



AI TRENDLETTER

Künstliche Intelligenz: Wirklichkeit und Visionen

Interview

„Den komplexen Herstellungsprozess von Software mit KI optimieren und neu gestalten“

Interview mit Prof. Dr. Jürgen Döllner, Leiter des KI-Labors am Hasso-Plattner-Institut

von Michael Ihringer

Das [Hasso-Plattner-Institut für Digital Engineering](#) (HPI) erforscht den Einsatz Künstlicher Intelligenz (KI) im Software-Engineering am neuen KI-Labor für Software-Technik. Das Ziel ist es, diejenigen KI-Ansätze für das Management komplexer Software-Projekte zu identifizieren, mit denen sich die größtmöglichen Effizienzpotenziale realisieren lassen.



Prof. Dr. Jürgen Döllner, der Leiter des KI-Labors am Hasso-Plattner-Institut in Potsdam

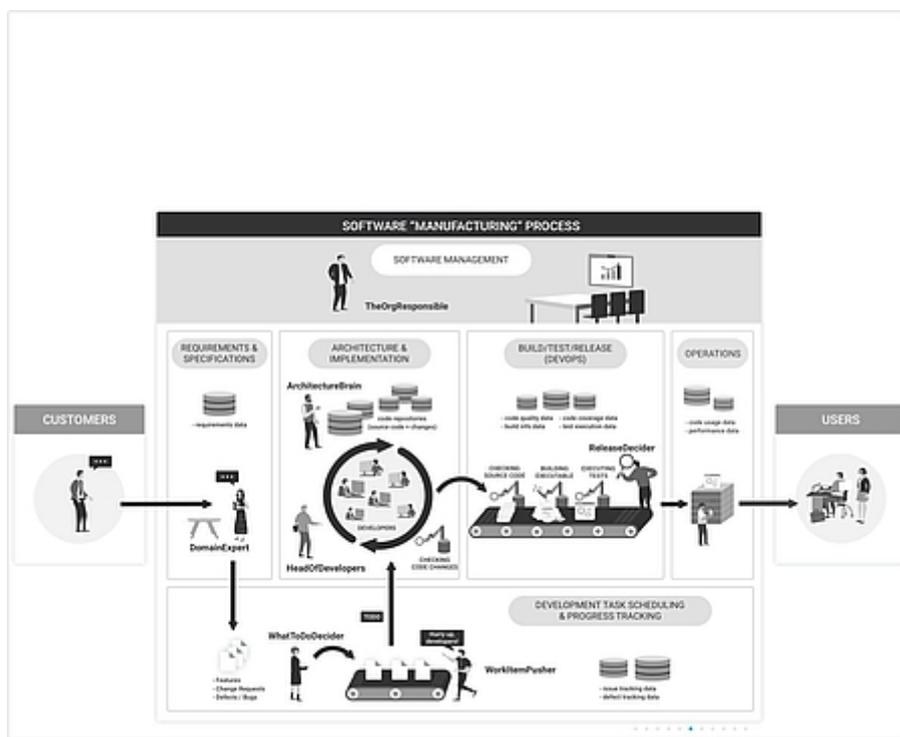
[Prof. Dr. Jürgen Döllner](#), der Leiter des KI-Labors, erläutert im Interview die Hintergründe. Außerdem beschreibt er die Ziele der geplanten HPI-Studie „[AI for Software Engineering](#)“, an der interessierte Unternehmen teilnehmen können, die das Potenzial von KI für ihr eigenes Software-Engineering kennenlernen wollen. Die eingereichten Software-Entwicklungsprojekte sollen seit mindestens zwei Jahren laufen und zehn oder mehr Entwickler involvieren. Die ausgewählten Unternehmen erhalten einen

schnellen Einstieg in das Zukunftsthema „KI im Software-Engineering“ mit Guidelines für den Aufbau einer eigenen KI-Strategie, einer Analyse und Bewertung der Software-Entwicklung anhand des eingereichten Projekts und ein Benchmarking innerhalb des Teilnehmerfeldes.

