

Softwaretechnik – eine Begriffsbestimmung

Udo Kelter

14.03.2003

Zusammenfassung dieses Lehrmoduls

Gegenstand der Softwaretechnik ist die kosteneffiziente Entwicklung qualitativ hochwertiger Softwaresysteme. Diese Begriffsbestimmung bietet noch sehr viel Interpretationsspielraum: weit gefaßt fällt die gesamte Praktische Informatik und mehr unter die Softwaretechnik, eng gefaßt ist es ein spezielles Forschungsgebiet; der Inhalt von Lehrbüchern mit dem Titel Softwaretechnik orientiert sich wiederum an anderen Kriterien. In diesem Lehrmodul untersuchen wir den Begriff Softwaretechnik aus mehreren Perspektiven. Eine wesentliche Basis unserer Betrachtungen ist eine Analyse des Arbeitsmarkts für Informatiker, aus der wir Anforderungen an die Ausbildung von Informatikern im allgemeinen und an die Inhalte einer Softwaretechnik-Vorlesung im besonderen ableiten.

Vorausgesetzte Lehrmodule: keine

Stoffumfang in Vorlesungsdoppelstunden: 1.0

Inhaltsverzeichnis

1	Eine erste Eingrenzung	3
2	Zum Selbstverständnis von Ingenieurdisziplinen	5
3	Das Qualifikationsprofil von Informatikern – eine Analyse aus Sicht des Arbeitsmarkts	10
3.1	Probleme bei der Analyse des Arbeitsmarkts	11
3.2	Eine Grobeinteilung des Arbeitsmarkts	12
3.3	Erwartete Qualifikationen von Informatikern	14
4	Softwaretechnik als Lehrgebiet	15
	Literatur	17
	Index	17

1 Eine erste Eingrenzung

Was ist eigentlich **Softwaretechnik**¹ (*software engineering*)?

Man könnte auf diese Frage ganz pragmatisch mit einer Liste von Prüfungsfragen antworten, die man vor Ort in der Prüfung beantworten können sollte. Das wäre aber, wie sich gleich zeigen wird, zu kurz-sichtig, der Begriff Softwaretechnik ist mehrdeutig, und die Begriffsvarianten sollten den Lesern² dieser Buchs durchaus bewußt sein. Ferner bietet die Suche nach der Definition des Begriffs Softwaretechnik einen guten Anlaß, sich explizit mit dem generellen Selbstverständnis eines Ingenieurs auseinanderzusetzen.

Zunächst weist der Wortbestandteil “-technik” (*engineering*) darauf hin, daß es sich hier um eine Ingenieurdisziplin handelt. Das Lexikon definiert den **Ingenieur** als einen “wissenschaftlich gebildeten Fachmann der Technik, der technische Gegenstände, Anlagen, Verfahren oder Systeme erforscht, plant, entwirft, konstruiert, fertigt, vertreibt, überwacht oder verwaltet”. Deutsche Ingenieure haben im internationalen Vergleich den Ruf, (zu) technikverliebt zu sein, also ein Maß an technischer Perfektion anzustreben, das wirtschaftlich nicht mehr rentabel ist. Von einem modernen Ingenieur wird man daher generell ein kosteneffizientes Handeln in allen o.g. Bereichen verlangen.

Softwaretechnik als Ingenieurwissenschaft. Faßt man die Wortbildung “Softwaretechnik” analog auf wie die Bezeichnung Elektrotechnik, also die Ingenieurwissenschaft, die unterschiedlichste Bereiche abdeckt, in die elektrische Geräte oder Bauteile involviert sind,

¹ Teilweise wird auch die Bezeichnung Software-Technologie (*software technology*), also die Lehre von der Softwaretechnik, benutzt. Daß an Universitäten Lehre betrieben wird, braucht aber nicht besonders betont zu werden.

