

Wie ein Beitrag für die SEUH aussehen sollte

Jochen Ludewig, Universität Stuttgart

ludewig@informatik.uni-stuttgart.de

Zusammenfassung

Achtung, hier dient ein bereits publizierter Artikel (Ludewig, 2010) als Blindtext der Formatvorlage, die für Einreichungen zur SEUH verwendet werden sollte. Entsprechend gibt es eine Abbildung und Literaturangaben, die nichts mit dem ursprünglichen Artikel zu tun haben. Die Formatvorlage orientiert sich an den Vorgaben der Softwaretechnik-Trends (Kelter, 2010).

Der Artikel beschreibt an drei Personas, welche Voraussetzungen Personen erfüllen sollten, die ein Studium der Softwaretechnik oder der Informatik mit dem Schwerpunkt Softwaretechnik beginnen. Er zeigt am Beispiel der Personas Anke, Bernd und Carl, wie ein erfolgreiches Studium verläuft und welche berufliche Perspektive es bietet.

Die wichtigsten Vorgaben

Für unseren elektronischen Tagungsband wollen wir mit möglichst geringem Aufwand ein einheitliches, ansprechendes Erscheinungsbild erzielen. Wir können aber nicht verhindern, dass die Autoren mit unterschiedlichen Textsystemen arbeiten. Darum stellen wir zwei Templates zur Verfügung, eines für \LaTeX , eines für Microsoft Word.

Wenn Sie eines dieser Templates verwenden, sollte damit bereits sichergestellt sein, dass Ihr Beitrag den Vorgaben entspricht. Wenn nicht, sollten Sie sich an den folgenden Angaben orientieren und am Ende das Erscheinungsbild Ihrer Seite mit dem der Muster (PDF-Darstellung) vergleichen, um eventuell noch weitere Anpassungen vorzunehmen.

Hier sind die wichtigsten Vorgaben, wie sie auch in diesem Muster verwendet werden:

- Rand unten 3 cm, sonst (oben, links und rechts) 2 cm. Bitte nutzen Sie das Schriftfeld möglichst gut aus. Bitte keine Seitennummerierung, keine Kopf- und Fußzeilen!
- Text in einem Serifenfont (wie hier Charter) in 10 pkt Schriftgröße. Manche andere Fonts (z. B. Times) fallen teilweise viel kleiner aus und erfordern dann eine 11 pkt Schrift.
- Überschriften in einem Sans Serif-Zeichensatz, z. B. Arial oder Helvetica. Der Titel (Format Titel) und die Autorenangaben sind nicht fett.

Formatvorlage	Schriftgröße	fett	oben	unten	Zeile
Titel	30		0	12	36
Autor	12		0	9	1fach
Zus.fass.	13	X	0	3	1fach
Üb.schr. 1	13	X	12	3	1fach
Üb.schr. 2	11	X	8	2	1fach
Üb.schr. 3	10	X	4	0	1fach
Standard	10		4	0	12
Folge	10		0	0	12
Referenz.	10		4	0	12

Tabelle 1: Formatdefinitionen

Die Formate sind in Tabelle 1 genauer beschrieben, die Formatbezeichnungen beziehen sich dabei auf das Word-Template. Nach Überschriften, Bildern und Tabellen steht das Format Standard; alle direkt folgenden Absätze verwenden das Format Folge, d. h. sie werden ohne Abstand mit Einrückung angehängt.

Listen (Format Liste1) werden 6 mm eingerückt; wenn sie nummeriert werden, ist evtl. eine stärkere Einrückung notwendig. Referenzen beginnen linksbündig, werden dann 6 mm eingerückt.

Titel und Überschriften werden nicht im Blocksatz formatiert, sondern zentriert (Titel und Autoren) bzw. linksbündig gesetzt. Bilder und Bildunterschriften sind zentriert und haben jeweils 12 pkt Abstand nach oben.

Bitte kontrollieren Sie am Ende die Formatierung Ihres Beitrags. Und prüfen Sie auch Rechtschreibung und Kommasetzung. Wir wollen und können Ihre Einreichung nicht bearbeiten; aber wir werden nur Beiträge annehmen, die im Wesentlichen den Vorgaben und den üblichen Ansprüchen an eine zu publizierende Arbeit entsprechen.

Anfang Blindtext

Wenn Sie überlegen, welches Studienfach Sie wählen sollten, kann dieser Text für Sie von Interesse sein. Er beschreibt an Anke, Bernd und Carl, drei typischen Studenten, welcher Weg vom Schulabschluss zum Beruf des Software-Ingenieurs führt.

Wir verwenden also das Konzept der „Persona“, das in der Softwaretechnik auch an anderen Stellen erfolgreich eingesetzt wird. Eine Persona ist eine fiktive

Person, die als Beispiel dient. Natürlich ist das Bild der Realität damit drastisch reduziert; aber die meisten unserer Absolventen werden sich in wenigstens einer der drei Figuren wiedererkennen.

