

Interview mit Marei Peischl

TeX/LaTeX – der „Nerd-Faktor“ ist definitiv hoch

IT Spektrum sprach mit der TeX/LaTeX-Expertin Marei Peischl über TeX, Donald E. Knuth, computergesteuerten Satz und warum TeX heute noch viele Anwendungen hat, wieso es nahezu fehlerfrei ist und welchen Nerd-Faktor es braucht, um damit zu arbeiten.



Johannes Mainusch: Wir sind uns kürzlich auf dem Tech-Camp in Hamburg begegnet. Du wurdest mir als TeX-Expertin vorgestellt. Lass uns mal ganz vorne anfangen, denn möglicherweise gibt es Leser, die TeX nicht kennen. Was also ist TeX?

Marei Peischl: TeX ist ein uraltes Programm aus den 70ern. Es ist ein Textsatzsystem, das darauf abzielt, Buchstaben möglichst sinnvoll auf einer Seite zu platzieren. Als ich an der Uni war, hieß es oft, TeX benutze man ausschließlich für naturwissenschaftliche Dokumente, da es ursprünglich für den Formelsatz entwickelt wurde.

Also, wenn man beispielsweise das Integral von x^2 im Bereich von -1 bis $+1$ setzen möchte?

$$f(x) = \int_{-1}^1 x^2 dx$$

Fragen, die wir in der Physik immer nervig finden, aber genau. Die Grundidee dahinter ist, Formeln und Schrift generell schön darstellen zu können. Donald Knuth, von dem vielleicht einige schon gehört haben, ist der Autor des legendären Informatik-Buchs „The Art of Computer Program-

ming“ (TAOCP). Er ärgerte sich darüber, dass es Anfang der 70er-Jahre kein Programm gab, das einen guten Satz automatisch ermöglichte. Bis dahin wurden wissenschaftliche Dokumente meist von Hand gesetzt und mit den wenigen vorhandenen Programmen war er unzufrieden. Also entschied er sich, die Software selbst zu entwickeln, um ein vernünftig aussehendes Buch schreiben zu können.

Und das TeX-System, das er dafür schrieb, ist fehlerfrei?

Witzigerweise kann es als eines der wenigen Programme von sich behaupten, na-

hezu fehlerfrei zu sein. Es kann zwar sein, dass noch ein Fehler drin ist, aber dann hat ihn bisher noch niemand gefunden. Die Grundidee war ursprünglich, dass man für das Finden eines Bugs Geld bekommt. Anfangs 2,56 Dollar, wobei sich die Belohnung eine Zeit lang jedes Jahr verdoppelte. Irgendwann meldete keiner mehr etwas, und es ging nur noch um Kleinigkeiten wie Rechtschreibfehler in der Dokumentation. Donald Knuth hat sich wahnsinnig viele Gedanken darüber gemacht, wie er TeX aufbaut, bevor er es entwickelt hat. Und er hat sehr früh eine Community zum Debuggen animiert. Auch darüber hat er lesenswerte Artikel geschrieben.

Hast du das Programm mal gesehen? Kann man den Quellcode irgendwo finden?

Das kann man. Ich habe den auch tatsächlich als Buch gedruckt, da ist die Doku und alles mit drin. Knuth war ein Verfechter des Literate Programming, bei dem Quellcode und Dokumentation eng miteinander verknüpft sind, was die Lesbarkeit und Wartbarkeit des Codes erhöht. Das liest sich tatsächlich sehr schön.

In welcher Programmiersprache hat er das geschrieben?

Die Programmiersprache heißt WEB. Die hat er auch gleich miterfunden. Die mitt-

Donald E. Knuth entwickelte das Textsatz-System *TeX* 1978, als er an seinem Buch „The Art of Computer Programming“ arbeitete und unzufrieden mit den vorhandenen Werkzeugen für den Satz von mathematischen Formeln und komplexen Texten war. Er wollte ein System schaffen, das hochwertige Druckergebnisse liefern konnte und speziell auf die Bedürfnisse von technisch-wissenschaftlichen Publikationen zugeschnitten war.

