

computor – your computing tutor

Endbericht | Call 20 | Projekt ID 8012

Lizenz CC BY 4.0

Inhalt

1 Einleitung....3

2 Projektbeschreibung....3

3 Verlauf der Arbeitspakete....5

3.1 AP1: Detailplanung und Formales am Projektstart....5

3.2 AP2: Vorbereitung, Architektur und Wissenstransfer....5

3.3 AP3: v26.3 Entwicklung: KI-Assistenz....6

3.4 AP4: v26.3 Testing und Stabilisierung....6

3.5 AP5: v26.3 Produktionsbetrieb....6

3.6 AP6: v26.10 Entwicklung: Web-Modus....7

3.7 AP7: Kurserstellung und didaktische Inhalte....7

3.8 AP8: Technische Dokumentation....8

3.9 AP9: v26.10 Stabilisierung und Öffentlicher Release....8

3.10 AP10: Dokumentation und Formales am Projektende....9

4 Umsetzung Förderauflagen....9

5 Liste Projektendergebnisse....9

6 Verwertung der Projektergebnisse in der Praxis....10

7 Öffentlichkeitsarbeit/ Vernetzung....11

8 Eigene Projektwebsite....12

9 Geplante Aktivitäten nach netidee-Projektende....12

10 Anregungen für Weiterentwicklungen durch Dritte....12

1 Einleitung

Dieser Bericht beschreibt das netidee-Projekt computor (Call 20, Projekt 8012), das von November 2025 bis Oktober 2026 an der Technischen Universität Graz (TU Graz) unter der Leitung von Christopher Albert durchgeführt wurde.

computor ist eine Open-Source-Plattform für den Unterricht in Programmieren und Data Science. Lernende lösen dabei Aufgaben in Visual Studio Code (VS Code), einem kostenlosen und weit verbreiteten Code-Editor. Wer nicht weiterkommt, kann die KI-Tutorin Lena um Hilfe fragen: Sie erklärt, gibt Hinweise, liefert aber keine Lösung. Die Webanwendung computor.at öffnet dieselbe Arbeitsumgebung in einer VS-Code-Instanz im Browser, enthält drei öffentliche Python-Kurse und bietet Lehrenden ein Dashboard mit dem Lernfortschritt ihrer Kurse. Alle Teile stehen unter offenen Lizenzen und lassen sich auf eigenen Servern betreiben.

Zum Projektende am 31. Oktober 2026 sind alle elf geplanten Projektergebnisse fertiggestellt. computor ist seit März 2026 in der Lehre der TU Graz im Einsatz und ist Ende September 2026 öffentlich gestartet.

2 Projektbeschreibung

Ziel und Zielgruppen. Wer programmieren lernt, braucht schnelle und konkrete Rückmeldungen. Während in großen Lehrveranstaltungen Tutor:innen oft nicht die Zeit haben, allen Studierenden ausführliche und zeitige Unterstützung zu liefern, antworten KI-Chatbots sofort – meist aber mit der fertigen Lösung, während die Eingaben der Lernenden oft auf fremden Servern verarbeitet werden. computor soll Lernenden Hilfestellungen geben, die das eigene Nachdenken anregen und dabei die Daten an der eigenen Einrichtung halten. Zielgruppen sind Lehrende und Studierende in Naturwissenschaft, Technik und Data Science sowie Schulen, Unternehmen mit eigenen Schulungen, Kursanbieter:innen und Selbstlernende.

Die VS-Code-Extension. computor existiert zum einen als VS-Code-Extension, also als Erweiterung, die dem Standardeditor neue Funktionen gibt. Um sie leicht zugänglich zu machen, kann sie mit einem Klick aus dem Visual Studio Marketplace installiert werden, dem offiziellen Katalog für VS-Code-Erweiterungen.