Interessierte Unternehmen können ihre Bewerbung bis zum 6. Dezember mit einer kurzen Projektbeschreibung per Online-Formular unter www.ai4se.de einreichen. Die Bekanntgabe der Studienteilnehmer erfolgt am 15. Dezember, Studienbeginn ist dann der 1. Januar 2020.

Herr Prof. Döllner, KI ist eines der Trendthemen unserer Zeit. Warum kommt KI ausgerechnet im Software-Engineering, wo man an der Quelle sitzt, bislang nicht zum Einsatz?

Prof. Dr. Jürgen Döllner: Erste praktische Einsatzgebiete von KI lagen in der optischen Zeichenerkennung – um zum Beispiel handschriftlich geschriebene Adressen zu erkennen. In vielen Anwendungen konnte KI ihre Wirkung jedoch nicht voll entfalten, da die Leistung der Hardware noch nicht ausreichte oder die Daten keine ausreichende Qualität aufwiesen. Letztlich ist KI auf hohe Rechenleistungen und Big Data angewiesen. Ohne sinnvolle Trainingsdaten zum Beispiel kann KI keine ausreichend genauen Entscheidungen treffen. Das ist ein wichtiger Grund, warum Maschinelles Lernen sich nicht sofort verbreitet hat.



Teildisziplinen, Aufgaben und Rollen im Prozess der Software-Herstellung

Im Software-Engineering selbst findet KI bislang kaum Anwendung. Hier hat es nie einen echten Paradigmenwechsel gegeben: Auch im „agilen Zeitalter“ gleicht die Software-Herstellung immer noch eher einem Kunsthandwerk als einem industriellen Prozess mit planbarem, kalkulierbarem Ergebnis.

Daten, die mit Big-Data-Methoden ausgewertet werden könnten, gibt es heute aber genug. Jedes einzelne Software-Werkzeug bringt ja sein eigenes Repository mit.

Es ist in der Tat eine eigenartige Situation: In Repositorien und all den anderen Werkzeugen haben wir ausgezeichnete, präzise Daten über alle wesentlichen Aspekte der Software-Entwicklung vorliegen. Wir wissen zu jedem Zeitpunkt, wer welchen Code an welchem Modul geändert hat. Wir wissen auch, wann und wo es einen Fehler gab und wer ihn bearbeitet hat.

Diese und unzählige weitere Daten werden seit Jahrzehnten gesammelt. Aber es fehlte bislang schlicht und ergreifend die Erkenntnis, dass man aus diesen Daten wertvolles Wissen generieren kann: Die Idee von „Software-Analytics“ ist relativ neu und wird bis heute in der Praxis kaum umgesetzt. Die Daten werden meist nur von den Experten genutzt, die die betreffenden Werkzeuge bedienen.

Wenn ich die Daten in ihrer Gesamtheit betrachte, kann ich zum Beispiel Rückschlüsse auf Evolution, Qualität und Risiken meines

Software-Projekts und meines Entwicklerteams ziehen. Darum geht es bei Software-Analytics: die Repositorien systematisch auszuwerten, um übergeordnetes Wissen zu generieren, das allen Verantwortlichen im Software-Herstellungsprozess hilft, vor allem dem Software-Management.



Visualisierung eines komplexen Systems als Software-Landkarte

Bei Big Data stellt sich immer die Frage, wer die Hoheit über die Daten hat und ob man die eigenen Daten hergeben möchte. Inwieweit spielen solche Befindlichkeiten im Software-Engineering eine Rolle?

Wir nutzen bei der Analyse von Software-Projekten ausschließlich Daten des jeweiligen Projekts. Hier ist die Eigentümerschaft von vornherein klar. Allerdings ist der traditionelle Herstellungsprozess von Software stark gegliedert und zeichnet sich datentechnisch durch eine Silo-Struktur aus: Es gibt keine durchgängige Datenbasis – vielmehr haben wir verschiedene Stufen, die jeweils eine eigene Datenhaltung haben. Deshalb ist es der erste Schritt in Richtung Software-Analytics, alle Datensilos zu koppeln und deren Inhalt in einen gemeinsamen Informationsraum über die Software-Entwicklung zu überführen.