LaTeX, eine Erweiterung von *TeX*, wurde in den 1980er-Jahren von Leslie Lamport entwickelt, um das Schreiben und Verwalten von Dokumenten zu erleichtern. Während *TeX* hauptsächlich eine Textsatz-Engine ist, stellt *LaTeX* eine Reihe von Makros bereit, die das Erstellen von Dokumenten erheblich vereinfachen (TeX Users Group, tug.org/texlive/Contents/live/readme.html.dir/readme.de.html, TeXwelt, texwelt.de/fragen/786/was-ist-der-unterschied-zwischen-tex-und-latex, Wikipedia, de.wikipedia.org/wiki/TeX). Diese Erweiterungen von *TeX* haben dazu beigetragen, dass *LaTeX* sich als Standardwerkzeug für wissenschaftliches Schreiben und Dokumentation etabliert hat.

I have been paying a reward to everyone who discovers new bugs in TEX, and doubling the amount every year. Last December I made two payments of \$40.96 each, and my chequebook has been hit for five \$81.92 payments in recent months. I am desperately hoping that this incentive to discover the final bugs will produce them before I am unable to pay the promised amount.
(Surely in 1998 I won't be writing cheques for \$83,886.08?)



aus: Donald E. Knuth, The Errors of TEX, in: Software: Practice and Experience, Wiley, Vol. 19(7), July 1989, S. 641, siehe auch: yurichev.com/mirrors/knuth1989.pdf

lerweile modernisierte Form davon heißt CWEB.

Also hat Knuth auch noch eine Programmiersprache gebaut, mit der er TeX programmieren kann, um dann sein geschriebenes TAOCP-Buch richtig setzen zu können? Was für ein Nerd!

Er macht halt bei allem seinen eigenen Kram, baut neue Dinge, um damit andere Dinge zu tun. Knuth ist ein Urgestein der Informatik, der vom Konzept bis zur Umsetzung alles wahnsinnig durchdenkt. Das hat mich an seiner Arbeit sehr früh wahnsinnig beeindruckt. Als Physikerin will ich auch immer wissen, wie alles funktioniert.

Und wie erfolgte die Bezahlung für so einen Fehlerfund?

Per Papier-Scheck. Diese Schecks sind irgendwann so berühmt geworden, dass Leute Fotos damit gepostet haben und es sogar Fälschungen gab. Deshalb bekommt man seit 2006 keinen echten Scheck mehr, sondern Donald Knuth erfand eine hypothetische Bank, die „Bank of San Serife“, und stellte die Schecks darüber aus.

Wie in sans serif, also wie eine schnörkellose Schrift?

Ja, genau. Man erhält von dieser Bank of San Serife für einen entdeckten Fehler einen Scheck und gleichzeitig wird auf der Webseite der Stanford Uni eingetragen, wie viele Dollars man als Guthaben hat. Dabei sind auch die Dollars speziell, denn ein Knuth Dollar sind 256 US Cent. Ein Guthaben von $0x\$b40.28$ im hexadezimalen System von Knuth Dollar wären also 7373.20 USD.

Viel von der TeX-Geschichte fand vor deiner Zeit statt, was brachte dich dazu?

Ich bin mir nicht mehr ganz sicher, wann genau ich angefangen habe, vielleicht mit 16, also ungefähr 2009. Am Ende der Schulzeit habe ich einmal pro Woche einen Tag als Frühstudentin an der Uni verbracht, um weniger in die Schule zu gehen. Meine Kommilitonen an der Uni sagten mir, dass sie TeX gerne für ihre Facharbeiten in der Schule genutzt hätten, wenn sie damals davon gewusst hätten ...

Weil die davor mit MS-Word versucht haben, ihre Hausaufgaben zu schreiben?



Marei Peischl

30 Jahre, lebt in Hamburg und wurde hier unter anderem Expertin im Backen von Franzbrötchen
Studium: Physik

Interessen: Segelboote, aktives Mitglied im Chaos Computer Club, verbringt ihren Jahresurlaub oft beim Auf- und Abbau von Chaos-Veranstaltungen in der Internetmanufaktur (das Netzwerkteam für den Chaos Communication Congress und die Camps)

Engagement im TeX/LaTeX-Umfeld:

- Unternehmen peiTeX
- LaTeX3 Programming Layer Tutorial
- DEPP: Mit diesem Experten-Werkzeug können die standardmäßig sehr umfangreichen LaTeX-Installationen auf das notwendige Maß verkleinert werden. DEPP (Dependency Printer for TeX Live) analysiert die Abhängigkeiten Ihrer TeX-Dokumente und ermöglicht es, nur die tatsächlich benötigten Pakete zu installieren und zu verwalten, siehe: gitlab.com/islandoftex/texmf/depp
- Diverse Veröffentlichungen und Vorträge im LaTeX-Umfeld, siehe auch: Einführung in LaTeX2, peitex.de/materialien/LaTeX-Kurskript.pdf