Lena, die KI-Tutorin. Lena, unsere KI-Tutorin, ist in die computor-Erweiterung für VS Code eingebaut. Lernende schreiben Lena über dieselbe Chatfunktion, über die sie auch menschliche Tutor:innen erreichen können. Lena kennt die Angabe, den Code der Lernenden und die letzten Testergebnisse. Sie antwortet mit Erklärungen,

Rückfragen und Hinweisen, um Studierende bei der Fehlersuche zu unterstützen, wobei sie weder vollständige Lösungen noch korrigierten Code ausgibt. Dafür sorgt eine zweistufige Prüfung, die Versuche erkennt, ihr trotzdem eine Lösung zu entlocken. Ihre Verhaltensregeln beruhen dabei auf der Lehrerfahrung von Winfried Kernbichler.

Lena verwendet Qwen, ein Open-Weight-Sprachmodell. Das bedeutet, dass das trainierte Modell veröffentlicht wurde und auf eigenen Rechnern laufen kann. Die TU Graz betreibt Qwen auf ihrer eigenen Hardware, das heißt, dass Fragen und Code der Studierenden die Universität nicht verlassen und Lernende kein Konto bei einem kommerziellen KI-Anbieter benötigen.

Arbeiten im Browser. Unter computor.at können angemeldete Lernende eine vollständige VS-Code-Umgebung direkt im Webbrowser öffnen. Dafür müssen sie nichts installieren und haben dort dieselben Funktionen wie in der VS-Code-Extension am eigenen Rechner, einschließlich Lena. Man sieht die Aufgabenstellungen, kann Code schreiben und testen sowie die Code-Beispiele direkt abgeben. Die automatischen Tests überprüfen, ob das geschriebene Programm die Vorgaben erfüllt und die erwarteten Zahlen und Grafiken erzeugt.

Öffentliche Kurse. Unter *computor.at/learn* stehen drei Kurse unter dem Namen „Data Science mit Python“ mit 70 Übungen auf Deutsch und Englisch zur Verfügung, die auch ohne Konto angesehen werden können. Mit der Registrierung erhalten Lernende die Möglichkeit, die Übungen zu bearbeiten, ihren Fortschritt zu speichern und Lena Fragen zu stellen.

Dashboard für Lehrende. Lehrende sehen in *computor.at* ein Dashboard mit Kursstatistiken und Fortschritt jeder bzw. jedes Studierenden, welches für jede Aufgabe die Anzahl der Testversuche, ihre Ergebnisse und die häufigsten Fehler darstellt. In der Kursübersicht geben Markierungen Auskunft über Unstimmigkeiten im Arbeitsfortschritt von Studierenden und weisen Lehrende zum Beispiel auf mögliche Rückstände hin. Tutor:innen haben eine eigene Ansicht, in der sie die Abgaben sehen und bewerten können.

Self-Hosting und Open Source. Self-Hosting heißt, dass eine Einrichtung die Software auf eigenen Servern betreibt, statt einen fremden Dienst zu nutzen. Unser Server (computor-backend) verwaltet Benutzer:innen, Kurse, Abgaben und Tests und stellt die Webanwendung bereit. Die Erweiterung liegt im Repository *computor-vscode*, der Lena-Dienst in *computor-agent*, das Installationsprogramm in *computor-installer* und die Kurse in *data-science-python*. Die Serverteile werden als Docker-Container

ausgeliefert, einem verbreiteten Format für Serversoftware, mit dem sich eigene Instanzen in wenigen Befehlen aufsetzen lassen. Die Software steht unter der MIT-Lizenz; Dokumentation und Kurstexte unter CC BY 4.0.

Team. Christopher Albert (Projektleitung), Winfried Kernbichler (Senior Expert, Didaktik), Georg Graßler (Architektur und Einbindung in die Lehre), Maximilian Mandlez (Lead Developer) sowie die studentischen Mitarbeiter:innen Stefanie Strobl und ab September 2026 Jakob Thaler (Kurse und Dokumentation). Das Team arbeitete insgesamt 1.814 Stunden am Projekt.

3 Verlauf der Arbeitspakete

3.1 AP1: Detailplanung und Formales am Projektstart

Tätigkeiten und Ergebnisse. Im November und Dezember 2025 unterzeichneten wir die Fördervereinbarung, erstellten den Detailprojektplan, die Liste der Projektergebnisse mit Lizenzen und Webadressen und richteten die netidee-Projektseite ein. Der erste Blogbeitrag erschien am 11. Dezember 2025. Die erste Förderrate wurde beantragt und genehmigt.

