

☐ „Ich bin ja kein Entwickler“

*Nicht nur Softwareentwickler*innen entwickeln Software! Was es in einem FOSS-Projekt sonst noch so zu tun gibt, erklärt Meike vom Project [Accidental Contributions](#) (Runde 16).*

Es ist rund 20 Jahre her, dass ich hinter den Vorhang Freier Software blickte. Das klang (und klingt heute noch) sehr verheißungsvoll: alle können mitentwickeln, können Verbesserungen und neue Funktionen einbauen oder Fehler beheben. Dem ersten Impuls „Ich bin dabei!“ folgte unmittelbar der zweite: „Ach nee, ich kann ja gar nicht programmieren.“ Auch das dürfte vielen heute nicht anders gehen.

Softwareentwicklung und Programmieren werden allzu oft als Synonyme gedacht, ein weit verbreiteter Irrglaube. Wie schön wäre es, wenn sich künftig ein dritter Impuls anschließen würde: „Aber ich habe andere Fähigkeiten; wo soll ich anfangen?“

Das Full-Stack-Multitalent – ein Modell von gestern

Ich bin noch in einer Zeit IT-sozialisiert worden, in der man von „Homepages programmieren“ sprach und weitgehend reine HTML-Projekte meinte. So als würde man „programmieren“, wenn man ein Textdokument strukturiert. Und es klang, als handele es sich dabei um Geheimwissen, für das man jahrelang studiert haben musste und nicht um etwas, das man sich an zwei Wochenenden im Selbststudium draufschaffen konnte. Meine zwei Wochenenden hatten mich also zur Geheimwissenden gemacht, und so konnte ich gut mein Studium finanzieren.

Daran muss ich heute denken, wenn ich die Ehrfurcht vor den Menschen mit Programmierkenntnissen höre. Nicht falsch verstehen, ich habe auch Respekt vor Programmierer*innen. Es ist ein Job, in dem man sich nicht auf Gelerntem ausruhen kann. Aber es sind eben auch Fähigkeiten, die dank freiem Wissen im Selbststudium erlernbar sind. (Gut, man braucht auch mehr als zwei Wochenenden.)

Diese Ehrfurcht schafft eine Kluft, die Menschen mit anderen Talenten davon abhalten mag, Softwareentwicklung als etwas zu betrachten, in das sie sich einbringen können. Deshalb schreibe ich diesen Blogpost: als Aufforderung zu mehr Selbstbewusstsein.

Natürlich gibt es Multitalente. Viele Programmierer*innen mussten zwangsläufig verschiedenste Gewerke beherrschen, schlicht weil es mangels besseren Wissens von ihnen erwartet wurde. In Stellenanzeigen finden wir immer noch vielfach nicht näher spezifizierte „Full-Stack-Entwickler“-Gesuche. Was auch immer man sich heute darunter vorstellen mag (sämtliche

Softwareentwicklung braucht viele Talente

Dröseln wir doch einmal auf, welche Rollen ein typisches Softwareentwicklungsprojekt benötigt. Ich konzentriere mich hier auf Open-Source-Projekte, die oftmals dezentral und asynchron entwickelt werden. Diese Liste ist selbstverständlich nicht abschließend und Rollen können auch von mehreren Personen (oder mehrere Rollen von einer Person) übernommen werden.

Konzeption

Konzepter*innen führen Bedarfsanalysen durch, übersetzen Ziele und Wünsche in lösbare Anforderungen, zeigen Widersprüche oder Lücken in der Vorplanung auf. Sie erstellen in Abstimmung mit weiteren Rollen ein Konzept. Hilfreich sind Talente in folgenden Bereichen:

- Empathie und Kommunikation
- Folgenabschätzung
- Texte schreiben
- Grundverständnis von Softwarebedienung
- Grundverständnis von Programmlogik

UX-Design

Nein, nicht „Design“. Hier geht es um die Gestaltung von Prozessen. UX steht für User Experience, zu deutsch: Benutzungserfahrung. Beabsichtigt ist eine *positive* Erfahrung. Dieses Gewerk hat einige Nähe zur Konzeption, ist aber typischerweise auch für die Visualisierung zuständig.

Als künftige Nutzende werden exemplarische „Personas“ entwickelt, deren Nutzungsszenarien die Entwicklung begleiten sollen. Für eine Anwendung mit grafischer Oberfläche werden beispielsweise optimale Klickpfade erarbeitet und der Ablauf in Form von Wireframes (Skizzen) grafisch dargestellt. Hilfreiche Fähigkeiten:

- Grundverständnis für Psychologie
- Empathie und Kommunikation
- Kenntnis unterschiedlicher Software-Bedienkonzepte (Erwartungshaltung basierend auf Erfahrungen mit vergleichbarer Software)
- Wissen über Barrierefreiheit
- Erfahrung mit Usability-Methodik
- Erfahrung mit Visualisierungssoftware
- Texte/Beschriftungen für User-Interaktionen schreiben

Programmieren

Geschenkt. Softwareentwicklung kommt nur in ganz seltenen Fällen ohne Programmieren aus. In diesem Blogpost soll es ja darum gehen, welche Gewerke außerdem gebraucht werden.

