

SOFTWARE ENGINEERING

POSITIONSPAPIER

Während Software vor wenigen Jahren noch eine eher untergeordnete Rolle in einer von Hardware dominierten Welt einnahm, konnte sich das Software Engineering (dt.: Softwaretechnik) mittlerweile u.a. dank fortschreitender Standardisierung von Hardware als eigenständige Disziplin emanzipieren und treibt seither in ganzen Industrien Innovationen und Transformationsprozesse an. Software nimmt heute eine zentrale Rolle bei der digitalen Transformation ein. Die Softwaretechnik spielt dabei aufgrund der Dominanz der Software in neuen Technologien eine besondere Rolle: Software wird immer komplexer und ist allgegenwärtig. Vor allem aber durchdringen heute software-intensive Technologien sämtliche Lebensbereiche moderner Gesellschaften. Deshalb müssen einerseits junge Menschen besser auf die Nutzung der Potenziale software-intensiver Technologien, deren Risiken und die Anforderungen der Wissensgesellschaft vorbereitet werden; Bildung für einen sicheren, ressourcenschonenden und verantwortungsbewussten Umgang mit Softwarelösungen, etwa im Umfeld von IKT-Anwendungen, und der Ausgestaltung des sozialen Miteinanders kommt in einer digitalisierten Welt wachsende Bedeutung zu [1]. Andererseits liegt eine mindestens ebenso große Herausforderung darin, künftige Software-Ingenieure auszubilden. Insbesondere die Entwicklung komplexer Softwaresysteme erfordert neben fachlichen Kompetenzen ein hohes Maß an interdisziplinären und überfachlichen Kompetenzen. Hier muss das Bildungssystem in die Lage versetzt werden, kontinuierlich auf hohem Niveau Software-Ingenieure auszubilden.

Software Engineering treibt ohne Frage den digitalen Wandel wie keine andere Disziplin voran und befindet sich selbst in einem Wandel. Diesen Wandel zu durchdringen und konkrete Vorschläge für die Gestaltung von Software Engineering Themen zu erarbeiten ist das **Ziel des ZD.B Arbeitskreises Software Engineering**.

Insgesamt stellt der digitale Wandel die Wirtschaft, die Forschung und die Lehre gleichermaßen vor neue **Herausforderungen**. Zum einen herrscht in Bayern ein massiver Bedarf an qualifizierten Mitarbeiterinnen und Mitarbeitern, der aktuell nicht gedeckt werden kann. In 2017 gab es 55.000 offene Stellen mit Informatikbezug in Deutschland, davon 63% in der Software-Entwicklung.[2] Zum anderen müssen Inhalte und Formate - also insgesamt die Didaktik des Software Engineering - (weiter)entwickelt und überdacht werden, um dem Anspruch Software Engineering als Schlüsselkompetenz zu stärken und dem kontinuierlichen Wandel des Software Engineering gerecht zu werden. Zum Beispiel, welchen Stellenwert haben neue Vorgehensmodelle, etwa agile Methoden, in der Wirtschaft? Welchen in der Lehre? Wie lassen sich Kompetenzen im Software Engineering zielgerichtet adressieren? Wie kann Methodenwissen, etwa zu agilen Methoden, praxisnah und praxisrelevant eingeübt werden?