Abweichungen zum Plan. Der Aufwand lag mit 10 Stunden um 2 Stunden unter dem Plan. Nach dem Zwischenbericht haben wir 22 Stunden von Georg Graßler aus Dezember 2025 herausgenommen, weil sie bereits in einem anderen Projekt erfasst waren. Sie betrafen AP1 (2 Stunden), AP2 (8 Stunden), AP3 (10 Stunden) und AP8 (2 Stunden).

3.2 AP2: Vorbereitung, Architektur und Wissenstransfer

Tätigkeiten und Ergebnisse. Winfried Kernbichler unterrichtet seit vielen Jahren Programmieren für Physikstudierende an der TU Graz. In Interviews hielten wir fest, welche Bewertungsschemata er verwendet und welche Hinweise Studierenden weiterhelfen, ohne ihnen die Lösung für die Aufgabe zu geben. Daraus entstanden die Regeln für Lenas Verhalten, während eine Lehrveranstaltungsbefragung mit der Vorgängerversion von *computor* den Bedarf der Studierenden erhob. Die Architektur für den Browser-Modus und die KI-Anbindung wurden auch dokumentiert. Georg Graßler bereitete sich auf die interimistische Projektleitung von Jänner bis März 2026 vor.

Erkenntnisse. Weil die didaktischen Regeln vor der Entwicklung schriftlich vorlagen, konnten wir Lena gegen klare Erwartungen testen.

Abweichungen zum Plan. Der Aufwand lag mit 44 Stunden um 8 Stunden unter dem Plan (Korrektur Dezember 2025, siehe AP1).

3.3 AP3: v26.3 Entwicklung: KI-Assistenz

Tätigkeiten und Ergebnisse. Wir bauten die erste Version von Lena in die Erweiterung ein. Mit ihr erhalten Studierende im Chat Hilfe, ohne dass Musterlösungen preisgegeben werden. Nach einer Abgabe können sie außerdem Feedback zu Codequalität und Syntax anfordern. Die Chat-Darstellung in der Erweiterung wurde überarbeitet, und Übungsaufgaben aus der Vorgängerversion von computor lassen sich übernehmen. Lena spricht das Sprachmodell über eine Programmierschnittstelle an, die OpenAI eingeführt hat und die heute viele Modelldienste und Open-Source-Modellserver anbieten. Wer *computor* selbst betreibt, kann daher jeden Modelldienst mit dieser Schnittstelle verwenden, auch ein Modell auf eigener Hardware.

Besondere Erfolge. Das Sprachmodell lief von Beginn an zuverlässig auf Hardware der TU Graz.

Abweichungen zum Plan. Der Aufwand lag mit 446 Stunden um 10 Stunden unter dem Plan (Korrektur Dezember 2025, siehe AP1).

3.4 AP4: v26.3 Testing und Stabilisierung

Tätigkeiten und Ergebnisse. Im Jänner und Februar 2026 testeten wir v26.3 in internen Sitzungen, wo die Teammitglieder die Rollen von Studierenden, Tutor:innen und Lehrenden übernahmen. Alle kritischen Fehler aus diesen Sitzungen wurden behoben und die Kursinhalte aus der Vorgängerversion v25.10 übertragen, sodass zu Beginn des Sommersemesters v26.3 bereit für den Lehrbetrieb war.

Erkenntnisse. Die Rollenspiele fanden mehr praktische Probleme als einzelne technische Tests, weil sie dem Ablauf einer echten Kurswoche folgten.

Abweichungen zum Plan. Keine.

3.5 AP5: v26.3 Produktionsbetrieb

Tätigkeiten und Ergebnisse. Seit März 2026 läuft *computor* in der Lehrveranstaltung *„Programmieren in der Physik“* der TU Graz. 130 Studierende haben dort ihre Übungen bearbeitet, getestet, abgegeben und Lena um Hilfe gefragt. Auch die Tutor:innen haben computor genutzt, um die Abgaben der Studierenden zu bewerten und ihnen Feedback zu geben. Während des Semesters kamen außerdem ein Wartungsmodus und Chatverläufe zwischen Studierenden und Tutor:innen hinzu. Fehler in Übersetzungen und in der Formeldarstellung der Aufgaben wurden korrigiert.