² Personenbezeichnungen werde ich in diesem Text – wie im allgemeinen Sprachgebrauch üblich – in der männlichen Form im Sinne einer geschlechtsneutralen Bezeichnung verwenden; in der Softwaretechnik besteht kein Anlaß, zwischen weiblichen und männlichen Personen zu unterscheiden. Da die Verwendung der männlichen Form von manchen Personen als nicht geschlechtsneutral angesehen wird, werden als Kompensation in diesem Text konkrete Personen in Beispielen meist weiblichen Geschlechts sein.

dann ist die Softwaretechnik die Ingenieurwissenschaft, die unterschiedlichste Bereiche abdeckt, in die informationsverarbeitende Systeme involviert sind. Mit dieser Definition deckt die Softwaretechnik thematisch nahezu die gesamte Informatik ab, auf jeden Fall die komplette praktische Informatik, ferner große Teile der theoretischen Informatik. Bisweilen wird der Begriff Softwaretechnik im allgemeinen Sprachgebrauch tatsächlich so breit verstanden; ein prominentes Beispiel für dieses Verständnis ist der Studiengang (!) Softwaretechnik an der Universität Stuttgart³. Ludewig [Lu99] argumentiert pointiert in Richtung dieses Verständnisses, und er stellt die griffige Formel auf, daß sich die Softwaretechnik zur Informatik verhält wie die Elektrotechnik zur Physik. So verstanden ist Softwaretechnik nicht reduzierbar auf den Inhalt einer einzelnen Veranstaltung, sondern durchzieht das gesamte Studium.

Softwaretechnik als Teilgebiet der Praktischen Informatik. Üblicherweise hingegen versteht man die Softwaretechnik als ein Teilgebiet der praktischen Informatik unter anderen (wie Compilerbau, Betriebssysteme, Datenbanken usw.); in diesem Fall muß das Themenspektrum deutlich enger gefaßt werden. Ferner muß noch unterschieden werden zwischen der Auffassung als Wissenschaftsgebiet und als Lehrgebiet.

Das *Wissenschaftsgebiet* umfaßt die Kernthemen des Gebiets, aber auch viele verästelte und spezialisierte Themen; die wissenschaftlichen Aktivitäten liegen typischerweise eher in speziellen neuen Problemfeldern. Durchforstet man also Konferenzbände und forschungsorientierte Zeitschriften zum Thema Softwaretechnik, wird man vor allem auf derartige spezielle Themen stoßen.

Breite, um nicht zu sagen vage Definitionen verwenden auch Standardsorganisationen oder Normierungsgremien, wenn sie den Begriff Softwaretechnik definieren. Ein Beispiel ist die folgende Definition, die dem IEEE Standard Glossary of Software Engineering Terminolo-

³Dieser steht dort als bewußte Alternative in Konkurrenz zum Studiengang Informatik.

gy entnommen ist: *software engineering* ist danach *the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software, that is, the application of engineering to software*. Derartige Definitionen sind teilweise durch die Frage geprägt, was denn ein Softwaretechniker mit seinem Wissen Nützliches anfangen kann.

Der Themenumfang des *Lehrgebiets* kann eher am typischen Inhalt von Lehrbüchern abgelesen werden; er konzentriert sich normalerweise auf die Kernthemen, die “wichtig” und bereits “gereift” sind. Analysiert man gängige Lehrbücher über Softwaretechnik, findet man einige Themen sehr häufig, was auf einen hohen Konsens unter den Autoren hindeutet. Die gängigsten Themen umfassen:

- Systemanalyse mit Methoden bzw. Notationen wie Entity-Relationship-Diagrammen, objektorientierter Analyse (OOA) unter Verwendung diverser Diagrammtypen der UML usw.
- Programmwurfsmethoden und zugehörige Beschreibungssprachen für Software-Architekturen
- Qualitätssicherung
- Vorgehensmodelle, Aufwandsschätzung und Projektmanagement

Die Stoffauswahl in Lehrbüchern ist im übrigen auch beeinflusst durch das Ziel, Themen auszugrenzen, die schon in anderen typischen Veranstaltungen im Informatik-Curriculum abgedeckt sind.

2 Zum Selbstverständnis von Ingenieurdisziplinen

Im vorigen Abschnitt haben wir erwähnt, daß Softwaretechniker ingenieurmäßig arbeiten (sollten), ohne diesen Begriff näher zu qualifizieren. Dies soll hier nachgeholt werden.

Worin unterscheidet sich die Arbeitsweise eines Ingenieurs von der eines Rechtsanwalts oder Schriftstellers? Wir konzentrieren uns hier auf Entwicklungstätigkeiten.

Zunächst gehört zu jeder Ingenieurdisziplin eine bestimmte Klasse von Systemen, die Ingenieure dieser Disziplin konstruieren können. Ein Statiker kann die Dimensionierung eines Betonsturzes ausrechnen, nicht hingegen die Dimensionierung eines Leistungstransformators. In breiten Gebieten wie der Elektrotechnik oder der Informatik spezialisieren sich Ingenieure auf einzelne Teilgebiete, entweder schon in der Ausbildung oder später durch die berufliche Praxis.