Design

Ja, jetzt meine ich tatsächlich die visuelle Gestaltung. Nebenbei bemerkt, das Deutsche versteht unter Design sehr eng das Sichtbare, wohingegen das Englische hier einen sehr weiten Begriff von Gestaltung hat, der auch Planung und Konstruktion umfasst.

Nicht jedes Softwareprojekt braucht eine grafische Oberfläche, es gibt auch durchaus viele Programme, die „in der Konsole“ bzw. einem Terminal-Fenster ausgeführt werden, also prinzipiell textbasiert. Doch auch hier finden zunehmend Visualisierungen statt, beispielsweise durch farbige Hervorhebungen von Warnhinweisen oder Fortschrittsbalken. Auch hier können Designer:innen wertvolle Beiträge leisten.

Leider wird dieses Gewerk oft als „Hübsch-machen“ missverstanden. Design sorgt aber für Verständlichkeit und Orientierung, für erlernbare Bedienprinzipien und Schutz vor Überfrachtung. Farbgebung und Illustrationen geben einem Projekt Wiedererkennbarkeit und früher oder später stellt man fest, dass man für nahezu jedes Tool ein Logo (nebst Logovarianten) haben möchte. Und sei es nur, damit das Projekt auf GitHub nicht so nackig aussieht.

Hilfreich neben der Design-Erfahrung:

- Erfahrung mit responsivem Design (es gibt keine „normale“ Bildschirmbreite)
- Wissen über Barrierefreiheit
- Teamfähigkeit und Kommunikation

Texting

Wir haben sicherlich alle schon mal über unverständlichen Fehlermeldungen gegrübelt. Man könnte daraus schließen, dass die Multitalent-Entwickler*innen dort kein voll besetztes Team hatten. Die Fähigkeit, zielgruppengerechte Texte zu schreiben, ist wohl die am meisten unterschätzte im Softwareentwicklungsbereich.

Texter*innen kümmern sich um knappe und zugleich verständliche Textinformationen unter Verwendung zielgruppengerechter Formulierungen. Beispielsweise die Beschriftung und Platzhaltertexte von Formularelementen, unmissverständliche Handlungsaufforderungen oder Rückfragen („Alle Elemente nicht löschen? – Ja|Nein“) oder unterstützende Hilfetexte ohne befremdlichen Fachjargon. Typischerweise gehört auch das Übersetzen englischer Textelemente des Quellcodes in die Oberflächensprache dazu. Hilfreich sind hier also:

- Englischkenntnisse
- Kenntnis der Sprachvoraussetzungen der Zielgruppe

- Fähigkeit, sich auf unterschiedliche Sprachstile einzustellen (vorzugsweise einschließlich einfacher Sprache)
- Bereitschaft, Texte in Programmcode zu suchen

Projektmanagement

Dieses Gewerk ist gerade im Bereich der Freien Software stark unterschätzt. Natürlich gibt es die erfolgreich sich selbst organisierenden Teams. Aber unzählige Vorhaben drohen unterzugehen, wenn sich niemand für die Fertigstellung zuständig fühlt.

Sei es die Definition von „fertig“ oder zumindest „gut genug für einen ersten Wurf“, bei der eine zweite Perspektive in einem 1-Personen-Team schon ungemein hilfreich ist. Oder sei es die Koordinierung der verschiedenen Gewerke, das Klären von Verfügbarkeiten, wenn man auf jemandes Zuarbeit wartet. Oder das Organisieren regelmäßiger Status-Update-Termine. Projektmanagement ist schon allein deswegen hilfreich, weil es mindestens einem weiteren Menschen nicht egal ist, ob oder wann ein Projekt fertiggestellt wird.

Durchgängiges Projektmanagement sorgt außerdem dafür, dass Informationen an einer Stelle zusammenlaufen und dokumentiert werden. Das verhindert, dass Teilprojekte von neuen Beitragswilligen immer wieder neu formuliert und angegangen werden. (Ihr glaubt gar nicht, wie oft das im ehrenamtlichen Bereich passiert.) Was du dafür brauchst:

- Empathie und Kommunikation
- Organisiertheit
- Freude an Dokumentation
- Motivationstalent
- Geduld