Erkenntnisse. Ein ganzes Semester Betrieb zeigte, wo Überwachung und automatische Wiederherstellung des Servers nachgebessert werden mussten. Diese Änderungen gingen in v26.10 ein.

Abweichungen zum Plan. Keine. Das Ziel von mindestens zwei Kursen und mindestens 100 Nutzer:innen ist erreicht.

3.6 AP6: v26.10 Entwicklung: Web-Modus

Tätigkeiten und Ergebnisse. Von März bis Juli 2026 entwickelten wir v26.10 mit der Webanwendung *computor.at*. Lernende können dort VS Code direkt im Browser öffnen und ohne Installation arbeiten. Lena wurde mit den Erfahrungen aus dem Sommersemester erweitert und gibt jetzt Rückmeldungen zu Code und Ergebnissen, die sich auf die konkrete Aufgabe beziehen. Für Lehrende entstand das Dashboard mit Kursstatistiken, Fortschritt jeder bzw. jedes Studierenden, Testversuchsverläufen, Feedback-Historie, häufigen Fehlern bei Aufgaben sowie nachvollziehbaren Markierungen. Tutor:innen bewerten Abgaben in einer eigenen Ansicht. Auch die Docker-basierte Installation für den eigenen Betrieb wurde fertiggestellt und dokumentiert.

Grafiken. Grafiken in Abgaben prüft jetzt ein regelbasierter Plot-Grader, der mithilfe von KI-Werkzeugen an echten Abgaben verbessert und als Open Source veröffentlicht wurde.

Abweichungen zum Plan. Geplant war eine Bewertung von Grafiken durch KI, der verbesserte regelbasierte Plot-Grader braucht jedoch keine Bild-KI. Dadurch erhalten Abgaben einheitliche Ergebnisse und Bilder studentischer Arbeiten gehen an kein Sprachmodell.

3.7 AP7: Kurserstellung und didaktische Inhalte

Tätigkeiten und Ergebnisse. Gemeinsam mit Winfried Kernbichler machten wir aus rund 70 Übungen der Lehrveranstaltung „*Programmieren in der Physik*“ die drei öffentlichen Kurse „*Data Science mit Python*“. Die Grundlagen umfassen 21 Übungen zu Programmierumgebung, Konsole, Python-Syntax, Funktionen und Indizierung. Der Aufbau-Kurs enthält 31 Übungen zu Kontrollstrukturen, logischen Operatoren und der Auswertung von Messdaten aus Laborübungen. In der Vertiefung geht es in 18 Übungen um Gleichungssysteme, Kurvendiskussionen, Differentialgleichungen und objektorientierte Programmierung. Die Kurse gibt es sowohl auf Deutsch als auch auf Englisch. Der Schwerpunkt liegt vor allem auf dem Einlesen, Auswerten und Darstellen von Messdaten, um Lernenden insbesondere Datenverarbeitung und -auswertung

näher zu bringen. Musterlösungen werden nicht veröffentlicht, damit Lernende selbst arbeiten müssen.

In den Kursen kommen Lena, die automatischen Tests und die Fortschrittsanzeige zum Einsatz. Vor und nach dem öffentlichen Start haben wir die Kurse auf computor.at selbst durchgearbeitet, teils mit automatisierten Browser-Tests, und die dabei gefundenen Fehler der Plattform behoben. Seitdem öffnen Browser-Arbeitsplätze den gewählten Kurs direkt, Formeln werden zuverlässiger dargestellt und Lenas Antworten sind besser formatiert.

Abweichungen zum Plan. Der Aufwand lag mit 145 Stunden um 10 Stunden unter dem Plan, weil die studentische Mitarbeiterin Stefanie Strobl im Juli und August 2026 nicht angestellt war.

