daraus zwangsweise eine lösungsorientierte Organisationsstruktur.

Meine Empfehlung ist, dieses Thema als erstes anzugehen. Die wichtigsten Fragen sind:

- Welche Standardprozesse finden statt?
- Welche Relevanz haben sie?
- Welche Informationen sind notwendig und liefern Mehrwert?
- Wie groß ist der Aufwand, diese Informationen auch zu erzeugen?

Das Werkzeug für einen IT-Prozess ist immer nur so gut, wie die Informationen, die in ihm gepflegt werden. Dieser Pflegeaufwand muss in einem allgemein akzeptierten Verhältnis zu den operativen Tätigkeiten stehen. Ausnahmslos alle Mitarbeiter müssen dieses Vorgehen mittragen, sonst ist absehbar, dass Datenmüllhalden geboren werden, die alle positiven Effekte solcher Prozessarbeit eliminieren.

Sollten Einzelne hier nicht zu überzeugen sein, hat das oft mit dem nächsten Punkt zu tun.

2. Transparenz

Sie hat im Wesentlichen zwei Folgen. Erstens ermöglicht erst Transparenz eine vernünftige Planung, und das in jeder Hinsicht. Nur dann

kann ich vernünftig planen, wenn nicht andauernd durch Überraschungen Sachzwänge entstehen, die die Mannschaft dann damit beschäftigen, das Schlimmste zu verhindern. Beispiele sind der Einsatz von Mitarbeitern, der Ausbau von Infrastruktur, Evolution oder Migration von Anwendungen – alles, was zu einem rund funktionierenden IT-Betrieb beiträgt, sollte man als Planungshorizont unbedingt auf dem Schirm behalten. Last but not least kann ich auch erst in einem derart vorhersagbaren Betrieb neue Projekte sinnvoll planen.

Zweitens ermöglicht erst Transparenz zu beurteilen, welche Aufgaben mit welchen Mitteln lösbar sind. Budget und Budget-Treue, tatsächlicher Bedarf, tatsächliche Kosten sind alles verschiedene Aspekte von Transparenz. Transparenz ist die Voraussetzung für eine bedarfsgerechte Planung. Nur wenn der tatsächliche Bedarf mit realistischen Budgets abgeglichen wird – vielleicht auch noch angereichert um strategische Wachstumsoptionen – kann ich in einer internen Betrachtung seine Relevanz an die Unternehmensführung vermitteln. Neudeutsch spricht man auch gern vom *Business Case*.

Zu entwickeln sind hier Systemanalyse, Verbrauchsanalyse, Bedarfsanalyse, Trends aus Kapazitätsmonitoring, Kostenmonitoring und deren Auswertung sowie ein eigener, realistischer Budgetplan. Wenn solche Budgets nicht genehmigt werden, war es schon immer eine gute Idee, die Pläne und Kalkulationen gut aufzubewahren.

3. Projektfähigkeit

Die Projektfähigkeit einer Organisation – sprich, wie viele Projekttag kann das Team selbst leisten, wie viele interne Zuarbeit braucht ein Projekt, wie viel kann man extern vergeben, welchen Zeitraum kann das Team zur Steuerung externer Projektressourcen verwenden – diese Projektfähigkeit einer Organisation ist natürlich immer nach oben begrenzt. Die Grenze mag je nach Organisation unterschiedlich hoch sein, aber sie existiert, und man ist gut beraten auszuloten, wo in etwa sie liegt.

Typischerweise lernt man recht schnell, wann man darüber liegt. Meiner Einschätzung nach ist sie normalerweise niedriger als vermutet. Der Einsatz externer Ressourcen führt dann nur zu mehr Komplexität und Kosten, mitnichten aber zu größerem Projektfortschritt, wenn man erst einmal jenseits der Tragfähigkeit dieser organisatorischen Grenzen angelangt ist.

Nicht, dass sich danach nicht doch normalerweise alle im Kreis aufstellen und gegenseitig auf die Schulter klopfen. Es ist ja auch eine Win-Win-Situation: Die einen haben super Geld verdient, und die anderen ein tolles Projekt gemacht. Nur: Mangels Transparenz weiß halt niemand, dass es netto nichts gebracht hat.

Ganz dringend sind also Projekt-Reviews immer und konsequent durchzuführen. Ich persönlich tendiere auch schon vorab bei der Projektvergabe zu einer Planung der Arbeitspakete, die zwischen internen und externen Ressourcen unterscheidet und einen realistischen Korridor zur Bewertung eingehender Angebote aufzeigt. Man kann sich sicher Abweichungen erklären lassen, vielleicht auch als Projekt-Vergebender hinzu lernen. Auf jeden Fall hat man aber eine Gesprächsgrundlage. Anbieter können einem so nicht schon vorab ein X für ein U vormachen.

Aktuell kann ich nach zwei Jahren mit zum Teil intensiven Projektphasen belastbar sagen, wie viele externe Consulting-Tage mein Team pro Jahr betreuen kann, ohne die täglichen Aufgaben eklatant zu vernachlässigen. Alle Projekte darüber hinaus sind verschwendetes Geld und können eigentlich gar nicht zu einem erfolgreichen Abschluss führen. Will ich mehr Projekte realisieren, muss ich diese mit eigenem Personal aufstocken. Mein Erfahrungswert ist, dass nicht mehr als 25 Prozent der intern geleisteten Mannstunden extern vergeben werden sollten.

4. Bereinigung von Altlasten

Im überwiegend konservativen deutschen IT-Establishment wird das Bereinigen von Altlasten sehr kontrovers diskutiert, sobald man als Berater nur langsam in diese Richtung steuert. Es war in jedem Umfeld, in dem ich bisher tätig war, der Überaufreger. Es findet sich immer jemand, der einem erklärt, was alles auf gar keinen Fall geht.

