

## **Zurück in die Zukunft?**

### **Zur Geschichte der Rahmen(lehr)pläne Informatik SekII in Berlin (West)**

Lothar Sack  
Dickhardtstr. 26  
12161 Berlin  
[LotharSack@t-online.de](mailto:LotharSack@t-online.de)

Helmut Witten  
Brandenburgische Str. 23  
10707 Berlin  
[helmut@witten-berlin.de](mailto:helmut@witten-berlin.de)

**Abstract:** Am 8.-10. Oktober 1984 fand an der FU Berlin die erste GI-Fachtagung zur Informatik in der Schule statt [AH84]. Nach 25 Jahren kehrt diese Tagung, für die sich inzwischen die Abkürzung INFOS eingebürgert hat, wieder nach Berlin zurück, das Motto lautet *Zukunft braucht Herkunft - 25 Jahre INFOS*. Dieser Leitspruch soll mit dem vorliegenden Beitrag an einem konkreten Beispiel aufgegriffen werden: der Geschichte der Rahmen(lehr)pläne Informatik SekII in Berlin (West). Beide Autoren haben über viele Jahre an der Entwicklung der Berliner Rahmenpläne mitgewirkt. Das ermöglicht einen exemplarischen Längsschnitt zur Entwicklung der Rahmenpläne in einem Bundesland.

### **1 Ein Blick 25 Jahre zurück**

Auf der GI-Tagung 1984 hielten Wilfried Koch und Lothar Sack ein Referat über Konzepte und Inhalte für einen Lehrplan Informatik in der gymnasialen Oberstufe [KS84]. Dieser Rahmenplan wurde zum Schuljahr 1984/85 in Kraft gesetzt und galt fast 10 Jahre [S85], bis er durch den Nachfolgeplan [S93] ersetzt wurde. Die ersten Gehversuche zum Informatikunterricht in Berlin lagen 1984 bereits 15 Jahre zurück, in den Jahren 1972 [S72] und 1977 [S77] waren Vorgänger-Rahmenpläne erschienen, wobei der 77er Plan auf der Grundlage eines Modellversuches an der Fritz-Karsen-Schule in Berlin-Neukölln erarbeitet worden war [S78]. Während im Plan von 1972 lediglich zwei Kurshalbjahre beschrieben wurden, waren es 1977 bereits vier Semester. Im Plan von 1985, der uns im ersten Teil dieses Artikels schwerpunktmäßig beschäftigen wird, sind drei Lernjahre Informatik für die Klassenstufen 11-13 beschrieben. Es finden sich nur Grundkurse in diesem Plan, Leistungskurse waren den späteren curricularen Entwicklungen vorbehalten.

Erfahrungen mit Unterricht und Schülerverhalten bei den frühen Rahmenplänen führten zu grundsätzlichen Überlegungen, die in den Rahmenplänen von 1984 [S85] und 1993 [S93] berücksichtigt wurden: Jeweils ein Drittel der Schüler setzte den Unterricht im folgenden Unterrichtsjahr nicht fort. Ursache hierfür war nicht etwa ein schwindendes Interesse an der Informatik, sondern die geringe Wertigkeit, die dem Fach Informatik im Fächerverbund des dritten Aufgabenfeldes zugebilligt wurde. Abgesehen von der Wahl als Abiturprüfungsfach war der Besuch von mehr als zwei Informatik-Kursen für viele Schüler unter abiturtaktischen Gesichtspunkten nur interessant, wenn sie überdurchschnittliche Noten erhielten. So konkurrierte das Fach Informatik etwa mit dem Fach Sport.

Das führte zu zwei Überlegungen: Erstens erschien es konsequent, nach jedem Unterrichtsjahr eine sinnvolle inhaltliche Abrundung anzustreben, so dass die Schüler auch bei nicht vollständigem Durchlauf durch die Kursfolge mit einem (einigermaßen) zutreffenden Bild von der Informatik und ihren Bezügen zur realen Welt diesen Unterricht verließen. Zweitens erschien es sinnvoll, die Schüler in unterschiedlichen Rollen zur Informatik zu sehen: als von Informatik-Anwendungen Betroffener, als Benutzer von Informatik-Systemen, als jemand, der Informatiksysteme analysiert und versteht, sowie als Konstrukteur solcher Systeme. In jedem Kurs sollten die verschiedenen Rollen angesprochen und bewusst gemacht werden, mit fortschreitendem Unterricht trat die Rolle des kompetenten Konstrukteurs stärker in den Vordergrund.