Innerhalb einer Systemklasse gibt es i.a. alternative Technologien und Produkte, mit denen sich das gleiche Problem lösen läßt. Daten können bspw. drahtgebunden oder drahtlos per Funk oder Infrarotstrahlung übertragen werden. Jede einzelne Technologie hat Vor- und Nachteile, die ihren Anwendungsbereich bestimmen. Eine erste Leistung eines Ingenieurs besteht somit darin, zu einer gegebenen Problemstellung die geeigneten Technologien zu sichten und unter Berücksichtigung aller Randbedingungen die am besten geeignete auszuwählen.

Standardisierte Vorgehensmodelle und Architekturen. Durch die Wahl einer bestimmten Technologie oder Kombination von Technologien wird oft eine Grob- oder sogar Detailarchitektur des Systems vorgegeben, die wesentliche Komponenten und Schnittstellen des Systems festlegt. Die Komponenten sind vielfach standardisiert, für ihre Entwicklung brauchen also keine Aufwände mehr angesetzt zu werden. Hierzu ein Beispiel aus der Informatik: Informationssysteme werden vielfach als Client-Server-System realisiert, wobei ein Client-Rechner über ein Netzwerk und TCP/IP-Protokolle mit einem Server-Rechner kommuniziert und wobei serverseitig ein Applikationsprozeß über eine SQL-Schnittstelle mit einem Datenbankserverprozeß kommuniziert. Tatsächlich wird man sich in diesem Beispiel nicht nur auf den SQL-Standard, der i.w. nur die Abfragesprache spezifiziert, festlegen, sondern sogar auf ein konkretes Datenbankmanagementsystem, das diverse benötigte Zusatzfunktionen anbietet.

Eine Technologie ist immer auf eine bestimmte Klasse von Problemen ausgerichtet; je genauer diese Klasse eingegrenzt ist, desto mehr und detailliertere Details der Architektur können durch die Technolo-

gie vorgegeben werden.

Nach Auswahl der Technologie(n) besteht die weitere Arbeit darin, die vordefinierten Komponenten anzupassen oder im Einzelfall Komponenten auch völlig neu zu konstruieren. Insgesamt ist aber die weitere Vorgehensweise gut vorstrukturiert und muß nicht immer wieder neu erfunden werden. Die Lösung eines Problems hängt nicht von genialen Geistesblitzen oder künstlerischen Eingebungen des Ingenieurs ab, eine tragfähige Lösung kann von jedem ausgebildeten Ingenieur entwickelt werden. In manchen Bereichen geht die Vorstrukturierung so weit, daß verschiedene Ingenieure mit hoher Wahrscheinlichkeit ähnliche Lösungen entwickeln werden. Nichtsdestotrotz verbleiben i.d.R. genügend Freiräume, in denen ein Ingenieur seine Kreativität entfalten kann.

Die Verwendung von Standardarchitekturen und -komponenten bringt ganz enorme Vorteile mit sich:

- Es ist kein Aufwand für eine komplette Neuentwicklung erforderlich. In der Standardarchitektur ist sehr viel Erfahrung kondensiert, die nicht jeder Ingenieur neu sammeln muß.
- Die Standardarchitektur reduziert die Komplexität des Gesamtsystems auf überschaubarere, leichter behandelbare Komponenten. Sie vereinfacht es, funktionale bzw. qualitative Eigenschaften (z.B. Ausfallsicherheit, Lebensdauer) des Systems vorherzusagen.
- Sie bildet eine Grundlage für die Projektplanung und Kostenschätzung.

Qualitätssicherung. Ein ganz zentraler Aspekt ingenieurmäßigen Arbeitens ist ferner die Qualitätssicherung. Ausgangspunkt ist die Erkenntnis, daß immer Einflußfaktoren vorhanden sind, die zu Qualitätseinbußen führen. Diese können wir ganz grob wie folgt klassifizieren:

1. Konstruktionsfehler
2. Materialfehler bzw. Fehler im Herstellungsprozeß der Produkte

Wenn die möglichen Schäden hoch sind, wird man die Risiken sogar quantifizieren; beispielsweise interessiert die Wahrscheinlichkeit eines GAUs bei einem Atomkraftwerk.