Testen/Qualitätssicherung/Fehler melden

Softwareentwicklung profitiert von fortlaufenden Tests am meisten, wenn der jeweilige Arbeitsstand von Menschen getestet wird, die den Code nicht selbst geschrieben haben. Gemeint sind hier nicht Usability-Tests mit vollkommen Fremden (darum kümmert sich eher das Gewerk UX-Design). Das geht auch ganz niedrigschwellig: „Kannst du dir das mal rasch anschauen?“

Ob Code-Qualität oder grafische Bedienelemente, Fehlermeldung oder die Prüfung auf Barrierefreiheit: es findet sich bestimmt jemand, den man fragen kann. Für diesen Blogpost belasse ich es dabei, dass sich jede:r ungeachtet der eigenen Talente eingeladen fühlen darf, Software zu testen und Rückmeldung zu geben. Allenfalls erforderlich:

- Respektvolle Kommunikation

Öffentlichkeitsarbeit

Sinnstiftend ist die entwickelte Software erst, wenn sie auch genutzt wird. Und dazu muss erst einmal bekannt gemacht werden, dass es sie gibt. Marketing, PR (public relations) oder eben Öffentlichkeitsarbeit sind die Gewerke, die sich darum kümmern.

Dafür sollte erst einmal eine Strategie erarbeitet werden: Wer nochmal sind unsere Zielgruppen? Wo erreichen wir sie und wofür brauchen sie die Software? Wozu wollen wir sie auffordern und wie messen wir den Erfolg der Maßnahmen?

Je nach Zielgruppe wird dort darüber gesprochen, wo sich Interessierte finden können. Das können Vorträge auf Open-Source-Konferenzen sein (idealerweise mit Aufzeichnung), Interviews bei einschlägigen Podcasts, Pressemitteilungen, Informationen auf Social-Media-Plattformen ... oder auch bezahlte Werbeplätze. Aber für letztere ist bei Open-Source-Software nur in den seltensten Fällen Budget vorhanden. Besonders hilfreich ist hier:

- Zielgruppen-/Branchenkenntnis
- Kommunikations- und Begeisterungstalent
- Erfahrung mit Open-Source-Communities („Marktschreierei“ und haltlose Produktversprechen vermeiden)
- Organisationstalent

Softwareentwicklung ist Teamarbeit

Auch wenn viele von uns mehrere dieser Gewerke beherrschen, halte ich es für wichtig, dass wir uns zumindest klarmachen, dass das so ist. Damit wir uns fragen, in welchem Job wir gut sind oder welchen wir gern machen und wobei wir uns besser Unterstützung holen sollten.

Es ist okay, wenn du als Programmiererin keine gute Texterin bist. Solange du es dir bewusst machst und die Aufgaben zusammenschreibst, damit jemand anderes den Job machen kann. Es ist okay, wenn du als Projektmanager auch für Konzeption und Qualitätssicherung verantwortlich bist. Du darfst aber die gute Idee kundtun, einen Sparringspartner zu benötigen.

Und du, die du vorhin noch den Impuls hattest, „Ich kann ja gar nicht programmieren“, du bist jetzt hoffentlich motiviert, deine Fähigkeiten einzubringen. Programmierer*innen haben wir in der Open-Source-Landschaft schon genug. Wir brauchen aber dringend deine Fähigkeiten!

„Ich bin dabei – wo soll ich anfangen?“

Du kennst bestimmt ein Open-Source-Projekt, das dir persönlich wichtig ist oder bei dem dich schon immer eine Kleinigkeit gestört hat. Suche nach einer offiziellen Website oder einem entsprechenden Projekt auf [GitHub](#). Suche zunächst nach Contribution guidelines. Üblicherweise musst du dich zuerst registrieren. Wenn du genau hinschaust, wirst du nirgendwo lesen, dass du für deine Beiträge bestimmte Qualifikationen mitbringen musst ;-)

Mach dir bewusst, welche Fähigkeiten du anbieten und einbringen kannst und stelle dich damit vor. Je nach Community könnte die Kommunikation z. B. über eine Mailingliste oder ein Chatsystem stattfinden, das findest du schon. Lege etwaige Selbstzweifel ab. Allein deine Bereitschaft, etwas beizutragen, ist schon ein Grund zur Freude. Du kannst dich natürlich auch schon auf die Suche begeben, wo du deinen Verbesserungsvorschlag vorstellen kannst oder wo eine Aufgabe beschrieben ist, die du dir zutraust. Und sei nicht frustriert, wenn eine Reaktion nicht sofort erfolgt. So arbeiten wir halt: dezentral und zeitversetzt.

Und wenn du selbst dich in keinem der Gewerke siehst, kannst du zumindest schon etwas beitragen, indem du diesen Artikel mit anderen teilst, das wäre ja auch schon Öffentlichkeitsarbeit.

Wirklich alle können etwas zur Open-Source-Entwicklung beitragen!

Revision #1

Created 13 October 2026 10:22:55 by patricia

Updated 13 October 2026 10:27:49 by patricia