Neben der erforderlichen Reflexion über die notwendigen Inhalte und Formate innovativer Software Engineering Themen in der Lehre gilt es, die Zugänge zu den Basiskompetenzen der Softwaretechnik für andere Disziplinen, etwa im ingenieur-technischen oder wirtschaftlichen Bereich, zu öffnen. Softwaretechnik ist insgesamt nicht mehr allein in der Hochschullehre der Informatik essentiell, sondern immer mehr Disziplinen benötigen Grundkenntnisse im Software Engineering, noch dazu in wachsendem Umfang. Daher müssen wir ein gemeinsames Verständnis für ein Basiscurriculum erarbeiten, das die Vielfalt an Themen und Methoden auch außerhalb der Informatik transportiert, ohne dabei in Beliebigkeit zu verfallen. Gleichzeitig kann Software Engineering und seine Lehre an Hochschulen von methodischen Zugängen in andere Disziplinen stark profitieren, etwa den Geistes- oder Sozialwissenschaften, um den notwendigen Brückenschlag von einer bislang gestaltungsorientierten in eine erkenntnisorientierte Disziplin zu schaffen. Software wird von Menschen für Menschen erstellt; die damit verbundenen Prozesse unterliegen daher auch kulturellen Normen und Gesetzen, die es grundlegend zu erforschen und zu durchdringen gilt. Ebenso spielt der Mensch im Software Engineering eine zentrale Rolle – der Erfolg von Softwareentwicklungsprojekten hängt zum einen stark von einem funktionierenden Entwicklungsteam und zum anderen vom Funktionieren disziplinübergreifender Kommunikationsprozesse ab.

Damit der Brückenschlag hin zu einem vermehrt evidenzgetriebenen Software Engineering gelingt, muss das empirische Software Engineering mehr Wertschätzung in der Forschung erfahren. Darüber hinaus müssen bessere Zugänge zu Fördermöglichkeiten für Forschungsprojekte an Hochschulen mit hohem experimentellen Charakter geschaffen werden. Der Wandel in eine erkenntnisorientierte Disziplin ist in Forschung und Praxis bereits in vollem Gange, sollte sich aber auch etwa in öffentlichen Förderlinien widerspiegeln, damit sich Forschung auch losgelöst von konkreten industrielevanten Fragestellungen und Problemen entfalten kann. Um Bayern nachhaltig im Zuge der Digitalen Transformation und dem damit verbundenen Wandel im Software Engineering zu stärken, sehen wir insbesondere in folgenden Bereichen unmittelbaren **Handlungsbedarf**, dem wir uns im Rahmen des Arbeitskreises widmen möchten.

Stärkung des Lernens und der Lehre

Aktuell könnte oft der Eindruck entstehen, dass bei der Verbreiterung der Kompetenzen an staatlichen bayerischen Hochschulen und der damit verbundenen Schaffung neuer Professuren, etwa im Bereich Data Science and Engineering oder Artificial Intelligence, die kontinuierliche und nachhaltige Stärkung grundlegender Kompetenzen im Software Engineering zwar als wichtig erachtet, diese aber dennoch vernachlässigt und geschwächt wurden. Solche Basiskompetenzen umfassen:

[1] Siehe auch Hightech-Strategie Innovationen für Deutschland, Bundesministerium für Bildung und Forschung

[2] Bitkom Studie aus 2017 <https://www.bitkom.org/Presse/Anhaenge-an-Pls/2017/11-November/Bitkom-Charts-IT-Fachkraefte-07-11-2017-final.pdf>

- Technische Kompetenzen
- Methodenkompetenzen und weitere überfachliche Kompetenzen (wie Software Engineering-spezifische Soft Skills), etwa wenn es um den Umgang mit Anforderungen und damit verbundener Erhebungs- oder Verhandlungstechniken oder um praktische Projektorganisation und Management geht.

Gleichzeitig ist die Anzahl der Studierenden und der Bedarf an qualifiziertem Software Engineering Nachwuchs auf einem Rekordhoch.

Eine Bedarfserhebung und die damit verbundene Schaffung neuer Stellen (Mittelbau und Professuren) in der bayerischen Hochschul-landschaft stellt dabei einen wesentlichen und notwendigen Meilenstein dar. Die Förderung von Projekten, die sich vor allem der systematischen Verbesserung der Lehre widmen, sollte als weitere Möglichkeit in Betracht gezogen werden. Diese Projekte wären im Bereich (fach)didaktischer Fundierung oder der Erprobung neuer, auch interdisziplinärer, Lehrformate angesiedelt. Sie böten, z.B. als Verbundprojekte aufgesetzt, gleichzeitig eine hervorragende Gelegenheit, Synergien zwischen Universitäten, an denen eher grundlagenorientierte Themen verankert sind, und Hochschulen für angewandte Wissenschaften, an denen eher anwendungsnahe Themen untersucht werden, gezielt zu nutzen.