Nach dem Erkennen und ggf. Quantifizieren der Risiken wird man geeignete Gegenmaßnahmen planen und durchführen. Bei Konstruktionsfehlern wendet man z.B. das “4-Augen-Prinzip” an, d.h. ein Konstruktionsergebnis wird von einer unabhängigen anderen Person kontrolliert. Beispielsweise wird die Statik, die unser Statiker berechnet hat, im Bauamt noch einmal nachgerechnet. Gegen Materialfehler geht man z.B. durch redundante Komponenten an. Die Maßnahmen, die im konkreten Fall ergriffen werden, müssen auf die geforderte Qualität bzw. Zuverlässigkeit abgestimmt werden; dies bedingt, derartige Qualitätsmerkmale schon bei der Systemplanung zu spezifizieren.

Die Qualitätssicherungs- (QS-) maßnahmen kosten letztlich Geld; diese Kosten sollten natürlich minimiert werden. Sofern also alternative QS-Maßnahmen verfügbar sind, ist die kostengünstigste zu wählen. Hierzu ein Beispiel aus der Informatik: Es ist empirisch nachweisbar, daß beim Suchen nach Fehlern in Programmen mit Reviews deutlich mehr Fehler pro Arbeitsstunde gefunden werden als mit Testverfahren; daher sollten zunächst Reviews als QS-Maßnahme eingesetzt werden.

Einer der wichtigsten Einflußfaktoren sowohl auf die Qualität als auch auf die Kosten ist der Ingenieur selbst! Qualitätsmängel können von direkten Fehlern bis hin zu einer ungünstigen Gestaltung (die aber nicht die Funktion des Gesamtsystems in Frage stellt) reichen. Qualitätsverbesserungen in diesem Punkt können nur durch bessere Ausbildung und Hebung des Qualitätsbewußtseins der Ingenieure erreicht werden.

Kostenbewußtsein. Direkten Einfluß auf die Kosten hat die Produktivität eines Ingenieurs. Bei der Entwicklung von Systemen, deren Größe sich in einer bestimmten Einheit messen läßt, ist die Produktivität eines Ingenieurs gegeben durch die Zahl der Einheiten, die er durchschnittlich pro Arbeitstag (oder -stunde) entwickelt. Neben dem Erstentwurf des Systems sind hierin auch alle Aufwände einzubezie-

hen, die zur Entdeckung von Fehlern und für ggf. folgende Nachbesserungen erforderlich sind. Hier zeigt sich, daß die Arbeitsqualität eines Ingenieurs direkten Einfluß auf die Entwicklungskosten hat und daß es auch aus dieser Perspektive unerläßlich ist, daß Ingenieure ein hohes persönliches Qualitätsniveau erreichen und halten.

Standesorganisationen. Wenn wir einmal davon ausgehen, daß nach einem erfolgreichen Studium (und einer ggf. anschließenden Phase der Spezialisierung in der Anfangszeit des Berufslebens) initial ein ausreichendes Qualitätsniveau erreicht ist, stellt sich die Frage, wie dieses angesichts der dynamischen Entwicklung in der Informatik auf Dauer erhalten werden kann. Das Bewußtsein, daß kontinuierliche Weiterbildung wichtig und notwendig ist, fehlt vielen Arbeitgebern, so daß auf alle Fälle Eigeninitiative jedes Absolventen sinnvoll ist. In diesem Kontext ist vor allem die Gesellschaft für Informatik (GI), die Standesorganisation der Informatiker, zu erwähnen. Die Ziele und Aktivitäten der GI sind analog strukturiert wie die anderer Standesorganisationen (z.B. VDE oder VDI):

- In einer Vielzahl von Fachgruppen werden einzelne Themen diskutiert. So gibt es u.a. Fachgruppen zu den Themen Softwaretechnik, Qualitätssicherung und objektorientierte Systementwicklung. Die Fachgruppen veranstalten Arbeitstreffen, die insb. auch dem Erfahrungsaustausch unter Berufstätigen dienen.
- Die GI gibt diverse Zeitschriften heraus, u.a. das “Informatik-Spektrum”, das vor allem Übersichtsartikel zu aktuellen Themen auf einem für Akademiker angemessenen Niveau anbietet.
- Schließlich veranstaltet die GI noch diverse wissenschaftliche Fachtagungen, die inhaltlich von den entsprechenden Fachgliederungen gestaltet werden, sowie Tutorien, die eher üblichen kommerziellen Angeboten ähneln.