## **1.1 Anwendungen und Auswirkungen**

Das Konzept des anwendungsorientierten Informatikunterrichts wurde in Berlin in den 70er Jahren entwickelt (vgl. [GR76], [SZ78]):

In der Entwicklung der Curriculum-Reform wurde die traditionelle Konzeption des durch die jeweilige Fachsystematik strukturierten Unterrichts als nicht mehr vereinbar mit der Erziehungs- und Ausbildungsfunktion der allgemeinbildenden Schule erklärt. Die Zielsetzung, Erziehung als Ausstattung zur Bewältigung von Lebenssituationen zu begreifen, zog die Forderung nach situationsbezogener Curriculumentwicklung nach sich [...].

Für den Unterricht in den Naturwissenschaften, der Mathematik und der Informatik wurde an mehreren Stellen diese Diskussion aufgegriffen und ein gesellschaftsbezogener, an den Lebenssituationen der Schüler orientierter Unterricht konzipiert [...].

Der anwendungsorientierte Informatikunterricht fußt auf dieser Konzeption. Intendiert wird nicht die Herausbildung einer allgemeinen Methode zur Lösung von Problemen beliebigen Inhalts. Vielmehr ist der durch den Einsatz der Datenverarbeitung in unserer Gesellschaft betroffene Schüler zu berücksichtigen und zur Selbst- und Mitbestimmung zu befähigen ([SZ78], S. 43).

In dem Beitrag von Koch/Sack zur Infos'84 heißt es entsprechend:

Daneben müssen Schüler einer allgemeinbildenden Schule über heutige und zukünftige Anwendungen der Informationstechnologie in fast allen gesellschaftlichen Bereichen informiert sein; dies umso mehr als eines der wesentlichen Motive zur Einführung dieses

Fachs gerade die gesellschaftliche Virulenz der Rechneranwendungen war und ist. Diese Information schließt die Kenntnis der Veränderungen mit ein, welche der Einsatz der Informationstechnologie bewirkt hat und bewirken wird. Schließlich sollen die Möglichkeiten und Grenzen der Einflussnahme und Gestaltung beim Rechnereinsatz für Betroffene und aktiv Beteiligte aufgezeigt werden ([KS84], S. 135).

Im Rahmenplan [S85] werden diese Themen als *Anwendungen und Auswirkungen* bezeichnet, der dafür ausgewiesene Stundenanteil beträgt ca. 25%. Im ersten Unterrichtsjahr sollen die Lernenden einen fundierten Überblick über typische Anwendungen der Informatik erhalten, Auswirkungen auf bestehende Berufe und das Thema Datenschutz sollen behandelt werden ([S85], S. 11). Im zweiten Lernjahr soll dies vertieft und durch Informationen über typische Datenverarbeitungsberufe ergänzt werden ([S85], S. 26).

Der anwendungsorientierte Ansatz wurde in der Fachdidaktik schon lange kontrovers diskutiert. So schrieb Rüdiger Baumann in [B96]:

Die Lehrer naturwissenschaftlicher Fächer sind von ihrer Sozialisation, ihren Interessen und ihrer Fachausbildung her gar nicht in der Lage, historische oder ethisch-politische Themen im Unterricht zu behandeln. [...] Angesichts dieser Sachlage stellt sich die Frage, wieso die genannte Aufgabe nicht von den hierfür zuständigen Fächern wahrgenommen wird (S. 134).