Wer es dabei bewenden lässt, schöpft allerdings nicht das volle Potenzial von Software-Analytics aus. Dafür ist es notwendig, Daten über viele Projekte zu sammeln und losgelöst von den Einzelprojekten, gewissermaßen anonymisiert, Wissen aufzubauen. Unser assoziierter Partner [Seerene](#) verfügt über eine Wissensbasis aus dem fortlaufenden, systematischen Scannen von mehreren 10.000 Open-Source-Projekten. Die gewonnenen Daten beschreiben die Software auf abstrakter Ebene: Für jedes Modul wird ein sogenannter hochdimensionaler Feature-Vektor generiert, der für jeden Zeitpunkt wesentlich Eigenschaften und Zustände des Moduls wiedergibt. So lassen sich unabhängig vom Quellcode Aussagen über die Evolution und die Software-Qualität machen.

Wie profitiert das einzelne Unternehmen von diesem projektübergreifend aggregierten Wissen?

Eine Wissensbasis, die man aus so vielen Projekten generiert, kann helfen, die eigene Entwicklung einzuordnen und zu verbessern. Dann lassen sich Fragen beantworten wie: Ist mein Herangehen an die Software-Entwicklung für meine Aufgabenstellung statistisch gesehen erfolgversprechend? Bedeutet es mittel- und langfristig mehr oder weniger Risiken?

Betrachte ich zurückliegende Projekte, deren Verlauf ich kenne, kann ich Rückschlüsse auf das laufende Projekt ziehen. So entsteht ein Frühwarnsystem, das auf Gefahren und Schwierigkeiten hinweist oder Konsequenzen ermittelt, die sich mir in der Zukunft stellen werden. Dann kann ich einschätzen, ob meine Software gewünschte Eigenschaften, etwa bezüglich Stabilität, Portabilität und Fehlerfreiheit, besitzt oder nicht.

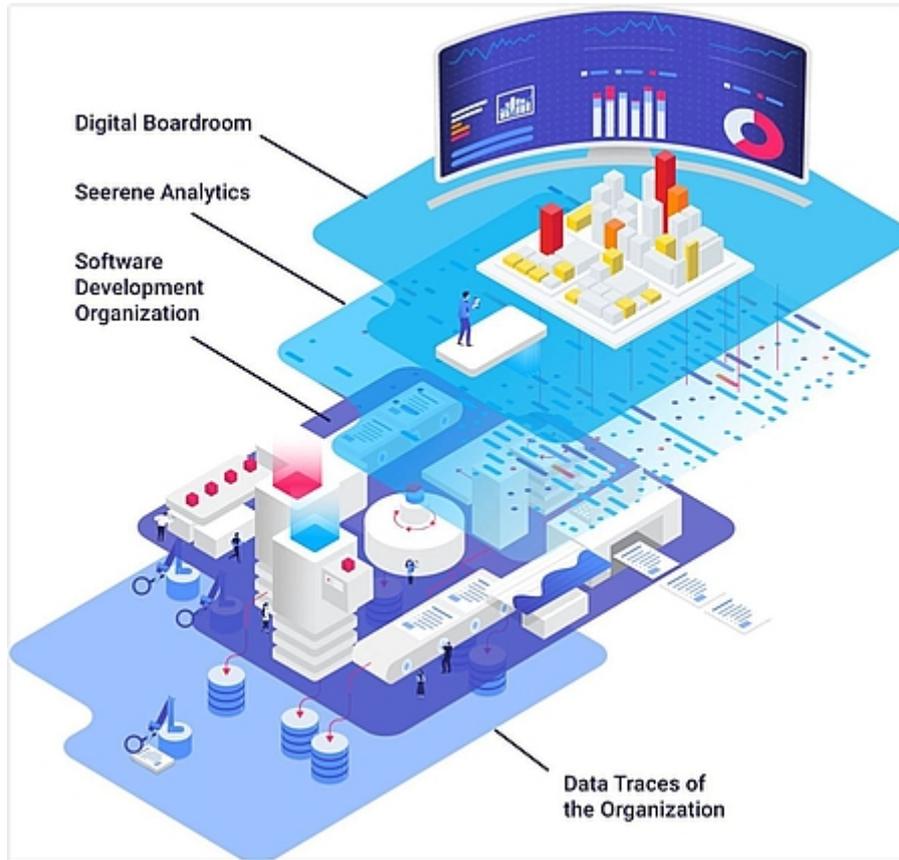
Natürlich geht es dabei letztlich immer nur um Wahrscheinlichkeiten. Deswegen ist die Wissensbasis entscheidend: Sie ist die Voraussetzung für die zuverlässige Prädiktion von Eigenschaften von Software-Systemen. Mit ihr kann ich Qualitätsaspekte, Konsequenzen und Fehlerwahrscheinlichkeiten vorhersagen. Und ich kann Aussagen über die Qualität der Software-Architektur, aber auch über die Zusammensetzung des Entwicklungsteams und seine Stärken, treffen.