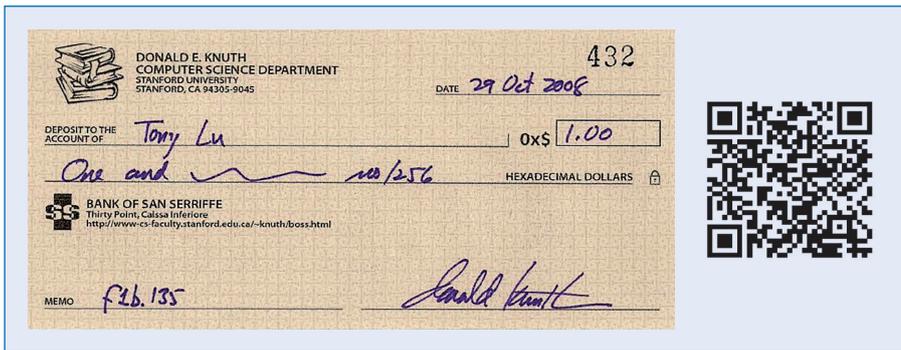
Ich war selbst kurz davor, meine eigene Facharbeit in der Schule zu schreiben, und ich fragte meine Kommilitonen oft, was sie anders machen würden, wenn sie das Wissen von heute schon vor dem Abi gehabt hätten. Meine Handschrift war schrecklich, und ab da war klar, dass ich alles mit TeX machen würde.

Mit Textverarbeitungsprogrammen wie Word oder Google Docs waren also bestimmte Funktionen damals nicht möglich.

Auch heute noch sind bestimmte Dinge in modernen Textverarbeitungsprogrammen problematisch, wie saubere Seitenumbrü-



Stanford Lecture: Don Knuth – „Pi and The Art of Computer Programming“ (2019), siehe: www.youtube.com/watch?v=3DKo219ZHMw&list=PLoROMvodv4r0AaVAVR_dyCigSBMcYjevYB&index=1



The Bank of San Serriffe mit Empfängerliste (www-cs-faculty.stanford.edu/~knuth/boss.html) und Scheck (commons.wikimedia.org/wiki/File:Knuth_cod.JPG)

che oder das korrekte Einfügen von Bildern. Die Grundidee bei TeX ist, das Layout vom Inhalt zu entkoppeln. Obwohl viele Leute immer noch im Layout herumbasteln, ermöglicht dieser Ansatz, sich auf den Inhalt zu konzentrieren und das Layout getrennt zu behandeln. Das ist der Gegensatz zu den WYSIWYG-Editoren und ermöglicht so effizientes Arbeiten, besonders bei Schriftstücken mit vielen Seiten. So ein gutes Layout hat einfach eine bessere Wirkung.

Das heißt, wenn ich eine Kapitelüberschrift schreibe, dann mache ich so was wie `\section{NameDesKapitels}`?

Klassisch ja. Das kann man noch erweitern, indem man Objekte wie das Inhaltsverzeichnis getrennt steuert, aber das sind Details. Letztendlich geht es darum, einem Text eine Eigenschaft zuzuweisen und ihn dann entsprechend zu verarbeiten. Ein weiterer Vorteil, der mit Word schwierig ist, ist die Möglichkeit, Dokumente sauber in eine Versionskontrolle für Code einzupflegen, denn es handelt sich beim Dokument ja nur um Text.

Man kann also die Änderungsverfolgung zwischen Dokumenten einfacher machen?

Ja, und ich kann aus einer Input-Datei verschiedene Outputs erzeugen. Zum Beispiel kann ich aus einem Input sowohl Folien als auch ein Begleitskript mit Fließtext erstellen. Ich kann Textbausteine einfügen, die nur in einem der Outputs erscheinen, und Verlinkungen innerhalb des Dokuments ermöglichen, wie in einem Inverted Classroom, wo man von einer Übungsaufgabe direkt zur Lösung springen kann. Das alles bleibt in einer Quelldatei, wodurch ich Fehler vermeide, die durch das Bearbeiten mehrerer Dateien entstehen könnten.

Jetzt warst du also an der Schule in deinem Uni, bist TeX begegnet und hast

dich damit angefreundet. Aber wie kam es dann von dieser Begeisterung zum Hauptberuf?