Bei dieser pessimistischen Sicht der Dinge stellt sich allerdings die Frage, ob fachübergreifendes und fächerverbindendes Lernen, wie es auch von der KMK gefordert wird, mit dieser Haltung wirksam möglich ist. Die Wirklichkeit ist nicht nach Fächern sortiert. Wenn Fachlehrerinnen und -lehrer darauf beharren, nur ihre Fachsicht zu vermitteln, werden die Lernenden mit der in den Präambeln der Lehrpläne geforderten Bewältigung der schwierigsten Aufgabe allein gelassen, nämlich die verschiedenen Perspektiven zu integrieren und auf praktische Lebenssituationen zu beziehen. Darüber hinaus kann man mit guten Gründen bezweifeln, ob sich z. B. die Sozialkunde dieser Aufgabe schon aus zeitlichen Gründen überhaupt widmen kann. Auch müsste man dem Sozialkundelehrer in analoger Weise zugestehen, sich auf seine fehlende Sozialisation und Kompetenz in Fragen der Informatik zurückzuziehen.

## 1.2 Algorithmik

Die Algorithmik wurde vor 25 Jahren als „Hauptgegenstand und -methode der Informatik“ gesehen ([S85], S. 8). Damit stellte sich der Berliner Rahmenplan in die Tradition des algorithmenorientierten Unterrichts, der nicht zuletzt durch die GI-Empfehlungen *Zielsetzungen und Inhalte des Informatikunterrichts* aus dem Jahr 1976 zum *Mainstream* innerhalb der Bundesrepublik geworden war. Diese Gewichtung spiegelte sich auch im vorgesehenen Stundenvolumen für diese Thematik, mehr als 50% der Unterrichtszeit waren dafür vorgesehen.

Im ersten Unterrichtsjahr war die „Einführung in grundlegende Elemente der Algorithmisierungsmethodik an Hand kleiner (sog. stand alone-) Programme“ vorgesehen ([S85], S. 11). Dabei sollte in die Methode der strukturierten Programmierung eingeführt wer-

den, ein Programmier(sprachen)kurs wurde abgelehnt. „In jedem Fall sollen Aufgaben aus den Anwendungsbereichen gewählt werden, die für die Informatikanwendung repräsentativ sind. [...] Mathematische und physikalische Problemstellungen sind nur sparsam zu benutzen“ (ebd., S. 12).

Im zweiten Unterrichtsjahr stand einerseits der Prozedur- und Modulbegriff im Vordergrund, es sollten damit Methoden erlernt werden, „die geeignet sind, komplexere algorithmische Aufgaben zu bearbeiten“ ([S85], S. 26). Zum anderen sollte „der Schüler [...] eine Reihe von Standardalgorithmen kennen lernen, die in vielfältigen Anwendungen eine Rolle spielen“ (ebd.). Ein wichtiger Aspekt waren die Spezifizierung und die Validierung der Programme, Prozeduren und Moduln ([KS84], S. 136). Darüber hinaus sollte zur Vorbereitung des Projektunterrichts auch in den *Software-Life-Cycle* eingeführt werden ([S85], S. 33).

Problematisch war und blieb die Einführung in die Algorithmik anhand von sog. *Micky-Maus-Programmen*. Hier wurde mit dem Folgeplan [S93] Neuland betreten. Als Einstieg war nunmehr die Benutzung und Analyse eines dokumentierten Systems mittlerer Komplexität vorgesehen: „Durch diesen Einstieg wird schon frühzeitig das Denken in komplexen Zusammenhängen gefördert“ ([S93], S. 12). Dieser Ansatz wurde später von der Paderborner Schule der Informatik-Didaktik mit der *Dekonstruktion von Informatiksystemen als Unterrichtsmethode* weiter ausgebaut und in Richtung Objektorientierung vertieft [HMS99].

### 1.3 Rechnerorganisation und Rechnerbedienung

Der erste Informatik-Rahmenplan in Berlin, das *vorläufige Grundprogramm* [S72], ist typisch für den rechnerorientierten Ansatz (vgl. [GR76], [SZ78]). Entsprechend der seinerzeit zur Verfügung stehenden Rechentechnik stand die maschinennahe Programmierung im Vordergrund, angestrebtes Bildungsziel war die *Entmystifizierung des Elektronengehirns*. Bei Helmar Frank, einem früher Verfechter dieses Ansatzes, hieß die Informatik bezeichnenderweise noch „Rechnerkunde“. Er forderte die Kenntnis einer maschinennahen Sprache als notwendige Voraussetzung zum Erlernen einer problemorientierte Sprache ([SZ78], S. 41f.).