Das beruht meiner Meinung nach auf zwei Gründen. Der eine ist Angst vor Veränderung, der andere ist Angst vor Neuem, was nicht das gleiche ist. Tatsächlich wird in den seltensten Fällen der Aufwand wirklich betrachtet, eine Landschaft komplett rundzuerneuern. Wie auch? Ohne Transparenz scheint der Betrieb der Altlasten ja vermeintlich günstiger zu sein.

Ohne geeignete Organisationsstrukturen und Prozesse gibt es normalerweise keine realistische Chance, solche technischen Veränderungen zu realisieren, ohne den Betrieb zu gefährden. Betriebsgefährdung – oder auch nur die Störung des

Betriebs – ist wiederum die allergrößte Angst einer IT-Organisation überhaupt. Schließlich tritt man an, um einen möglichst störungsfreien Betrieb zu gewährleisten. Das Feedback aus der Geschäftsführungsetage ist auch meistens ziemlich drastisch, wenn es zu nennenswerten operativen Einschränkungen kommt.

Fakt ist aber auch, dass dies der größte Hemmschuh für die Weiterentwicklung einer jeden IT-Landschaft ist. Gründe gibt es viele: Althergebrachte Konzepte, alte Software-Architekturen, nicht mehr wartbarer Code oder nicht mehr wartbare Hardware, nicht mehr existente Lieferanten, verkorkte Systemstrukturen, Software-Alterung (ja, die gibt es!) oder fehlende Systemisolation sind Themen, die sich im Laufe der Jahre anhäufen. Doch alle diese Themen binden administrative Zeit und Geld – beides Dinge, die bei Aufbau und Betrieb moderner, teils erheblich leistungsfähiger Infrastruktur dann fehlen. Diese Gründe geben als inhärente Betriebsrisiken schon im Normalbetrieb Anlass genug, regelmäßig aufzuräumen.

Grundsätzlich ist neben der reinen Bereinigung von Altlasten eine *Business Impact Analyse* sehr hilfreich. Hierbei löst man sich etwas von der reinen IT-Sicht und prüft, wo genau die einzelnen Anwendungen im Geschäftsablauf ankern. Danach klärt sich, welchen operativen Schaden das Unternehmen bei Ausfall dieser Abläufe nimmt. Und dann hat man ein Szenario der unternehmerischen Risiken. Ergänzt man dieses Szenario um die zu Grunde liegenden Systeme und deren Abhängigkeiten, erhält man eine Landkarte der größten Baustellen, und im besten Fall sogar eine finanzielle Argumentationshilfe.

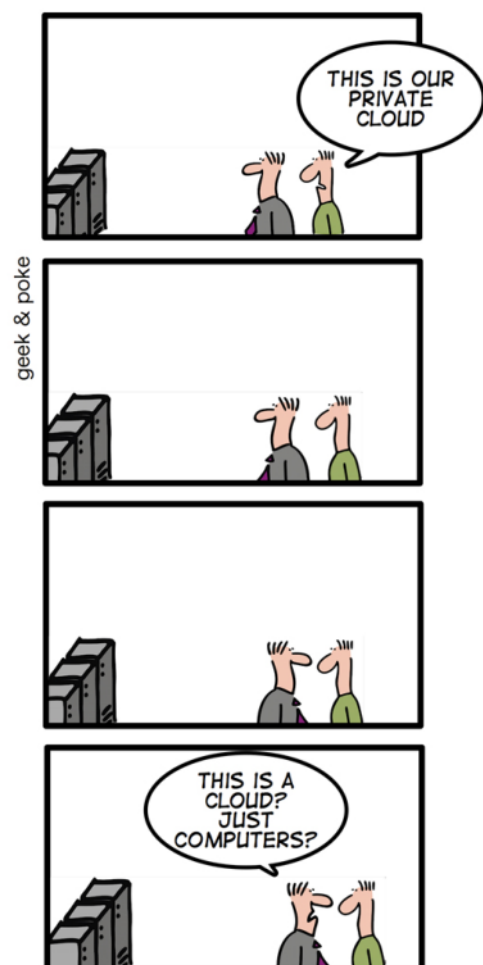
Sicher gibt es tatsächlich hier und da die eine Anwendung, ohne die tatsächlich nichts geht, die völlig resistent gegenüber Modernisierungsversuchen ist, oder bei der das Ende des Lebenszyklus wegen auslaufender Verpflichtungen winkt. Das ist dann die Ausnahme im Unternehmen! Aber genau das muss es auch bleiben: eine Ausnahme. Alles, was nicht niet- und nagelfest ist, gehört bereinigt.

5. Standardisierung

Sie gilt als der Heilige Gral der IT-Reorganisation. Wir standardisieren alles. Clients, Software-Ausstattung, System-Layouts, Standortkonzepte, Hard- und Software-Konfigurationen – was auch immer möglich ist, wird standardisiert. Dafür gibt es tatsächlich zwei sehr wesentliche Gründe.

Erstens bedeutet Standardisierung bei Einsatz von irgendetwas größere Stückzahlen desselben. Das reduziert den administrativen Aufwand, da nicht bei jedem Laptop oder bei jeder Serverinstallation das Rad neu zu erfinden ist. Einheitlich organisierte Systeme oder Standorte erlauben den Administratoren, sich in völlig anderen, vielleicht nicht täglich administrierten Umgebungen in kürzester Zeit zurecht zu finden. Die Betriebssicherheit steigt, da die Anzahl möglicher Fehlerquellen – meistens – sinkt. Ich sage "meistens", weil dann auch die Komplexität der jeweiligen Lösungen und der Support-Infrastruktur sowie deren Managementaufwand oft etwas zunehmen. Doch allein die Skalen-Effekte sprechen normalerweise für sich. Typischerweise schlägt sich diese Effizienzsteigerung auch im Beschaffungsprozess nieder, da mit Hard- und Software-Herstellern Rahmenverträge über größere Stückzahlen möglich werden, was in der Regel Rabatte zur Folge hat.