Standesorganisationen sind allerdings nicht nur mit der Weiterbildung ihrer Mitglieder befaßt, sondern entwickeln vielfach Empfehlungen zur Ausbildung, also der Gestaltung der einschlägigen Studiengänge und Ausbildungsberufe, bis hin zu allgemeinen ethischen

Leitlinien. Darüber hinaus nehmen Standesorganisationen Stellung zu einschlägigen aktuellen politischen Themen und wirken teilweise in Normierungsgremien mit.

Ob eine Ingenieurdisziplin ein Selbstverständnis entwickelt, hängt daher nicht zuletzt ab von der Existenz einer Standesorganisation⁴ und von der Selbstverständlichkeit, mit der die “Professionellen” Mitglied in ihrer Standesorganisation sind.

3 Das Qualifikationsprofil von Informatikern – eine Analyse aus Sicht des Arbeitsmarkts

Wir haben uns im vorigen Abschnitt sehr allgemein mit dem Selbstverständnis von Ingenieuren befaßt. In diesem Abschnitt wollen wir uns konkreter mit “Software-Ingenieuren” befassen und hierzu untersuchen, welche Qualifikationen und praktischen Lösungskompetenzen von Informatikern im Berufsleben – also auf dem Arbeitsmarkt – erwartet werden. Unser Ziel ist, hieraus Anforderungen an die Informatikausbildung im allgemeinen und an die Stoffauswahl einer Softwaretechnik-Vorlesung im besonderen abzuleiten. In der Tat wird sich ergeben, daß wir in einigen Punkten deutlich von der üblichen Stoffauswahl von Softwaretechnik-Vorlesungen abweichen werden.

Der Arbeitsmarkt für Informatiker kann nicht analysiert werden, ohne auf die Konkurrenzsituation zwischen Informatikern und Angehörigen anderer Berufsgruppen einzugehen. In diesem Zusammenhang muß vorausgeschickt werden, daß es der Informatik – leider – bisher nicht in ausreichendem Maße gelungen ist, professionelle Zuständigkeiten zu erringen, analog etwa zu Ärzten, Steuerberatern, Juristen, allen anderen Ingenieurberufen u.a. Mit professioneller Zuständigkeit ist hier gemeint, daß bestimmte Aufgaben aufgrund gesetzlicher Regelungen oder faktischer Gewohnheiten praktisch nur von Vertretern einer bestimmten Berufsgruppe übernommen werden. Das

⁴Standesorganisationen gibt es natürlich nicht nur in den Ingenieurdisziplinen, sondern auch für Ärzte, Mathematiker, Architekten und beliebige andere Berufe, insofern sind sie keine Spezialität von Ingenieurdisziplinen.

Fehlen professioneller Zuständigkeiten führt dazu, daß Informatiker sehr häufig in direkter Konkurrenz zu Angehörigen anderer Professionen stehen, wenn es um Stellenbesetzungen geht [Ha95].

Das Fehlen professioneller Zuständigkeiten korrespondiert übrigens auffallend mit dem geringen Organisationsgrad der Informatiker: Die Gesellschaft für Informatik repräsentiert mit rund 20.000 Mitgliedern nur ca. 10 - 20 % der hauptberuflich in der Informatik tätigen Arbeitnehmer.

3.1 Probleme bei der Analyse des Arbeitsmarkts

Idealerweise ginge man bei der Definition des Qualifikationsprofils von Informatikern davon aus, welche Tätigkeiten Informatiker im Berufsleben tatsächlich ausüben und welche Qualifikationen der Arbeitsmarkt von ihnen verlangt. Eine derartige Beschreibung des Arbeitsmarkts, die einigermaßen präzise ist, ist leider nicht einfach zu erhalten:

- Die Informatik ist eine vergleichsweise junge Disziplin, die sich (in einigen Bereichen) immer noch rasant weiterentwickelt und daher ständig zu neuen Tätigkeitsprofilen führt. Aus der Vergangenheit kann nicht einfach linear in die Zukunft projiziert werden.
- Informatiker werden in den unterschiedlichsten Anwendungsbereichen eingesetzt; dementsprechend vielfältig sind die Methoden, mit denen Software entwickelt wird, und dementsprechend gering sind die Gemeinsamkeiten zwischen entfernt voneinander liegenden Anwendungsbereichen.
- Informatik-Tätigkeiten werden nicht nur von Informatikern (hier als Sammelbegriff für klassische Informatiker, Wirtschaftsinformatiker, Technische Informatiker usw. verstanden) übernommen, sondern auch von Angehörigen anderer Berufe oder “Umschülern”. Diese Unterschiede sind vielfach in den Betrieben nicht genau erfaßt, d.h. eine exakte Beschreibung der Verhältnisse auf dem Arbeitsmarkt ist fast unmöglich.
- Die Nachfrage nach Arbeitnehmern in den Informatikberufen konnte in den letzten ca. 20 Jahren auch nicht annähernd mit Absolven-

ten der Informatikstudiengänge befriedigt werden: Den ca. 25.000 Absolventen von 1973 bis 1990⁵ stand ein Mehrfaches an offenen Stellen gegenüber. Dies führte dazu, daß aufgrund des massiven Nachfrageüberhangs⁶ sehr viele Stellen mit “unterqualifiziertem” Personal besetzt wurden, während Informatiker (u.a. wegen der höheren Gehälter) nur in bestimmten Bereichen eingesetzt werden. Bei einem ausgeglichenen Verhältnis von Angebot und Nachfrage würden sich auch die Tätigkeitsbereiche von Informatikern verschieben.

Aufgrund der vorstehenden Beobachtungen kann es nicht überraschen, daß es wenig aktuelle und quantitativ detaillierte Analysen der beruflichen Tätigkeiten von Informatikern gibt. Insgesamt kann also Zahlenmaterial über den vorhandenen Arbeitsmarkt, soweit er sich überhaupt beobachten läßt, nur recht grobe Anhaltspunkte liefern.

3.2 Eine Grobeinteilung des Arbeitsmarkts

Eine der wenigen ausführlichen Analysen des Arbeitsmarkts für Informatiker findet sich in [Ha95]. Dort wird ganz grob unterschieden zwischen

1. der Entwicklung von Standardsoftware, also Produkten, die einem relativ großen, anonymen Markt zu festen Preisen angeboten werden und die üblicherweise längerfristig weiterentwickelt werden;
2. der Entwicklung von Individualsoftware, also Systemen, die im Auftrag eines Anwenders speziell für diesen entwickelt werden. Man spricht hier auch oft vom Projektgeschäft. Vielfach geht es hier auch darum, einen Kunden zunächst dahingehend zu beraten, welche Funktionen durch Software unterstützt werden sollen.

Die Grenze zwischen diesen Arbeitsmärkten ist nicht ganz scharf. Hinzu kommen ferner diverse an dieser Stelle weniger wichtige Tätigkeitstypen, z.B. Ausbildung. Keine Rolle spielt, ob die Tätigkeiten bei

⁵Quelle: [Ha95], S. 36.

⁶Derartige Boom-Zeiten lagen in den 70er Jahren vor, ferner wieder seit ca. 1998.

einem Software-Haus oder in einer internen Abteilung eines großen Anwenders oder Herstellers anfallen.

Die Tätigkeiten, die bei der Entwicklung von Standardsoftware anfallen, erfordern meist eine sehr hohe fachliche Qualifikation; dies erklärt den vielfach sehr hohen prozentualen Anteil an Informatikern in den Entwicklergruppen (bis zu 90 %). Allerdings wird in Deutschland wenig Standardsoftware entwickelt; hier dominieren eindeutig die U.S.A. als Lieferant von z.B. Betriebssystemen, Datenbankmanagementsystemen, Office-Paketen usw.⁷. Daher ist der Gesamtmarkt für diesen Tätigkeitstyp vergleichsweise klein.

Bei der Entwicklung von Individualsoftware ist der Mittelrahmen üblicherweise wesentlich kleiner als bei Standardsoftware; der Kostendruck macht eine hohe Arbeitseffizienz unverzichtbar. Daher geht es vielfach eher darum, Standard-Produkte anzupassen bzw. zu konfigurieren und nur punktuell durch Software, die klassisch “von Null an” entwickelt worden ist, zu ergänzen. Die Beratung des Kunden und Installation der Systeme beim Kunden - beides keine Entwicklertätigkeiten im klassischen Sinn - haben hier oft einen signifikanten Anteil.

In beiden Bereichen gilt, daß bei stark innovativen Entwicklungen ein besonders hoher Prozentsatz an Informatikern eingesetzt wird, da nur ihnen eine ausreichend breite Qualifikation zugebilligt wird. Derartige technisch anspruchsvolle Tätigkeiten sind zwar sehr beliebt bei Informatikern, diese Arbeitsmärkte sind absolut gesehen aber vergleichsweise klein.