Der rechnerorientierte Ansatz wurde schon im folgenden Rahmenplan [S77] relativiert. Vor 25 Jahren wurde der Rechner in erster Linie als Werkzeug für den Unterricht gesehen:

Die Beschäftigung mit den Komponenten eines Rechnersystems muss sich jedoch stets den anwendungsbezogenen und algorithmischen Zielsetzungen des Unterrichts unterordnen. Leitfaden für den Unterricht sollte das im Umgang mit dem Rechner wachsende Bedürfnis des Schülers sein, auch Einblick in die Funktionsweise von Rechnersystemen zu erhalten ([S85, S. 9]).

In der Rechnerorganisation besteht die Gefahr, dass die Eigenschaften des konkreten Schulrechners zum Leitfaden des Unterrichts werden. Aus diesem Grund wird im neu erstellten Rahmenplan zwischen Rechnerorganisation als Unterrichtsgegenstand und den notwendigen Kenntnissen der Rechnerbedienung, deren Erwerb in den Unterricht zu integrieren ist, strikt unterschieden ([KS84], S.135).

Im Rahmenplan [S85] werden daher für die Rechnerbedienung keine eigenen Unterrichtszeiten ausgewiesen, für die Rechnerorganisation waren ca. 22% der zur Verfügung stehenden Zeit vorgesehen. Dieser Prozentsatz wurde im Folgeplan [S93] mit ca. 9% nochmals stark reduziert, eine ausführliche Behandlung der technischen Informatik war nur als Vertiefungsgebiet möglich.

#### **1.4 Software-Engineering und Projektunterricht**

Das wichtigste und in dieser Konsequenz neue Paradigma für den vor 25 Jahren vorgestellten Rahmenplan war die Verbindung von Projektunterricht mit Ideen aus dem seinerzeit noch jungen Fachgebiet Software Engineering (vgl. [JKS83]). Zur Frage, warum Projekte im Informatikunterricht durchgeführt werden sollten, schrieben Stefan Jänichen (heute Präsident der GI), Wilfried Koch und Gerd Schürmann aufgrund ihrer Erfahrungen in der Lehre an der TU Berlin:

Ohne Projekte ist die reale Beschränktheit der Ressourcen (z. B. Zeit, Qualität der Werkzeuge, Kenntnisstand der Beteiligten) bei der Herstellung eines Produktes nicht erkennbar. Auch noch so gute Planspiele können diese Erfahrung nicht ersetzen. [...] Durch ein Projekt können die unterschiedlichen Interessen vom Auftraggeber für ein Produkt, Hersteller eines Produktes, des Benutzers und des von einem Produkt Betroffenen deutlich gemacht werden. [...]

Weitere wichtige Erfahrungen werden Lehrer und Schüler aus einem guten Projektunterricht gewinnen. Hier ist besonders der Wert organisierter Teamarbeit zu nennen, aber auch fächerübergreifende Integrationsmöglichkeiten des Projektunterrichts u. ä.

Wir sind der Meinung, dass in keinem der klassischen Schulfächer die Chance so groß ist, dieses „Neuland“ entschlossen zu betreten, wie im Informatikunterricht. So ist heute im Informatikunterricht noch genügend Spielraum, Problemstellungen anderer Fächer in einem Projekt anzugeben. Werden wir die Chance nutzen oder aber die Informatik ebenfalls in das enge Korsett der Lehrpläne zwingen, aus dem die anderen Schulfächer nicht mehr herauskommen? ([JKS83], S. 26)

Wie wir weiter unten sehen werden, ist in den letzten 25 Jahren auch für die Informatik in Berlin das Korsett enger geworden. Doch zurück ins Jahr 1984: Welches Verständnis von Software-Engineering lag dem damaligen Rahmenplan zu Grunde? Es handelte sich um ein didaktisch vereinfachtes Wasserfallmodell:

Gegenstand des Unterrichts ist die phasenbezogene Vorgehensweise bei der Softwareerstellung:

- Problemanalyse,
- funktionelle Analyse,
- Entwurfsanalyse (Modularisierung),
- Modulprogrammierung und Modultest,

- Systemintegration,
- Beurteilung und Einsatz des Systems.