Das geht ja über klassische Big-Data-Anwendungen weit hinaus. An welcher Stelle kommt KI ins Spiel?

Aussagen bezüglich der Zukunft beinhalten grundsätzlich eine Unschärfe, die sich schlecht durch einfache Regeln formulieren

lässt. Ich brauche dafür das ganze Repertoire an Prädiktionsverfahren, wie es KI bietet. Da grundsätzlich kein Software-System dem anderen gleicht, muss ich mit dieser inhärenten Unschärfe arbeiten, muss Kongruenzen ermitteln.

Mit Unschärfe und Unvollständigkeit umzugehen, ist eine wesentliche Eigenschaft von KI. Genau diese brauche ich bei der Prädiktion, wenn es um das eigene Software-System geht. Am Ende habe ich damit die Voraussetzungen für ein systematisches Benchmarking geschaffen.



Management einer Software-Entwicklungs-Organisation mit Software-Analytics

Wo stehen wir mit dem KI-Einsatz im Software-Engineering? Reden wir über Praxisanwendungen oder über universitäre Forschung?

KI für das Software-Engineering entwickelt sich in den letzten Jahren rasant. Wir sind damit längst aus dem Prototypenstadium heraus. Hardware und Software im KI-Bereich sind zum Glück generisch und können auf nahezu alle Anwendungsfelder angepasst werden. Daraus resultiert die hohe Entwicklungsgeschwindigkeit, weil wir keine anwendungsfeldspezifischen Technologien benötigen, sondern immer auf einen gemeinsamen Kern zurückgreifen können. Deshalb können wir heute schon industriestabile Lösungen für Software-Analytics bauen.

Dazu ist am HPI zum 1. Oktober das KI-Labor in Betrieb gegangen, das Methoden und Konzepte von KI für das IT-Systems-Engineering erforscht. Die Forscher haben hier die Möglichkeit, ihre Verfahren und Methoden in einer skalierbaren Hard- und Software-Umgebung zu erforschen. Der Schwerpunkt liegt auf der Software-Technik, auf der Frage, welche Aspekte bei der Entwicklung von KI-Systemen eine Rolle spielen. Dafür ist es uns wichtig, Industriedaten zu bekommen und industrierelevante Fragestellungen untersuchen zu können. Für diesen Zugang sorgen unsere assoziierten Partner – Seerene etwa kann große Industriesoftware-Projekte beisteuern. Daraus resultieren einzigartige Daten, die eine Universität sonst nicht zur Verfügung hätte.

Für die aktuelle KI-Studie rufen Sie Unternehmen zur Teilnahme auf. Geht es darum, dass Sie Zugang zu Projektdaten bekommen, oder welche Rolle spielen die Teilnehmer genau?