Generell zu beobachten ist eine zunehmende Spezialisierung der Entwickler, insb. bei großen Arbeitgebern. Dies erklärt sich u.a. aus der immer größeren Komplexität der Systeme, die zu einem entsprechend hohen Lernaufwand führt, der nur durch eine längerdauernde Anwendung des Erlernten amortisiert werden kann. Insbesondere bei kleineren Arbeitgebern sind vor allem lösungsorientierte “Generalisten” gefragt, die eher den betrieblichen Nutzen der Systeme beachten und weniger die technische Eleganz.

⁷Das System R/3 der SAP ist eher die Ausnahme, die die Regel bestätigt.

Wie schon einleitend gesagt, ist es keineswegs sicher, daß diese derzeitigen - ohnehin vergrößert dargestellten - Charakteristika der Arbeitsmärkte auf unbestimmte Zeit erhalten bleiben.

Festgehalten werden kann immerhin, daß der vielfach vorhandene Eindruck, daß Informatiker fast nur programmieren oder dem Programmieren nahestehende Tätigkeiten ausüben, generell nicht zutrifft. Dieser Eindruck entsteht zu Beginn des Studiums u.a. dadurch, daß die Informatikausbildung in den beiden ersten Semestern - notwendigerweise - daran orientiert ist, Grundlagen über den Aufbau und die Programmierung von Rechnern zu vermitteln. Besonderes Gewicht liegt dabei darauf, das Programmieren zu erlernen. Im beruflichen Alltag machen die Programmier Tätigkeiten oft nur einen Bruchteil der Arbeitszeit aus⁸; einen erheblichen, oft sogar größeren Raum nehmen planerische und kommunikative Tätigkeiten ein.

3.3 Erwartete Qualifikationen von Informatikern

Unabhängig davon, wie sich die Informatiker tatsächlich auf die Arbeitsmarktsegmente verteilen, kann man die Frage nach der erforderlichen Qualifikation auch anhand der Erwartungen der Personalverantwortlichen untersuchen. Die generelle Erwartungshaltung an die Qualifikation von akademisch ausgebildeten Informatikern kann wie folgt stichwortartig zusammengefaßt werden⁹:

- breites Allgemeinwissen, sowohl praktisch als auch bzgl. der theoretischen Grundlagen
- der (ideale) Informatiker übersieht keine wesentlichen Einflußfaktoren, erfaßt alle Aspekte des Problems, hat Blick für die prinzipielle Machbarkeit, erkennt Risikofaktoren
- systematische Herangehensweise und Arbeitsmethodik (die oft letztlich am kosteneffizientesten ist), strukturiertes Denken

⁸Dies soll *nicht* zu dem Fehlschluß verleiten, daß gute Programmierkenntnisse verzichtbar seien!

⁹Die meisten dieser Eigenschaften werden übrigens analog von Ingenieuren anderer Fachgebiete erwartet.

- Abstraktionsfähigkeit
- Fähigkeit, über den Tellerrand des Tagesgeschäfts hinauszusehen und längerfristige Entwicklungen einzuschätzen
- ein gewisses Maß an Forschungsinteresse

Diese Eigenschaften werden nicht zuletzt als Gegensatz zu den Qualifikationen von Absolventen anderer Disziplinen gesehen, die nur eine weniger umfangreiche Nebenfachausbildung in Informatik erhalten haben, bzw. den Qualifikationen von “Amateuren”, Umschülern, Fachhochschulabsolventen und “DV-Praktikern”; denen typischerweise eher die Bedienung konkreter Systeme anvertraut wird und die daher Detailwissen über diese Systeme benötigen oder die für stärker schematisierte Aufgaben eingesetzt werden.

Auf die oben genannten Fähigkeiten wird um so mehr Wert gelegt, je höher die Qualitätsanforderungen und je länger die Lebensdauer der Systeme sind. Gerade bei langlebigen Systemen, die in vielen Versionen und Varianten existieren, können auch kleinere Unsauberkeiten langfristig zu aufwendigen Reparaturmaßnahmen führen.