HOW TO DISILLUSION YOUR BOSS



Zweitens, und das ist im Zeitalter von Cloud, *Virtual Desktop Infrastructure* und Consorten vielleicht der weit wichtigere Punkt, ist Standardisierung die Grundvoraussetzung für Automatisierung. Diese wiederum ist keine Science Fiction,

sondern das Fundament all dessen, was heute unter Cloud verstanden wird: Ohne automatisierte Installations- oder Konfigurationsprozesse keine Cloud. Ohne standardisierte Komponenten, deren Monitoring und Reporting keine Service-Preise im Rechenzentrum und keine Hosting-Provider.

Dabei sind Standardisierungseffekte schon in relativ kleinen Umgebungen mit zwei- bis dreihundert Clients deutlich spürbar. Bei stagnierendem Personalstamm ist die Entlastung der Administratoren unabdingbare Voraussetzung für Wachstum und für die Einführung neuer Anwendungen. Teilweise ist Standardisierung die Voraussetzung, um überhaupt vernünftige, zeitgemäße und moderne Anwendungen und Infrastrukturen zu realisieren.

6. Innovation

Innovation gilt als die vorwärts treibende Komponente der IT, schon allein deshalb, weil Innovation permanent Einzug in unser tägliches Leben hält. Selbst randständige oder wirtschaftsschwache Gesellschaftsschichten stehen unter dauerhaftem Anpassungsdruck. Das Anwendererlebnis eines Arbeitnehmers sollte nicht substantiell unter dem seiner privaten Erfahrungen liegen – zumindest, solange man keine Übergeeks als Maßstab nimmt. Heute geben Familien mit *WhatsApp*, *Wunderlist* und *Netflix* den Ton an.

Innovation versetzt aber auch in die Lage, neue und moderne Anwendungen zu realisieren, Skalen- und Synergieeffekte zu erzielen, Sicherheit und Zuverlässigkeit zu steigern. Vielleicht der wesentliche, zumindest der kontinuierlichste Treiber sind wachsende Systemkapazitäten, da zunehmend Daten anfallen, diese in ihrer Struktur größer werden, der Wunsch nach längerer Verfügbarkeit im Raum steht, die Zugriffszeiten aus dem Ruder laufen und vieles mehr. Egal in welchem Datenbestand man eine Bestandsaufnahme macht, wie und in welcher Qualität diese Daten vorgehalten werden müssen, ob es Ablage- oder Transaktionsgeschäft ist, oder wie lange sie per Gesetz vorgehalten werden müssen: Man steht immer vor exponentiell wachsenden Szenarien. Einschlägige Marktforscher reden von Explosion: Gartner und Intel prognostizieren zwischen 40 und 63 Prozent Datenwachstum pro Jahr bei gleichbleibenden Aufgaben.

Hält man dem die typischen Anforderungen von Zuverlässigkeit und Verfügbarkeit entgegen, so sind die Datenmengen immer schlechter mit

konventionellen Mitteln zu bewältigen. Innovation ist daher alleine schon im Plattform- und Infrastrukturgeschäft eine Notwendigkeit. Um ein Beispiel zu bemühen: Die Wiederherstellungsszenarien von mehreren Terabyte großen Datenbanken zusammen mit der Zeitvorgabe, nach einem Totalausfall innerhalb eines Arbeitstages wieder funktional zu sein, bedeuten für viele klassische Systemlandschaften das Ende. Andere Bereichen wie Visualisierung, Änderung im Anwenderverhalten, Modernisierung der Arbeitsweisen, Verschiebungen in der Endgerätewelt hin zu mehr Mobilität und mehr intuitiven Arbeitsweisen erzeugen ähnliche Herausforderungen an die Innovationsfähigkeit einer IT-Organisation.

Grundsätzlich treffen Innovationsprojekte auf scheinbar unüberwindliche Hürden in der Unternehmensorganisation. Wenn diese Hürden nicht aus der IT selbst kommen, finden sich stets Fachabteilungen, die Veränderungen in Architektur und Systemlandschaft skeptisch bis feindselig gegenüber stehen. Diese Hürden kann man erfahrungsgemäß nur mit einer ausgesprochen strukturierten Planung und Vorgehensweise überwinden.

Aus meiner Sicht sind Innovationsprojekte darauf angewiesen, dass die ersten fünf genannten Punkte bereits geklärt sind. Man muss wissen, in welchem Mikrokosmos genau ein Innovationsprojekt landet, damit es andocken kann. Man muss die Anforderungen bedarfsgerecht und strategisch definieren, um die richtige und passende Lösung auszuarbeiten. Man muss eine Organisation haben, die solch neue Technologien auch aufnehmen und betreiben kann. Während der Einführung müssen Mitarbeiter die notwendigen Freiräume haben, um sich mit den neuen Themen auseinander- und die Innovation ordentlich umzusetzen. Nur so gelingt die Integration ins Unternehmen qualitativ hochwertig und innerhalb des vorgegebenen Zeitrahmens. Last but not least muss auch das Budget passen.

Dabei muss Innovation in der IT nicht teuer sein. Jedenfalls nicht teurer als das, was ohnehin an Budget nötig ist. Jeder IT-Regelbetrieb investiert, um Support und Betrieb alternder Systeme aufrecht zu erhalten. Nach dem *End of Sale* gehen die Wartungskosten für diese IT-Systeme typischerweise zügig nach oben. Den Status Quo zu erhalten, kostet Geld. Ich wage zu behaupten, dass moderne Technologien gerade vor dem Hintergrund aktueller Trends wie Virtualisierung und *Software Defined Infrastructure* bei gleichem Finanzaufwand immer die Leistungsfähigkeit und Kapazität um mindestens den Faktor Vier verbessern,

betrachtet über einen Zeitraum von drei Jahren.

Die zuvor geforderte Transparenz schafft Klarheit über Bedarf, Budget und in letzter Konsequenz auch über den Rahmen der Möglichkeiten, die für ein Innovationsprojekt zur Verfügung stehen. All das vorausgesetzt: Wie geht man an die Implementierung heran?