Da bin ich so reingestolpert. Im ersten Semester, in dem ich Vollzeit an der Uni war, brauchten sie Tutoren für einen TeX-Kurs. Meine Uni hatte einen Block-Kurs, bei dem die Studierenden eine Woche lang intensiv betreut wurden. Ich habe das ein Semester gemacht, dann hörte der Kursleiter auf, und ich wurde gefragt, ob ich den Kurs weiterführen möchte. Ich hatte etwas Panik, denn die Teilnehmerzahl wurde von 60 auf 90 erhöht. Ich war im 2. Semester und hatte keine Ahnung, wie alles läuft. Ich hatte ein Skript, das mir nicht gefiel, also habe ich mein Semester damit verbracht, es komplett zu überarbeiten. Die überarbeitete Version wurde veröffentlicht, und seit 2011 aktualisiere ich es laufend.

Also musstest du schneller lernen als deine Studenten?

Ja, ich kann nicht etwas verkaufen, hinter dem ich nicht stehe. Daher habe ich versucht, es besser und anders zu machen. Statt nur Codeschnipsel zu geben, lehrte ich Konzepte und verschob den Fokus, um die Struktur klarzustellen. Beruflich bin ich Physikerin. Ich wollte Dinge erst ausprobieren, bevor ich sie bewerte. Neben dem Studium machte ich Kurse, und mein damaliger Chemie-Professor bot mir an, bei seinem Buchprojekt zu helfen. Er schlug vor, ein Gewerbe anzumelden, da er mehr als 7,50 Euro pro Stunde zahlen wollte. So wurde ich selbstständig.

Welche Bedeutung hat TeX heutzutage? Wer verwendet das?

TeX und LaTeX sind besonders beliebt in Bereichen, in denen präzise Formatierung und die Handhabung komplexer mathematischer und wissenschaftlicher

Inhalte wichtig sind. Oder für Prozesse mit viel Automatisierung. In Verlagen wird LaTeX weniger genutzt, ich denke, hauptsächlich aufgrund der Komplexität und des Fachwissens, das erforderlich ist. Viele Verlage finden XML-basierte Systeme praktischer, da sie auf den ersten Blick intuitiver zu nutzen sind. Trotzdem bietet LaTeX Vorteile, wie die einfache Nachverfolgung von Änderungen zwischen Auflagen und effiziente Layout-Umstellungen. Die Schwierigkeit liegt darin, ein funktionierendes System für Prüfmechanismen und Formatierungen zu etablieren. Ich arbeite mit LaTeX vor allem für technische Dokumentationen, da es eine modulare Struktur ermöglicht, die besonders bei Mehrsprachigkeit effizient ist. Außerdem erstelle ich Corporate Design Templates für Universitäten und Forschungsinstitute.

Hilft TeX dabei, Schriftstücke in einem Fachbereich einheitlich zu gestalten?

Ja, genau. So wie man mit PowerPoint-Templates Folien einheitlich gestalten kann, kann man das auch mit TeX machen. Wissenschaftler:innen können so auf ein einheitliches Layout zugreifen, was es einfacher macht, Dokumente konsistent zu halten. Das ist besonders praktisch, wenn man verschiedene Layouts für unterschiedliche Zwecke braucht, aber den gleichen Inhalt behalten möchte. Außerdem ist TeX nützlich für Branchen wie Banken oder Versicherungen, die strenge Vorgaben für Formatierung und Änderungsmarkierungen haben. Momentan arbeiten wir auch daran, Dokumente barrierefrei zu gestalten, damit sie für alle zugänglich sind.

Wie unterstützt dich LaTeX dabei?

LaTeX gibt mir schon eine solide Grundstruktur mit einer klaren Semantik. Frühe PDF-Formate hatten das nicht, sie waren ziemlich simpel. Zum Beispiel, wenn ich einen Link zu einer Überschrift setze, hatte das PDF-Format anfangs keine semantische Verbindung zwischen diesen Elementen. Das macht es schwierig, wenn man mit einem Screenreader arbeitet, da keine logische Beziehung zwischen den Teilen des Dokuments besteht. LaTeX hilft, diese Struktur und Semantik von Anfang an zu etablieren. LaTeX hilft auch bei der Einhaltung von Standards. Ich habe beispielsweise Pakete für die Einhaltung von Schweizer Rechnungsstandards gemacht, so zum Beispiel die Schweizer QR-Rechnung.