4 Softwaretechnik als Lehrgebiet

Welche Konsequenzen sind nun aus den vorstehenden Beobachtungen zu ziehen – insb. hinsichtlich der Frage, wie Informatiker qualifiziert sein sollten, um in Konkurrenz zu Angehörigen anderer Berufsgruppen vorteilhaft abzuschneiden, und welchen Stoff eine Vorlesung über Softwaretechnik anbieten sollte?

Offensichtlich sind eine generelle und breite praktische Kompetenz sowie fundierte Kenntnisse der theoretischen bzw. mathematischen Grundlagen unerlässlich. Mit anderen Worten sollte ein Informatiker in allen wichtigen Fächern der Praktischen Informatik – insb. Betriebssysteme, Rechnernetze, Compiler, Datenbanksysteme, Graphik – zumindest die wichtigen Architekturen und Systemkomponenten incl. der zugehörigen theoretischen Grundlagen kennen¹⁰. Basis hierfür sind

¹⁰Diese Forderung nach einer breiten praktischen Kompetenz hat im Fachbe-

allgemeine Programmierkenntnisse und die Grundlagen der theoretischen Informatik. Die vorgenannten Fächer werden typischerweise in eigenen Veranstaltungen abgedeckt und brauchen daher in einer Vorlesung über Softwaretechnik nicht behandelt zu werden.

Zu ergänzen sind die Kenntnisse in den Einzeldisziplinen durch übergreifendes “Querschnittswissen”, worunter hier Themen verstanden werden, die die oben erwähnte systematische Vorgehensweise ermöglichen. Die hier vorliegende Vorlesung Softwaretechnik I konzentriert sich dabei auf die folgenden Themen:

- Systemanalyse und -modellierung
- Architekturen: Standardarchitekturen und Architekturbeschreibungssprachen
- Projektmanagement und Vorgehensmodelle

Eine sehr kritische Frage ist die nach dem Umfang des Anwendungsgebiets. Die angewandten Informatikstudiengänge – beispielsweise die Technische Informatik und die Wirtschaftsinformatik – beantworten die Frage dahingehend, daß man sich bereits während des Studiums auf ein bestimmtes Anwendungsgebiet festlegt und in diesem relativ umfangreiche Grundlagen erwirbt, ferner Standardarchitekturen oder sogar Produkte für diesen Anwendungsbereich kennenlernt. Im Vergleich zur klassischen Informatik müssen als Kompensation der Umfang und Detaillierungsgrad, mit dem Informatikthemen behandelt werden, reduziert werden. Generell wichtig sind daher Methoden, durch die sich Informatiker Wissen aus dem Anwendungsbereich der zu entwickelnden Softwaresysteme aneignen; derartige Methoden gehören insb. zum Gebiet der Systemanalyse und -modellierung, was die Wichtigkeit dieses Teilgebiets noch einmal unterstreicht.

reich 12 der Universität Siegen zur Bildung sogenannter Kernfächer geführt. Ein Kernfach ist eine 2+1-stündige Veranstaltung im Hauptstudium, die anwendungsorientiertes Grundwissen in einem der o.g. Fächer vermittelt.

Literatur

- [Ha95] Hartmann, M.: Informatiker in der Wirtschaft - Perspektiven eines Berufs; Springer; 1995
- [Lu99] Ludewig, J.: Softwaretechnik in Stuttgart - ein konstruktiver Informatikstudiengang; Informatik-Spektrum 22:1, p.57-62; 1999/02

Index

- Arbeitsmarkt
 - Analysen, 12
 - Nachfrageüberhang, 11
 - Spezialisierung, 13
- Architektur, 6, 7, 16
- Individualsoftware, 13
- Informatik
 - Angewandte, 16
 - Praktische, 4, 15
 - Technische, 16
- Informatiker
 - Arbeitsmarkt, 10
 - professionelle Zuständigkeiten, 10
 - Qualifikationsprofil, 14
- Ingenieur, 3
- Ingenieurwissenschaft, 3, 5
- Kosten, 8
- Projektmanagement, 16
- Projektplanung, 7
- Qualitätssicherung, 7
 - Einfluß der Ausbildung, 8
- software engineering*, 3
- Softwaretechnik, 3
 - Lehrgebiet, 5, 15
 - Studiengang, 4
 - Wissenschaftsgebiet, 4
- Standardsoftware, 13
- Standesorganisation, 4, 9
- Systemanalyse, 16
- Technische Informatik, 16
- Vorgehensmodell, 6, 16
- Werkzeug, 16
- Wirtschaftsinformatik, 16