Der effektivste Weg zu neuen, komplexen Systemen folgt nach wie vor dem so genannten *Greenfield-Ansatz*: Der Bedarf ist geklärt; die Ziele sind eine möglichst elegante, kosteneffiziente und kosteneffektive Lösung; und Altlasten sind reduziert. Eine bessere Gelegenheit dafür gibt es kaum. Umgekehrt ruiniert Rücksicht auf Systembefindlichkeiten und Einschränkungen, die sich nicht ins aktuelle Jahrtausend transportieren lassen, die modernste Lösung. Nach Greenfield baut man auf einer optimierten Basis vom Reißbrett weg, was nur geht. So birgt man innerhalb des gesteckten Rahmens das größtmögliche Innovationspotential. Wenn die neue Umgebung funktioniert, migrieren Daten und Anwender aus der alten Welt sachte herüber, und die alten Systeme werden nach einer angemessenen Wartezeit einfach abgeschaltet und entsorgt.

Dieser Ansatz hat immer und grundsätzlich den Vorteil, dass zumindest zum aktuellen Zeitpunkt die neuen Systeme nach bestehenden Erfolgsrezepten entstehen. Keine Installationsreste aus jahrelangem Systembetrieb verbleiben, und die Effekte von Software-Alterung werden zumindest zum aktuellen Zeitpunkt auf Null gesetzt. Das Motto ist hier für mich immer: Abreißen, neu bauen und Daten migrieren, statt ewig an alten Systemen herum zu basteln. Wohlgermerkt, in etwas anderer Reihenfolge. Aber ich finde, so klingt es prägnanter.

7. Qualifikation

All das geht natürlich nur mit einem motivierten, eingespielten Team, das sich alle vorherigen Punkte zu eigen gemacht hat und auch vorbehaltlos an einem Strang zieht. Dass das Prozessmodell lebt, ist Voraussetzung für alles. Das Team muss die Transparenz mittragen und das dazu notwendige Reporting verlässliche Informationen liefern. In den Projekten müssen interne wie externe Kollegen aktiv zusammenarbeiten. Das Team muss Altlasten erkennen, kommunizieren und die notwendigen Zusammenhänge und Ideen für deren Bereinigung konstruktiv diskutieren. Das gesamte vorhandene Know-how bildet die Grundlage für die erfolgreiche Standardisierung. Dann kann man

nahezu beliebig innovative Lösungen besprechen – und gegebenenfalls sogar durchsetzen.

Die Kollegen brauchen Vertrauen, dass sie nicht abgehängt werden, und dass auch die schöne neue Welt sie mit ihren Fähigkeiten benötigt und fordert. Die zuvor beschriebenen Effizienzgewinne dürfen nicht bedrohlich erscheinen.

Erfahrungsgemäß sorgen zunehmend komplexe Lösungen dafür, dass genauso viel Arbeit wie vorher vorhanden ist. Es ändert sich jedoch die Qualität der Aufgaben. Nahezu allen derzeit in der IT vorhandenen Innovationstreibern ist gemein, dass sie zunehmend komplex und auch interdisziplinär sind. Die Trennung klassischer IT-Säulen wie *Windows*, *UNIX*, *Storage*, *Netzwerk* lässt sich in der hochgradig virtualisierten Welt nicht aufrecht erhalten. Das Plattform-Konzept (Server, Client, Anwendung – sobald man also architektonisch von Plattform spricht) hat nahezu immer interdisziplinäre Verschränkungen zur Folge. Wer das volle Potential zeitgemäßer IT-Konzepte entfalten will, muss die Trennung zwischen diesen Disziplinen aufheben.

Für Mitarbeiter bedeutet dies immer eine Aufwertung ihrer Rollen. Damit einher geht nahezu immer Schulung und Weiterbildung, um entstandene Wissenslücken zu schließen. Zunehmende Komplexität erfordert, verschiedenste Themen neben den ursprünglichen Kernkompetenzen im Blick zu behalten. Gefordert sind Menschen mit sogenannten T-Profilen, die sich zwar einzelne Themen in einer respektablen Tiefe erschlossen haben und beherrschen, die aber andere Themenbereiche abseits ihrer Kernaufgaben zumindest oberflächlich oder ambitioniert, wenn auch nicht spezialisiert im Blick haben, um neue Zusammenhänge zu erfassen und bearbeiten zu können. Dieses Selbstverständnis widerspricht in vielen etablierten IT-Organisationen einem lange geprägten Selbstverständnis der Spezialisierung auf ein Thema.

Daher muss man alle Kollegen wirklich auch konsequent auf diesen Weg der kontinuierlichen IT-Erneuerung mitnehmen. Mit Teams, die eine solche Bildungskultur leben und etabliert haben, kann man im Umkehrschluss nahezu jedes beliebige Innovationsprojekt in Angriff nehmen. So lange es mit Computern zu tun hat, wird man das Kind schon schaukeln.

Denkt mal drauf rum

Kommt all dies zusammen, sollte einem konkurrenzfähigen IT-Betrieb nichts mehr im Weg ste-

hen. In meinen persönlichen Benchmarks habe ich eigentlich immer ein Szenario darstellen können, das das marktübliche Selbstverständnis von Service-Level, Kosten und Aufwänden völlig aushebeln konnte. Allgemeinplätze wie „Cloud ist billig“ oder „Outsourcing steigert Qualität“ haben in den von mir geprägten Umgebungen tatsächlich keinen Bestand mehr. Meine wesentliche Erkenntnis ist:

Ein gut aufgestellter, qualifizierter Eigenbetrieb ist immer und unter allen Umständen billiger als eine einschlägige Cloud-Lösung bei egal welchem Service-Provider.