Wie wird aus einem TeX-Dokument ein PDF oder ein Schriftstück? Wird das kompiliert?

Das hängt davon ab, wie man „kompilieren“ versteht. Ursprünglich bedeutet kompilieren, Dinge zusammenzufassen und ein Endprodukt daraus zu erstellen. In der Softwarewelt funktioniert es etwas anders: TeX selbst ist eine Makrosprache, die expandiert wird. Also nur aus Ersetzung von Zeichenketten besteht. Es gibt ein paar grundlegende Funktionen, aber im Wesentlichen wird alles andere nur aus deren Verkettung erzeugt. TeX fasst also den tatsächlichen Inhalt mit den Formatierungsbefehlen zusammen und schreibt das dann je nach Konfiguration in eine PDF.

Ich hätte nicht gedacht, dass TeX und LaTeX noch so lebendig sind. Glaubst du, dass TeX irgendwann verschwinden wird, oder wird es lange bestehen bleiben?

Das hängt davon ab, wie man es nutzt. TeX und LaTeX haben das Problem, dass viele Leute es verwenden, ohne wirklich zu verstehen, wie es funktioniert. Oft fügen sie einfach ein paar Codeschnipsel ein

und wundern sich dann über kryptische Fehlermeldungen. Das führt dazu, dass sie frustriert sind und viel Zeit mit Fehlersuche verbringen, statt sich auf ihre Inhalte zu konzentrieren. Wenn man TeX jedoch richtig nutzt, mit gepflegten Templates und regelmäßigen Updates, kann es sehr leistungsfähig sein.

Es gibt auch viele Parallelentwicklungen, die versuchen, ähnliche Funktionen wie TeX zu bieten, aber oft nicht die gleiche Flexibilität haben. Die große Menge an Erweiterungen und Paketen im „Comprehensive TeX Archive Network“ (CTAN) und die aktive Community zeigen, dass TeX und LaTeX ein starkes Fundament haben, daher denke ich, dass sie definitiv weiterhin bestehen werden.

Marei, was wäre dein Wunsch an die gute Fee, der dann in 365 Tagen in Erfüllung geht?

Mein Wunsch wäre, dass mehr Leute erkennen, dass TeX und seine Pakete echte Software sind, die regelmäßige Pflege und Wertschätzung braucht. Es wäre super, wenn Verlage und Universitäten das verstehen und entsprechend unterstützen würden. Momentan wird oft erwartet,

dass TeX einfach funktioniert, ohne dass jemand dafür sorgt, und das führt zu Frustration und ineffizientem Arbeiten. Wenn wir gepflegte Templates und die nötigen Mittel hätten, könnten sich die Leute auf kreative und produktive Aufgaben konzentrieren, anstatt sich über Fehlermeldungen zu ärgern.

Dank Marei für das Gespräch.

Das Interview führte ...



Dr. Johannes Mainusch

(johannes.mainusch@kommitment.works)
Berater für Unternehmen, die Bedarf im Bereich IT, Architektur und agiles Management haben. Dr. Mainusch ist seit 2012 Mitglied der IT Spektrum-Redaktion.



Lernen Sie, die Qualität Ihrer Software nachhaltig zu verbessern und zu halten!

Alle Workshops mit Aufzeichnung – online & kostenlos

Workshop – Test-Gap-Analyse

Ungetestete Änderungen im Quelltext aufdecken

Die meisten Fehler treten bei langlebiger Software dort auf, wo viel geändert wird. In der Praxis gehen viele Änderungen versehentlich ungetestet in Produktion. Die Test-Gap-Analyse hilft Ihnen, diese vor dem Release zu finden. Erfahren Sie, wie die Analyse im Alltag funktioniert und welche Vorteile sie mit sich bringt!

25. September 24

10:30-12:00 Uhr

Sprache: Englisch

Anmeldung:

tmscl.me/tga-249-its



Workshop – Schnelles Feedback trotz langsamer Tests

Testselektion für historisch gewachsene Test-Suites

Mit einem Software-System wächst auch die Test-Suite – das kostet Ausführungszeit. Führt man eine Teilmenge der langsamen Test-Suite häufiger aus, bekommt man gezielt und schnell Feedback. Erfahren Sie, wie man diese Teilmenge mit Test-Impact-Analyse und Pareto-Testing effektiv auswählt und im Alltag nutzen kann!

23. Oktober 24

10:30-12:00 Uhr

Sprache: Englisch

Anmeldung:

tmscl.me/ts-2410-its