3.8 AP8: Technische Dokumentation

Tätigkeiten und Ergebnisse. Die Entwickler:innen-Dokumentation beschreibt Systemkonzept, Softwarestruktur und die Schnittstellen (API) des Servers. Die Anwender:innen-Dokumentation erklärt *computor* für Lernende sowie Lehrende und enthält außerdem einen Leitfaden zum eigenständigen Anlegen von Kursen und Aufgaben. Zusätzlich gibt es eine Anleitung, die die Installation mit Docker für Hochschulen und Unternehmen beschreibt, und einen Contribution Guide, der erklärt, wie externe Entwickler:innen mitarbeiten können. Die gesamte Dokumentation liegt in den GitHub-Repositories der Organisation computor-org unter CC BY 4.0.

Erkenntnisse. Dokumentation und Code liegen im selben Repository und wurden gemeinsam geändert, wodurch beide zum Projektende übereinstimmen.

Abweichungen zum Plan. Der Aufwand lag mit 217 Stunden um 16 Stunden unter dem Plan, weil Stefanie Strobl im Juli und August 2026 nicht angestellt war. Ab September 2026 übernahm der studentische Mitarbeiter Jakob Thaler einen Teil ihrer Aufgaben und arbeitete an der Dokumentation mit.

3.9 AP9: v26.10 Stabilisierung und Öffentlicher Release

Tätigkeiten und Ergebnisse. Ende Juli 2026 legten wir den Funktionsumfang von v26.10 fest. Im August und September testeten wir die Version und behoben die verbleibenden Fehler. Dafür trafen wir uns am 27. August 2026 zu einem eintägigen Release-Workshop, wo der Lead Developer sein Wissen über Betrieb und Entwicklung an das übrige Team übergab, damit *computor* auch ohne ihn gewartet werden kann. Der Server erschien als v2026.10 auf GitHub, die Computor-Erweiterung im Visual Studio Marketplace in der Versionsreihe 2027.3. Die drei öffentlichen Kurse, die im

Blogbeitrag vom 14. Oktober vorgestellt wurden, gingen Ende September 2026 auf *computor.at/learn* online.

Abweichungen zum Plan. Die Anstellung des Lead Developers wurde bis Ende September 2026 verlängert, um den Release abzuschließen. netidee genehmigte dafür im April 2026 zusätzliche 200 Stunden. Die geplanten Stunden der studentischen Mitarbeiterin teilten sich im September Stefanie Strobl und Jakob Thaler. Die Mehrkosten trug die TU Graz.

3.10 AP10: Dokumentation und Formales am Projektende

Tätigkeiten und Ergebnisse. Im Oktober 2026 entstanden dieser Endbericht und die einseitige Zusammenfassung. Wir erstellten die Endabrechnung und den Abruf der letzten Förderrate, ergänzten die netidee-Projektseite um alle Ergebnisse mit ihren Lizenzen und veröffentlichten den Abschlussbeitrag im Blog.

Abweichungen zum Plan. Keine.

4 Umsetzung Förderauflagen

Die Fördervereinbarung enthält keine besonderen Auflagen. Die allgemeinen Bedingungen zu offenen Lizenzen sind erfüllt. Die gesamte Software steht unter der MIT-Lizenz. Berichte und Dokumentation stehen unter CC BY 4.0, ebenso die Kurstexte, deren Code unter der MIT-Lizenz steht.

5 Liste Projektendergebnisse

Alle Projektergebnisse sind fertiggestellt und kostenlos öffentlich zugänglich.

Nr.	Projektergebnis	Lizenz	Webadresse
1	**Projektzwischenbericht.** Bericht über die erste Projekthälfte (April 2026).	CC BY 4.0	https://www.netidee.at/computor
2	**Projektendbericht.** Dieser Bericht.	CC BY 4.0	https://www.netidee.at/computor
3	**Entwickler:innen-Dokumentation.** Systemkonzept, Softwarestruktur, Server-API, Self-Hosting-Anleitung und Contribution Guide.	CC BY 4.0	https://github.com/computor-org (computor-backend, computor-vscode, computor-agent, computor-installer, plot-grader)
4	**Anwender:innen-Dokumentation.** Anleitungen für Lernende und Lehrende, einschließlich Leitfaden für Lehrende zum Anlegen von Kursen und Aufgaben.	CC BY 4.0	https://github.com/computor-org (computor-vscode, computor-backend, data-science-python)
5	**Zusammenfassung.** Einseitige Zusammenfassung von Projekt und Ergebnissen.	CC BY 4.0	https://www.netidee.at/computor
6	**Dokumentation Externkommunikation.** Maßnahmen, Aufwand und Nutzen, Lessons	CC BY 4.0	https://www.netidee.at/computor