Das muss sie auch sein, da ein Service-Provider erhebliche Strukturen schaffen muss, um seinen vielen Mandanten gerecht zu werden. Das sind Strukturen, an die man als Kunde dann auch erst einmal organisatorisch angeschlossen werden muss. Darunter kocht auch der Provider nur mit Wasser und folgt diversen technischen Grundprinzipien, die sich – nur weil „Cloud“ auf der Broschüre steht – nicht in Luft auflösen.

Das gilt sicher nicht unbedingt für Peak-Workloads. Eine dedizierte Leerstandsbetrachtung gehört darum auf jeden Fall in die zu schaffende Transparenz. Das gilt auch nicht für spezifische Geschäftsmodelle im Internet. Natürlich gibt es hier berechtigte Ausnahmefälle. Pauschal gilt diese Aussage jedoch explizit für jeden Mischbetrieb.

Jetzt könnte man auf die Idee kommen, nur einzelne Workloads nach außen geben und ein gemischtes Sourcing-Konzept zu verfolgen. Das wiederum widerspricht normalerweise den Optimierungsgedanken und -anforderungen der eigenen Infrastruktur sowie dem dafür gewünschten Innovationsgeist. Man kann den Eigenbetrieb nur dann in Effizienzszenarien führen, wenn man die modernen IT-Plattform-Gedanken auch in kleine und mittlere Unternehmen trägt. Baut man dort eine Plattform, braucht auch diese möglichst hohe Auslastung und granulare Skalierbarkeit. Nur das minimiert Leerstände. Anderenfalls kannibalisiert die IT-Organisation ihre eigenen Deckungsbeiträge zum Aufbau moderner Lösungen. IT-Eigenbetriebe, die bei einem Wechsel zum Cloud-

Provider viel Geld sparen, sind meinen Auswertungen nach immer diejenigen, die entweder veraltete Betriebsmodelle oder beachtliche Leerstände in ihren Kapazitäten haben. Oft ist es eine Mischung aus beidem.

Hat man die genannten IT-Management-Themen konsequent abgearbeitet, dann darf man langsam auch Schlagwörter wie *Operational Excellence* oder *Private Cloud* im internen Selbstmarketing verwenden. Der Erfolg kann hinsichtlich Kosteneffizienz, Performance und Flexibilität eigentlich nicht ausbleiben.

Im Wesentlichen folge ich in meinem aktuellen Wirkungskreis seit zwei Jahren diesem Fahrplan. Reflektiere ich alle genannten Punkte, komme ich bei den derzeit abgeschlossenen Referenzkalkulationen für einzelne dedizierte IT-Services im Eigenbetrieb auf interne Betriebskosten, die durchweg zwischen 12 und 40 Prozent des günstigsten Marktbegleiters im Outsourcing betragen. Stichprobenartig habe ich hierzu Dienste angefragt und Angebote verglichen. Auf interner Seite sind alle relevanten Kosten umgelegt, und es unterliegt auch ein Kostenmodell zur Service-Verrechnung, das umlagebezogen, fair und vollständig ist und die Kalkulation für nahezu alle intern erbrachten IT-Services erlaubt.

Diese Effizienz muss ein externer Anbieter erst einmal aufzeigen, geschweige denn kostenseitig treffen. Sollte ich in einem einzelnen Service unwirtschaftlicher im Eigenbetrieb sein, reduziert mir dieser eine Service noch immer den eigenen Leerstand, und nicht den des Providers.

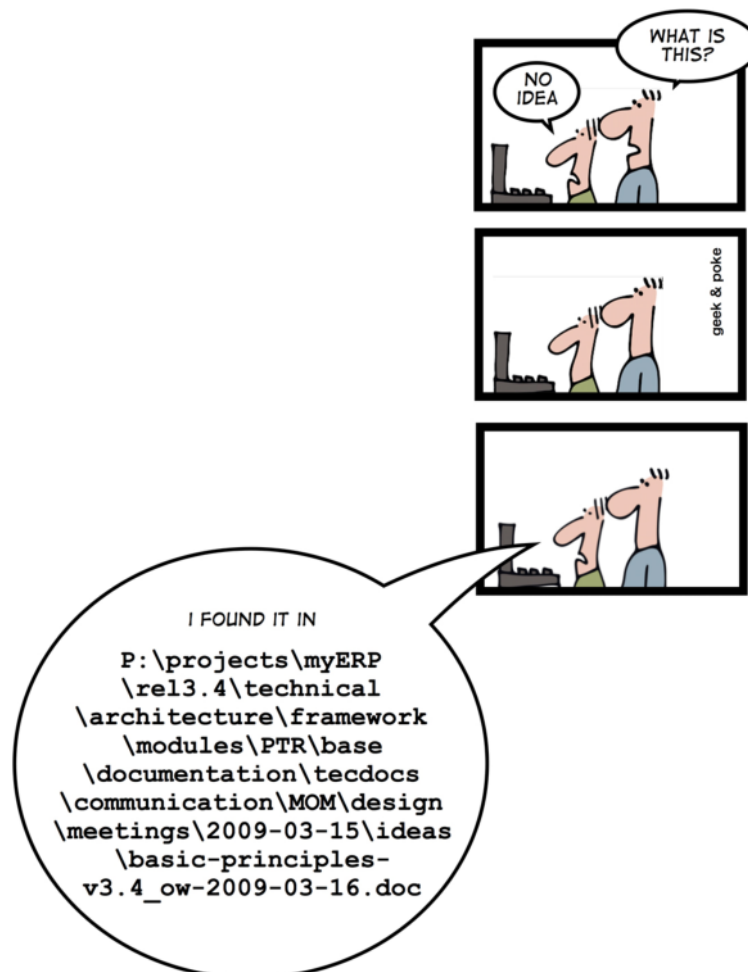
Alles in allem spricht also sehr viel für die konsequente Modernisierung von IT-Abteilungen. Jeder IT-Betrieb hat diese Potentiale und sollte sie konsequent ausschöpfen. Dann wären auch die gefühlten intellektuellen Abstände zu dem, was im angloamerikanischen Ausland IT-mäßig angestellt wird, nicht mehr so gigantisch.