Anke hat besondere Stärken auf der theoretischen, formalen Seite der Softwaretechnik. Bernd ist der Fachmann, der sich auszeichnet mit der Technologie auskennt, also mit den Methoden, Sprachen und Werkzeugen, und gern technische Probleme löst. Carl ist überdurchschnittlich auf dem Gebiet der Organisation und Kommunikation; wenn Projektgruppen gebildet werden, ist er sehr wahrscheinlich der Leiter seiner Gruppe.

Viele andere Profile sind denkbar und sinnvoll. Beispielsweise gibt es Qualifikationen, die stärker auf Systeme, also die Kombination von Hard- und Software ausgerichtet sind (Systems Engineering), auf die Gestaltung großer Softwaresysteme (Software-Architektur) oder auf die Anwendung der Software in Verwaltungen, Banken und Versicherungen (Wirtschaftsinformatik). Wir beschränken uns hier auf die drei genannten Profile, weil sie vermutlich von jedem Softwaretechnik-Studiengang akzeptiert und gefördert werden.

Was für jeden Software-Ingenieur gilt

Anke, Bernd und Carl studieren an deutschen Universitäten Softwaretechnik oder Informatik in einer Softwaretechnik-nahen Ausprägung. Wir nehmen an, dass sie nach dem Bachelor noch den Master und damit ein insgesamt mindestens zehensemestriges Studium absolvieren. Um Software-Ingenieur zu werden, reicht es aber nicht aus, mit irgendeinem Bachelor einen Master of Software Engineering zu erwerben. Schon das Bachelor-Curriculum muss deutlich auf die Softwaretechnik ausgerichtet sein.

Grundlagen

Anke, Bernd und Carl haben solide Grundkenntnisse in Mathematik. Die Analysis haben sie bereits auf der Schule gelernt, möglicherweise war sie im Studium noch zu ergänzen. Sie bildet aber keinen Schwerpunkt in der universitären Ausbildung.

Die Mathematik wird den zukünftigen Software-Ingenieuren als Informatik-spezifische Ingenieur-Mathematik vermittelt, also mit vielen praktischen Anwendungen. Anke nimmt zusätzlich an Lehrveranstaltungen für Mathematiker teil.

In Numerik, Stochastik und Statistik erhalten die drei ausreichende Grundlagen, um bei Bedarf daran

anzuknüpfen und sich die Themen selbst zu erschließen. Ob sie je davon Gebrauch machen werden, hängt stark von ihrer individuellen beruflichen Zukunft ab.

Anke, Bernd und Carl bekommen eine tragfähige Basis in Logik, diskreter Mathematik und Theoretischer Informatik. Vor allem verstehen sie, an welchen Stellen diese Grundlagen direkte Auswirkungen auf die Praxis haben. Modellierung und Abstraktion sind ihnen in diesem Zusammenhang von der formalen Seite aus nahegebracht worden; das wird in Lehrveranstaltungen des Software Engineerings durch praktische Anwendungen ergänzt.

Softwaretechnik

Im Zentrum des Studiums steht die Ausbildung in der praktischen Informatik, insbesondere in der Softwaretechnik.

Alle drei haben früh (spätestens im 3., besser im 2. Semester) eine Einführung in die Softwaretechnik erhalten. Der Umfang dieser Lehrveranstaltungen reicht aus, um sowohl die Management-Aspekte (also Softwareprojekt-Management, Qualitätssicherung, Planung und Kostenschätzung, Software-Prozesse und ihre Verbesserung) als auch die technischen Aspekte (Problemanalyse, Anforderungsspezifikation, Architekturentwurf usw. bis hin zum Re-Engineering) zu vermitteln. Je nach Angebot der Universität und Interessen der Studenten kommen Vertiefungen in speziellen Gebieten hinzu, beispielsweise im Requirements Engineering, in der Gestaltung der Bedienoberflächen oder in der Entwicklung von Systemen, die besonders zuverlässig sein müssen.

Vermittlung von Werten

Neben dem Wissen und Können vermitteln die Lehrveranstaltungen, sowohl die theoretischen als auch die praktischen, implizit das Wertesystem der Softwaretechnik, so dass die Absolventen „in der Wolle gefärbte“ Software-Ingenieure sind. Dazu trägt das Vorbild der Dozenten, die sich überwiegend als Software-Ingenieure sehen, wesentlich bei.

Codierung

Codierung ist ein wichtiger Bestandteil der Ausbildung, auch wenn sie natürlich nicht im Vordergrund steht. Jeder Software-Ingenieur sollte das Handwerk beherrschen, das die Basis seines Berufs bildet. Das geht weit über die Erstellung ausführbarer Programme hinaus. Darum werden Anke, Bernd und Carl vom ersten Semester an daran gewöhnt, einfache, gut lesbare Programme zu erstellen, deren Korrektheit nicht dem Zufall überlassen ist. Die Programmierung wird also nicht nach dem Motto „Hauptsache, es läuft“ betrieben. Vielmehr sind Standards, Stil und Form der Programme Gegenstand der Lehre und der Kontrolle im Übungsbetrieb.