Stärkung der Forschung und Entwicklung

Die Forschung muss auf allen Ebenen ein stärkeres Bekenntnis zu Software Engineering als erkenntnisorientierte und evidenzgetriebene Disziplin erfahren. Ohne fundierte empirische Forschung wird es nicht gelingen, die Grenzen neuer Methoden und Technologien reproduzierbar auszutesten und den Transfer grundlagenorientierter Beiträge in die Praxis gezielt zu unterstützen. Dieser Wandel erfordert insbesondere:

- Schaffung von Möglichkeiten zur Förderung von Themen mit experimentellem Charakter an Hochschulen, ohne dabei zwingend in ein Konsortium mit Industrievertretern treten zu müssen, gleichzeitig aber eine hohe Anwendungsrelevanz der Themen sicherzustellen.
- Schaffung von Möglichkeiten zur Förderung von Projekten in interdisziplinären Konfigurationen und (Forschungs-)Methodenvielfalt. [3] Die Kombination von beispielsweise Psychologie, Sozialwissenschaften und Software Engineering erlaubt es Fragestellungen zu beantworten, die einzelne Disziplinen nicht umfassend adressieren können, etwa die Fragen, wie Mechanismen der Selbstorganisation funktionieren und wie diese verbessert werden können.

Stärkung ethisch reflektierten Handelns

Die Gestaltung von Informations- und Kommunikationstechnologie erfordert nicht nur die Berücksichtigung technischer Leistungsanforderungen, sondern auch die Einbeziehung ethischer Überlegungen. Die voranschreitende Digitalisierung hat eine tiefgreifende Veränderung der Gesellschaft und aller ihrer Lebensbereiche zur Folge. Um sowohl am menschlichen als auch am ökologischen Wohlergehen orientierte ökonomisch wie technisch verantwortungsbewusste Handlungs- und Entscheidungskriterien entwickeln zu können, wird eine kritische Reflexion von Chancen und Risiken bereits jetzt gängiger Praktiken und zukünftiger Möglichkeiten digitaler Technologien benötigt [4]. Kann ein rationales (autonomes) IT-basiertes System mit softwarebasierten Algorithmen so gesteuert werden, dass seine Entscheidungen den Erwartungswert der Handlungskonsequenzen optimieren kann? Die utilitaristische Ethik, die als Maxime die nützliche Machbarkeit von IT-Technologie ansieht, kollidiert unter anderem mit einem fundamentalen Prinzip jeder zivilen und humanen Gesellschaft, das von Nida-Rümelin [5]

als Nicht-Verrechenbarkeit skizziert wird. Wir müssen deshalb Antworten auf die Frage finden, wie in der Ausbildung des Software Engineering ein digitaler Humanismus die menschliche Verantwortungs-fähigkeit stärken kann.

AUTOREN

Daniel Méndez (TU München – ZD.B)

Jürgen Mottok (OTH Regensburg – ZD.B)

Jörg Abke (HAW Aschaffenburg)

Gerd Beneken (Hochschule Rosenheim)

Axel Böttcher (Hochschule München)

Konrad Doll (HAW Aschaffenburg – ZD.B)

Georg Hagel (HAW Kempten)

Dieter Landes (HAW Coburg)

Jan-H. Passoth (TU München)

Alexander Popp (Universität der Bundeswehr München)

Yvonne Sedelmaier (HAW Coburg)

Janet Siegmund (Universität Passau – ZD.B)

[3] Siehe auch "Empirical Software Engineering: From Discipline to Interdiscipline, <https://arxiv.org/abs/1805.08302>

[4] Siehe "Das Ethik-Manifest des IT-Sicherheitscluster Bayern e.V."

[5] Siehe Julian Nida-Rümelin: Digitaler Humanismus, Die Zeit, 1.10.2017