Dies alles liest sich so vielleicht etwas allgemein, da ich keine spezifischen Einzelheiten wiedergeben möchte. Nichtsdestotrotz basiert jeder einzelne Punkt auf konkreten, selbst erlebten und selbst gemachten Erfahrungen. Es ist die Summe meiner Beobachtungen in der operativen IT-Branche aus den letzten 15 Jahren. Insofern: Denkt mal drauf rum.

Über Frank



Frank Benke leitet die IT in der Unternehmensgruppe *HAHN Automation*, einem globalen Unternehmen für Sondermaschinenbau. Nach seinem Studium der Informatik führte er einige Jahre ein kleines Systemhaus, bevor er für zehn Jahre als Trainer und Consultant unter anderem für *Hewlett-Packard*, *BASF* und *Continental* seinen Schwerpunkt in der IT-Infrastruktur fand. Neben seinen organisatorischen Aufgaben in der Abteilungsleitung beschäftigt er sich aktuell mit den technischen Schwerpunkten Industrie 4.0, *Hyperconverged Infrastructure* und *Software Defined Infrastructure*, *Virtual Desktop Infrastructure* und *Virtual Prototyping*. Frank engagiert sich in der *VMUG Deutschland West* (<https://www.vmug.com/germanywest>). Sein Blog ist unter <http://www.mybenke.org> zu finden.



DEEP IN THE CATACOMBS OF AN IT-PROJECT

Hilfreiches für alle Beteiligten Autorenrichtlinien

Selbst etwas für die UpTimes schreiben? Aber ja! Als Thema ist willkommen, was ein GUUG-Mitglied interessiert und im Themenbereich der GUUG liegt. Was sonst noch zu beachten ist, steht in diesen Autorenrichtlinien.

Der Schriftsteller ragt zu den Sternen empor,
Mit ausgefranstem T-Shirt.
Er raunt seiner Zeit ihre Wonnen ins Ohr,
Mit ausgefranstem T-Shirt.

Frei nach Frank Wedekind, Die Schriftstellerhymne

Wir sind an Beiträgen interessiert. Wir, das ist diejenige Gruppe innerhalb der GUUG, die dafür sorgt, dass die UpTimes entsteht. Dieser Prozess steht jedem GUUG-Mitglied offen. Der Ort dafür ist die Mailingliste <redaktion@uptimes.de>.

Welche Themen und Beitragsarten kann ich einsenden?

Die UpTimes richtet sich als Vereinszeitschrift der GUUG an Leser, die sich meistens beruflich mit Computernetzwerken, IT-Sicherheit, Unix-Systemadministration und artverwandten Themen auseinandersetzen. Technologische Diskussionen, Methodenbeschreibungen und Einführungen in neue Themen sind für dieses Zielpublikum interessant, Basiswissen im Stil von *Einführung in die Bourne Shell* hingegen eher nicht. Wer sich nicht sicher ist, ob sein Thema für die UpTimes von Interesse ist, kann uns gern eine E-Mail an <redaktion@uptimes.de> schicken.

Neben Fachbeiträgen sind Berichte aus dem Vereinsleben, Buchrezensionen, Konferenzberichte, humoristische Formen und natürlich Leserbriefe interessant. Wer nicht gleich mehrseitige Artikel schreiben möchte, beginnt also mit einem kleineren Beitrag.

Fachbeiträge sind sachbezogen, verwenden fachsprachliches Vokabular und anspruchsvolle Erläuterungen, besitzen technische Tiefe und ggf. auch Exkurse. Berichte aus dem Vereinsleben greifen aktuelle Themen auf oder legen Gedankengänge rund um die GUUG und ihre Community dar. Konferenzberichte zeigen, welche Veranstaltungen jemand besucht hat, was er/sie dort erfahren hat und ob die Veranstaltung nach Meinung des Autors beachtenswert oder verzichtbar war. Unterhaltsame Formen können ein Essay oder eine Glosse sein, aber auch Mischformen mit Fach-

artikeln (Beispiel: der „Winter-Krimi“ in Ausgabe 2013-3). Auch unterhaltsame Formen besitzen jedoch inhaltlichen Anspruch. Denn die UpTimes ist und bleibt die Mitgliederzeitschrift eines Fachvereins.

In der UpTimes legen wir daher auch Wert auf professionelle publizistische Gepflogenheiten und einheitliche Schreibweisen. Dafür sorgt zum Beispiel ein einheitliches Layout der Artikel, oder etwa die grundsätzliche Vermeidung von Worten in Großbuchstaben (entspricht typografisches Schreien) oder von Worten in Anführungsstrichen zum Zeichen der Uneigentlichkeit (entspricht Distanzierung von den eigenen Worten). Wichtig sind außerdem beispielsweise Quellenangaben bei Zitaten, Kenntlichmachung fremder Gedanken, Nachvollziehbarkeit der Argumentation sowie Informationen zum Autor nach dem Artikel.

In welchem Format soll ich meinen Artikel einsenden?

ASCII: Am liebsten blanke UTF8-Texte. Gern mit beschreibenden Anmerkungen oder Hinweisen (zum Beispiel mit Prozentzeichen zum Kenntlichmachen der Meta-Ebene).

L^AT_EX: Wir setzen die UpTimes mit L^AT_EX. Weil wir – wie es sich beim Publizieren gehört – mehrspaltig setzen und ein homogenes Erscheinungsbild anstreben, verwenden wir für die UpTimes bestimmte Formatierungen. Es ist nicht erwünscht, eigene Layoutanweisungen einzusenden. Wir behalten uns vor, Texte für die Veröffentlichung in der UpTimes umzuformatieren. Eine Vorlage mit den von uns verwendeten Auszeichnungen für Tabellen, Kästen und Abbildungen gibt es unter <http://www.guug.de/uptimes/artikel-vorlage.tex>.