	Learned in Kapitel 7 dieses Berichts.		
7	**VS-Code-Erweiterung „Computor“ mit KI-Tutorin Lena.** Kurse, Aufgaben, Tests und Abgaben direkt im Editor. Lena gibt Hinweise, ohne Lösungen zu verraten (ID computor-org.computor, Versionsreihe 2027.3).	MIT	https://marketplace.visualstudio.com/items?itemName=computor-org.computor, https://github.com/computor-org/computor-vscode
8	**Automatisches Review und Feedback.** Lena gibt Rückmeldung zu Code und Ergebnissen mit einem an der TU Graz betriebenen Open-Weight-Modell (Qwen). Grafiken prüft ein regelbasierter Plot-Grader, der mithilfe von KI-Werkzeugen mit echten Abgaben verbessert wurde.	MIT	https://github.com/computor-org/computor-agent, https://github.com/computor-org/plot-grader
9	**Vollständiges Self-Hosting-Paket.** Server mit Docker-Deployment (Releases v2026.10), Installationsprogramm, Lena-Dienst und Erweiterung.	MIT	https://github.com/computor-org (computor-backend, computor-installer, computor-agent, computor-vscode)
10	**Dashboard für Lehrende.** Kursstatistiken, Fortschritt je Studierender/m, Testversuche und Zeitverlauf, Feedback-Historie, typische Fehler, regelbasierte Unterstützungshinweise, Bewertungsansichten für Tutor:innen. Teil der Webanwendung.	MIT	https://computor.at, https://github.com/computor-org/computor-backend
11	**Drei öffentliche Beispielkurse „Data Science mit Python“.** Grundlagen, Aufbau und Vertiefung: 70 Übungen auf Deutsch und Englisch. Kostenlos zugänglich.	CC BY 4.0 (Kurstexte), MIT	https://computor.at/learn, https://github.com/computor-org/data-science-python

6 Verwertung der Projektergebnisse in der Praxis

An der TU Graz ist *computor* die Plattform der Lehrveranstaltung „*Programmieren in der Physik*“. 2026 gaben dort 130 Studierende ihre Übungen ab und fragten Lena um Hilfe. Tutor:innen können im Dashboard sehen, wer bei welcher Aufgabe hängt, und gezielt helfen.

Die drei Kurse auf *computor.at/learn* sind für alle frei zugänglich. Lernende haben dabei die Möglichkeit, im Browser oder in ihrer eigenen VS-Code-Installation zu arbeiten. Lehrende an anderen Schulen und Hochschulen dürfen die Kurstexte unter CC BY 4.0 übernehmen und anpassen.

Einrichtungen, die die Daten ihrer Lernenden auf eigenen Servern halten wollen, installieren das Paket mit dem Installationsprogramm und der Docker-Konfiguration. Sie haben dabei die Möglichkeit, eigene Kurse einzubinden und können ihr eigenes Sprachmodell wählen.

7 Öffentlichkeitsarbeit/ Vernetzung

Dieses Kapitel ist zugleich die Dokumentation der externen Kommunikation (Projektergebnis 6). Es beschreibt die getätigten Maßnahmen, bewertet deren Aufwand und Nutzen und fasst die „Lessons Learned“ zusammen.

netidee-Projektseite und Blog. Die Projektseite *netidee.at/computor* wurde zu Projektbeginn eingerichtet. Dort erschienen seitdem sechs Blogbeiträge, die den Kick-off (11. Dezember 2025), den Zwischenbericht (8. April 2026), die Funktionsweise von Lena (15. April 2026), die Gliederung des Python-Kurses in drei Stufen (22. Juli 2026), die drei öffentlichen Kurse nach dem Start (14. Oktober 2026) und den Projektabschluss (Oktober 2026) behandeln. Der Erstellungsaufwand war mittel, da vor allem Screenshots verwendet wurden und der Softwarestand verglichen wurde. Der Nutzen war jedoch hoch, weil die Beiträge die wichtigste öffentliche Dokumentation des Projekts sind.

netidee Summer Talk. Am 25. Juni 2026 stellte der Projektleiter *computor* beim netidee Summer Talk Call 20 – einem Online-Treffen der geförderten Projekte – in zwei Minuten vor. Der Aufwand war gering, der Nutzen mittel. Das kurze Format zwang zu einer klaren Botschaft, und der Austausch mit anderen Projekten brachte Anregungen.