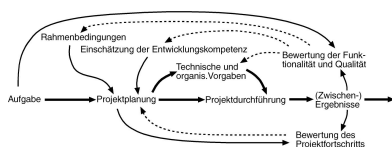


Abbildung 1: Eine Abbildung

Projekte und soziale Kompetenz

Hohes Gewicht hat auch die soziale Kompetenz; sie wird in Vorlesungen angesprochen und in den Seminaren, Praktika und Projekten geübt und reflektiert. Projektarbeit ist für alle drei obligatorisch. Das gilt insbesondere auch für Anke. Der Studienplan verhindert zuverlässig, dass der Abschluss ohne große Projekterfahrung erreicht werden kann.

Die Projekte sind zunächst inhaltlich und zeitlich klein, auf die Umsetzung klarer Anforderungen beschränkt. Im Laufe des Studiums verschiebt sich der Schwerpunkt hin zu den Problemen, die aus der Praxis bestens bekannt sind: Erhebung der Anforderungen, die alles andere als klar und vorgegeben sind, Koordination vieler Mitarbeiter im Projekt, Konflikte zwischen Termin-, Qualitäts- und Funktionalitätszielen, Arbeit mit neuen, keineswegs fehlerfreien Werkzeugen, Abstimmung und Koordination mit schwierigen Partnern, Urheberrechts- und Lizenzfragen. Anders als in der Praxis bietet das Studium die Möglichkeit, solche Probleme gründlich zu diskutieren und systematisch nach Lösungen zu suchen.

Anwendungsfach, ethische und soziale Einbettung

Der Fachbereich, an dem Anke, Bernd und Carl studieren, hat einen speziellen Schwerpunkt der Anwendung, beispielsweise die Softwaretechnik in der Industrie oder in der Medizin. Entsprechend gehören Lehrveranstaltungen aus einem Anwendungsfach zum Studienplan. Neben der Spezialisierung (die in vielen Fällen nicht die berufliche Zukunft prägt) wird allen Teilnehmern dieser Lehrveranstaltungen Respekt vor ihren zukünftigen Partnern und Kunden beigebracht, die Fähigkeit, sich mit kulturell anders geprägten Menschen, beispielsweise Juristen oder Bauingenieuren, über Software und Systeme auszutauschen.

Die ethischen Grundsätze und (vor allem) die Konzepte hinter einer Ethik sind im Studium thematisiert. In Seminaren haben sich die drei mit diesem Komplex aktiv auseinandergesetzt.

Im Laufe ihres Studiums treten sie einem Berufsverband bei, Anke der ACM, Bernd der IEEE-Computer Society und dem FiFF, Carl der GI. So bekommen

sie zu sehr geringen Kosten eine Fachzeitschrift und Informatik-Nachrichten aus der ganzen Welt.

Anke, Bernd und Carl suchen (und nutzen) die Gelegenheit, ihr Studium für einige Monate zu unterbrechen und ein (in der Regel gut bezahltes) Praktikum zu absolvieren, womöglich im Ausland. Damit erweitern sie ihre Erfahrungen und Sprachkenntnisse.

Spezialisierungen und Nebentätigkeiten

Anke reizt in ihrem Studienplan die Möglichkeiten zur Vertiefung in Mathematik und Theorie aus. Das gilt auch für die Wahl der Projekte und der Abschlussarbeit. Sie finanziert ihr Studium teilweise durch eine Hilfskraft-Tätigkeit als Tutorin.

Auch Bernd verdient (durch einen Job als Programmierer in einer mittelständischen Firma) Geld dazu, bis er das Stipendium eines Softwarehauses bekommt, mit dem er sich festlegt, dort nach dem Abschluss zu arbeiten.

Carl erwirbt Spezialwissen in BWL und zum Software-Recht. Zur Aufbesserung seiner Kasse übernimmt er immer wieder kleine Aufträge, bei denen es darum geht, Systeme aus Hard- und Software für eine spezielle Klientel zu beschaffen und zu installieren.

Anke, Bernd und Carl im Arbeitsleben

Anke sucht sich eine Möglichkeit zur Promotion. Nach Abgabe ihrer Dissertation arbeitet sie in einer Firma, die zur Entwicklung hochzuverlässiger Software formale Techniken einsetzt. Anke bemüht sich weiterhin, neben ihrer Arbeit auch Publikationen zu verfassen, so dass sie den Kontakt zur Wissenschaft nicht verliert.

Literatur

[Kelter 2010] KELTER, Udo: *Softwaretechnik-Trends*. 2010. – (Zeitschrift des GI-Fachbereichs Softwaretechnik.)

[Ludewig 2010] LUDEWIG, Jochen: Software-Ingenieur werden. In: *Informatik Spektrum* 33 (2010), Nr. 3, S. 288–291