Listings: Der mehrspaltige Druck erlaubt maximal 45 Zeichen Breite für Code-Beispiele, inklusive 1 Leerzeichen und einem Zeichen für den Zeilenumbruch innerhalb einer Code-Zeile (Backslash). Breitere Listings formatieren wir um, verkleinern die Schriftgröße oder setzen sie als separate Abbildung.

Bilder: Wir verarbeiten gängige Bildformate, soweit ImageMagick sie verdaut und sie hochauflösend sind. Am besten eignen sich PNG- oder PDF-Bilddateien. Plant bei längeren Artikeln mit 1 Abbildung pro 3000 Zeichen. Das müssen nicht Bilder sein, sondern auch Tabellen, Listings oder ein Exkurskasten sind möglich. Verseht Eure Bilder nicht mit Rahmen oder Verzierungen, weil die Redaktion diese im UpTimes-Stil selbst vornimmt.

Wie lang kann mein Artikel sein?

Ein einseitiger Artikel hat mit zwei Zwischentiteln um die 2.700 Anschläge. Mit etwa 15.000 Anschlägen – inklusive 3 Abbildungen – landet man auf rund vier Seiten. Wir nehmen gern auch achtseitige Artikel, achten dabei aber darauf, dass der Zusammenhang erhalten bleibt und dass es genug Bilder gibt, damit keine Textwüsten entstehen.

Wer Interesse hat, für die UpTimes zu schreiben, macht sich am besten um die Zeichenzahl nicht so viele Gedanken – auch für kurze oder lange Formate finden wir einen Platz. Die Redaktion ist bei der konkreten Ideenentwicklung gern behilflich. Für eine Artikelidee an [<redaktion@uptimes.de>](mailto:redaktion@uptimes.de) reicht es, wenn Ihr ein bestimmtes Thema behandeln wollt.

Wohin mit meinem Manuskript?

Am einfachsten per E-Mail an [<redaktion@uptimes.de>](mailto:redaktion@uptimes.de) schicken. Das ist jederzeit möglich, spätestens jedoch vier Wochen vor dem Erscheinen der nächsten UpTimes. Zum Manuskript ist ein kleiner Infotext zum Autor wichtig, ein Bild wünschenswert.

Nützlich ist, wenn der Text vor Einsendung durch eine Rechtschreibkorrektur gelaufen ist. `aspell`, `ispell` oder `flyspell` für Textdateien sowie die von LibreOffice bieten sich an. Wenn Ihr Euren Text an die Redaktion schickt, solltet Ihr also weitestmöglich bereits auf die Rechtschreibung geachtet haben: Nach der Einschickung ist Rechtschreibung und Typo-Korrektur Aufgabe der Redaktion. Die Texte in der UpTimes folgen der neuen deutschen Rechtschreibung.

Wie verlaufen Redaktion und Satz?

Wir behalten uns vor, Texte für die Veröffentlichung in der UpTimes zu kürzen und zu redigieren. Das bedeutet, dafür zu sorgen, dass der Artikel nicht ausufert, versehentliche Leeraussagen wegfallen, Syntax und Satzanschlüsse geglättet werden, dass Passiva und Substantivierungen verringert und Unklarheiten beseitigt werden (die zum Beispiel Fragen offen lassen oder aus Passivkonstruktionen resultieren, ohne dass der Schreibende das merkt). Manchmal ist dieser Prozess mit Nachfragen an den Autoren verbunden.

Die endgültige Textversion geht jedem Autoren am Ende zur Kontrolle zu. Dabei geht es um die inhaltliche Kontrolle, ob sich durch den Redaktionsprozess Missverständnisse oder Falschaussagen entstanden sind. Danach setzt die Redaktion die Artikel. Wenn der Satz weitgehend gediehen ist – also ein *Release Candidate* als PDF vorliegt – erhalten die Autoren als erste diesen RC. Danach wird die UpTimes dann veröffentlicht.

Gibt es Rechtliches zu beachten?

Die Inhalte der UpTimes stehen ab Veröffentlichung unter der CC-BY-SA-Lizenz, damit jeder Leser die Artikel und Bilder bei Nennung der Quelle weiterverbreiten und auch weiterverarbeiten darf. Bei allen eingereichten Manuskripten gehen wir davon aus, dass der Autor sie selbst geschrieben hat und der UpTimes ein nicht-exklusives, aber zeitlich und räumlich unbegrenztes Nutzungs- und Bearbeitungsrecht unter der CC-BY-SA einräumt.

Bei Fotos oder Abbildungen Dritter ist es rechtlich unabdingbar, dass der Autor sich bei dem Urheber die Erlaubnis zu dieser Nutzung einholt, und fragt, wie die Quelle genannt zu werden wünscht. Die Frage nach der CC-BY-SA ist hierbei besonders wichtig.

An Exklusivrechten, wie sie bei kommerziellen Fachzeitschriften üblich sind, hat die UpTimes kein Interesse. Es ist den Autoren freigestellt, ihre Artikel noch anderweitig nach Belieben zu veröffentlichen.

Bekomme ich ein Autorenhonorar?