Offene Repositories und Rückmeldungen. Quellcode, Dokumentation und Kursmaterial sind auf GitHub öffentlich (*github.com/computor-org*) zugänglich. Rückmeldungen erreichen das Team über die Schaltfläche „Problem melden“ in *computor.at* und VS Code sowie per E-Mail an *contact@computor.at*. Der Aufwand kann als gering bis mittel eingestuft werden, jedoch ist der Nutzen hoch, da Meldungen direkt zu Verbesserungen führen.

Visual Studio Marketplace. Die Computor-Erweiterung ist im offiziellen Erweiterungskatalog von VS Code gelistet, wo Lehrende und Entwickler:innen sie über die Suche finden können. Nach Einrichtung des Release-Ablaufs war der Aufwand gering und der Nutzen für die Sichtbarkeit hoch.

Einsatz in der Lehre an der TU Graz. Der Einsatz in einer regulären Lehrveranstaltung mit 130 Studierenden war Teil des normalen Lehrbetriebs. Der Nutzen war hoch, da dieser Einsatz die meisten Rückmeldungen brachte und anderen Lehrenden ein reales Beispiel vorzeigt.

Lessons Learned und Empfehlungen. Beiträge mit Screenshots, Kursaufbau oder Zahlen aus dem echten Einsatz waren leichter zu schreiben als allgemeine

Fortschrittsberichte. Beiträge zu festen Meilensteinen wie Zwischenberichte, Kursveröffentlichung und Start ließen sich gut planen. Wir empfehlen anderen Projekten, ihre Beiträge vom ersten Monat an entlang solcher Meilensteine festzulegen. Offener Code und ein einfacher öffentlicher Weg für Fehlermeldungen brachten konkrete und brauchbare Rückmeldungen.

8 Eigene Projektwebsite

Neben der netidee-Projektseite (*https://www.netidee.at/computor*) betreibt das Projekt die Website und Webanwendung *https://computor.at*. Die öffentlichen Kurse liegen unter *https://computor.at/learn*, Quellcode und Dokumentation unter *https://github.com/computor-org* und die Erweiterung unter *https://marketplace.visualstudio.com/items?itemName=computor-org.computor.* Die Websites weisen auf die Förderung durch netidee hin.

9 Geplante Aktivitäten nach netidee-Projektende

Die TU Graz betreibt *computor.at* und das Sprachmodell hinter Lena nach Projektende weiter, wodurch die öffentlichen Kurse und Lena kostenlos nutzbar bleiben. *computor* wird die Plattform von „*Programmieren in der Physik*“ bleiben und soll in weiteren Lehrveranstaltungen der TU Graz eingesetzt werden. Wir planen, die Erweiterung mit neuen VS-Code-Versionen kompatibel zu halten und die Kurse anhand der Rückmeldungen von Lernenden und Lehrenden zu erweitern. Lenas Hinweise wollen wir mit den neuen Erfahrungen aus der Lehre verbessern.

10 Anregungen für Weiterentwicklungen durch Dritte

Schulen, Hochschulen und Unternehmen können eine eigene computor-Instanz auf ihren Servern betreiben und ihr eigenes Sprachmodell einbinden.

Lehrende können die Kurse übersetzen, auf andere Fächer übertragen oder ganz neue Kurse schreiben.

Entwickler:innen können Lenas Regeln an andere Lehrstile anpassen oder automatische Prüfungen für weitere Aufgabentypen ergänzen.

Forschungsgruppen können untersuchen, wie Hinweise einer KI-Tutorin das Lernen beeinflussen, da Fragen, Hinweise und Abgaben im selben System erfasst sind.

Beiträge über die öffentlichen Repositories sind willkommen.