Die KI-Studie verfolgt zwei Ziele: Zum einen wollen wir Unternehmen begeistern, uns ein ausgewähltes Projekt als Untersuchungsgegenstand zu überlassen. Zum anderen hat sie den Zweck, dass Unternehmen die Möglichkeiten und Grenzen von KI für das Software-Engineering konkret kennenlernen. Wir wollen diese Projekte ein Jahr lang beobachten und Aussagen generieren, die wir mit den Unternehmen besprechen. So werden die Unternehmen genauer beurteilen können, wo im Software-

Entwicklungsprozess der Einsatz von KI für sie sinnvoll ist und wo nicht.

Der Vorteil für die Unternehmen liegt also darin, erste Erfahrungen zu sammeln und sich auszuprobieren?

Es ist ein Forschungsprojekt, an dem die Unternehmen sich beteiligen. Sie erhalten Zugang zu aktuellen Forschungsergebnissen und können sich früher als andere orientieren, inwieweit und wo KI für ihre eigenen Software-Entwicklungen eine Rolle spielen kann. Die Umsetzung in der Praxis erfolgt durch unseren Industriepartner Seerene mit seiner Software-Analytics-Plattform.

Wir gehen davon aus, dass sich schon im Rahmen der Studie messbare Verbesserungen in den Unternehmen erzielen lassen. Es erscheint uns wichtig, die Unternehmen mit Werkzeugen in Kontakt zu bringen, um zu sehen, wie diese in die internen Prozesse zu integrieren sind.

Wir müssen uns darüber im Klaren sein, dass kein Unternehmen es sich mittelfristig leisten kann, auf die Produktivitätssteigerung durch den Einsatz von KI im Software-Engineering zu verzichten. Wer in den nächsten Jahren seine Software-Entwicklung nicht zeitgemäß gestaltet und moderne Technologie dafür einsetzt, wird der Konkurrenz kaum standhalten. Da sprechen wir jetzt über einen Paradigmenwechsel: Für Unternehmen geht es nicht mehr darum, noch eine neue Programmiersprache, ein neues Framework, eine neue Datenbank oder ein neues Vorgehensmodell einzuführen. In den nächsten Jahren gilt es, den komplexen Herstellungsprozess von Software mit KI zu optimieren und neu zu gestalten.

Und bei diesem Paradigmenwechsel genießen die Studienteilnehmer dann den Vorteil des frühen Einstiegs?

Genau – sie haben den Vorteil des frühen Einsatzes dieser Technologie, um die eigene Software-Entwicklung konkurrenzfähig und zukunftsfähig zu machen. Dieser Paradigmenwechsel wird wesentlich mitentscheiden, ob Firmen mit ihrer Software-Entwicklung in der Zukunft erfolgreich sein können.

Wie kann man teilnehmen?

Die Studie beginnt im Januar, bis dahin nehmen wir Projektvorschläge entgegen. Mit den Bewerbern beraten wir dann, ob ihr Projekt geeignet ist – es soll eine gewisse Mindestgröße haben und schon eine gewisse Laufzeit erreicht haben. Dann fragen wir noch ab, welche Software-Repositoryen eingesetzt werden, um sicherzustellen, dass wir die Anbindung an unsere Analysensysteme realisieren können. Wenn alles passt, beobachten, interpretieren und analysieren wir ab Januar mit KI-basierten Methoden die Software-Entwicklungsprozesse. Am Ende bewerten und verbessern wir sie.

Herr Prof. Döllner, vielen Dank für das Interview!



Michael Ihringer

ist Gründer und Inhaber der in-house Agentur. Zuvor verantwortete er als Marketing-Direktor die PR und den Marktauftritt des amerikanischen Anbieters von Anwendungs-Technologien Intersystems in Deutschland, Österreich, der Schweiz und Osteuropa. Bis 1994 konnte Ihringer als Geschäftsführer eines mittelständischen Softwarehauses umfangreiche eigene Erfahrungen in Entwicklung und Vertrieb von Branchenanwendungen sammeln.

Bildnachweise

Privat, Seerene

[AI Trendletter](#)

[Impressum](#)

|

[Kontakt & Anfrage](#)