Für Fach- und literarische Beiträge zahlt die GUUG dem Autor nach Aufforderung durch die Redaktion und Rechnungstellung durch den Autor pro Seite 50 € zuzüglich eventuell anfallender USt. Beiträge für die Rubrik „Vereinsleben“,

Buchrezensionen und Artikel bezahlter Redakteure sind davon ausgenommen. Gleiches gilt für Pa-

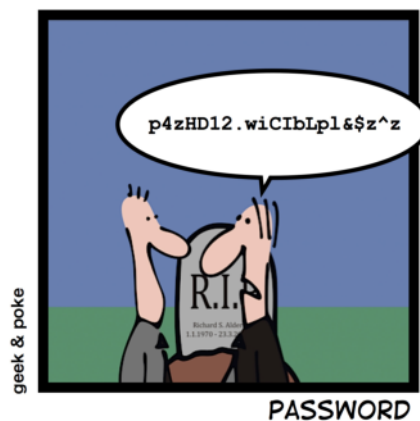
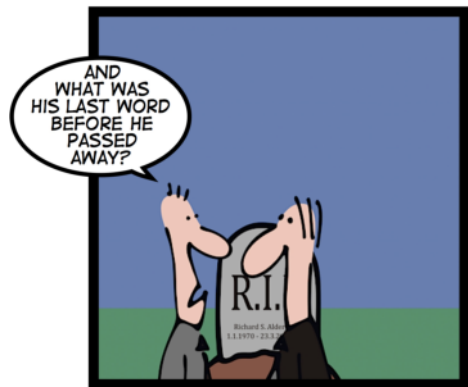
per, wenn die UpTimes die Proceedings der Konferenz enthält.



Nächste redaktionelle Ausgabe: UpTimes 2016-2, Winter-Ausgabe

- Redaktionsschluss: Sonntag, 6. November 2016.
- Erscheinung: Sonntag, 18. Dezember 2016.
- Gesuchte Inhalte: Fachbeiträge über Unix und verwandte Themen, Veranstaltungsberichte, Rezensionen, Beiträge aus dem Vereinsleben.
- Manuskript-Template: <http://www.guug.de/uptimes/artikel-vorlage.tex>
- Fragen, Artikelideen und Manuskripte an: <kehrer@guug.de> oder an die Redaktionsmailingliste <redaktion@uptimes.de>

SIMPLY EXPLAINED



Über die GUUG German Unix User Group e.V.

Vereinigung deutscher Unix-Benutzer

Die Vereinigung Deutscher Unix-Benutzer hat gegenwärtig rund 700 Mitglieder, davon etwa 90 Firmen und Institutionen.

Im Mittelpunkt der Aktivitäten der GUUG stehen Konferenzen. Ein großes viertägiges Event der GUUG hat eine besondere Tradition und fachliche Bedeutung: In der ersten Jahreshälfte treffen sich diejenigen, die ihren beruflichen Schwerpunkt im Bereich der IT-Sicherheit, der System- oder Netzwerkadministration haben, beim *GUUG-Frühjahrsfachgespräch* (FFG).

Seit Oktober 2002 erscheint mit der *UpTimes* – die Sie gerade lesen – eine Vereinszeitung. Seit 2012 erscheint die *UpTimes* einerseits zu jedem FFG in Form einer gedruckten Proceedings-Ausgabe (ISBN), und andererseits im Rest des Jahres als digitale Redaktionsausgabe (ISSN). Daneben erhalten GUUG-Mitglieder zur Zeit die Zeitschrift *LANline* aus dem Konradin-Verlag kostenlos im Rahmen ihrer Mitgliedschaft.

Schließlich gibt es noch eine Reihe regionaler Treffen (<http://www.guug.de/lokal>): im Rhein-Ruhr- und im Rhein-Main-Gebiet sowie in Berlin, Hamburg, Karlsruhe und München.

Warum GUUG-Mitglied werden?

Die GUUG setzt sich für eine lebendige und professionelle Weiterentwicklung im Open Source-Bereich und für alle Belange der System-, Netzwerkadministration und IT-Sicherheit ein. Wir freuen uns besonders über diejenigen, die bereit sind, sich aktiv in der GUUG zu engagieren. Da die Mitgliedschaft mit jährlichen Kosten

Fördermitglied	350 €
persönliches Mitglied	70 €
in der Ausbildung	30 €

verbunden ist, stellt sich die Frage, welche Vorteile damit verbunden sind?

Neben der Unterstützung der erwähnten Ziele der GUUG profitieren Mitglieder auch finanziell davon, insbesondere durch die ermäßigten Gebühren bei den Konferenzen der GUUG und denen anderer europäischer UUGs. Mitglieder bekommen außerdem *c't* und *iX* zum reduzierten Abopreis.

Wie GUUG-Mitglied werden?

Füllen Sie einfach das Anmeldeformular unter <https://www.guug.de/verein/mitglied/> aus und schicken Sie es per Fax oder Post an die unten auf dem Formular angegebene Adresse.

Impressum

Uptimes – Mitgliederzeitschrift der
German Unix User Group (GUUG) e.V.

Herausgeber: GUUG e.V.

Grube-Nassau-Straße 3

D-56462 Höhn

E-Mail: <redaktion@uptimes.de>

Internet: <http://www.guug.de/uptimes/>

Autoren dieser Ausgabe: Anika Kehrer, Nils Magnus, Lenz Grimmer,
Hartmut Streppel, Patrick Koetter, Jürgen Plate und Anika Kehrer, Jürgen
Plate, Frank Benke

V.i.S.d.P: Anika Kehrer

Oslostr. 9

D-81829 München

Chefredaktion: Anika Kehrer

Redaktion: Mathias Weidner, Wolfgang Stief

LaTeX-Layout (PDF): Robin Därmann

XHTML-Layout (ePub): Mathias Weidner

Titelgestaltung: Hella Breitkopf

Bildnachweis: Titelbild *Rackschraube* von Hella Breitkopf. Comicreihe
geek & poke von Oliver Widder, CC BY SA, mit zusätzlicher freundli-
cher Genehmigung von Oliver Widder. Andere Quellennachweise am
Bild, sofern nicht unter CC BY SA.

Verlag: Lehmanns Media GmbH, Hardenbergstraße 5, 10623 Berlin

ISSN: 2195-0016

Für Anzeigen in der UpTimes wenden Sie sich bitte an <werbung@guug.de>.

Alle Inhalte der UpTimes stehen, sofern nicht anders angegeben, unter der CC BY SA
(<https://creativecommons.org/licenses/by-sa/3.0/de/>).

Alle Markenrechte werden in vollem Umfang anerkannt.