Jeder Arbeitsschritt zielt auf die Herstellung eines abschließenden Dokuments, das Ausgangspunkt für den nächsten Arbeitsschritt ist:

- Anforderungsdefinition als abschließendes Dokument der Problemanalyse,
- funktionelle Spezifikation als Ergebnis der funktionellen Analyse,
- Entwurfsspezifikation als Ergebnis der Entwurfsanalyse,
- Modulprogramme und Testprotokolle als Ergebnis von Modulprogrammierung und -test,
- einsatzfähiges und dokumentiertes System als Ergebnis der Systemintegration,
- dokumentiertes System mit gezielten Bewertungen und Änderungsvorschlägen als Ergebnis der Beurteilung und des Einsatzes des Systems ([KS84], S. 138).

Das Wasserfallmodell ist heutzutage sicher nicht mehr state of the art, die Nachteile dieses Vorgehensmodells sind in der Praxis vielfach deutlich worden. Aus diesem Grund hat die IT-Industrie inzwischen viele ergänzende bzw. alternative Vorgehensmodelle entwickelt (s. z. B. <http://de.wikipedia.org/wiki/Wasserfallmodell>).

Der Gedanke, ein Softwareprojekt als pädagogisches Projekt in der Schule durchzuführen, wurde von Wolfgang Ambros ganz grundsätzlich kritisiert:

Es ist geradezu absurd, unter Berufung auf das pädagogische Projekt der Arbeitsschule [...] einen Berufszweig zu nehmen, zu dessen Wesen knallharte Konkurrenz, Kalkulation und Leistungsausbeutung gehört. Soziale und selbstbestimmende [sic!] Lernkomponenten in der pädagogischen Schutzhütte und industrielle Effektivität in der Wettbewerbssituation sind unvereinbar wie Feuer und Wasser ([BZ92], S. 194).

Gleichwohl sieht auch Wolfgang Ambros eine Reihe guter Gründe, „softwaretechnologische Methoden“ in den Informatikunterricht einzubeziehen. Auch Projekte im Informatik-Unterricht erachtet er als sinnvoll, allerdings nicht als Simulation eines professionellen Software-Projekts. Es geht Ambros also eher um die Frage des *wie* als um die Frage des *ob* (ebd., S. 195f.).

In den nachfolgenden Berliner Informatik-Rahmenplänen, beginnend mit [S93], ist die Ausrichtung an dem Wasserfallmodell deutlich zurückgenommen, Software-Projekte und Software-Engineering sind aber weiterhin wichtige Methoden und Inhalte. Inzwischen wird auch diskutiert, Aspekte von eXtreme Programming (XP) – dem Gegenentwurf zum Wasserfallmodell (s. z. B. [http://de.wikipedia.org/wiki/Extreme\\_Programming](http://de.wikipedia.org/wiki/Extreme_Programming)) – als Methode im Informatikunterricht einzusetzen [L08].

Der Berliner Rahmenplan von 1985 sah Projekte nur für das dritte Lernjahr vor:

Allerdings ergibt sich erst für die fortgeschrittenen Schüler die Möglichkeit, Probleme der Anwendung, Algorithmik, Rechnerorganisation und Rechnerbedienung bei der Bearbeitung realitätsnaher Projekte integriert und in ihrer tatsächlich auftretenden Relevanz zu erfahren und dabei in heute typische Methoden zur Herstellung komplexer Software eingeführt zu werden ([KS84], S. 135).

Für die Organisation der Projektarbeit wurden mehrere Alternativen vorgesehen:

- ein ganzjähriges Projekt (diese Variante wird in [JKS83] empfohlen),
- zwei halbjährige Projekte,
- ein halbjähriges Projekt und im anderen Halbjahr ein selbstgewähltes Thema.

In jedem Fall war also ein mindestens halbjähriges Projekt verpflichtend.

Im Plan aus dem Jahr 1993 [S93] wurde die Möglichkeit eines ganzjährigen Projektes zurückgenommen. Es war nur noch für das dritte Semester ein halbjähriges Projekt vorgesehen, für das vierte Semester wurden alternativ vier mögliche Vertiefungsgebiete vorgeschlagen:

- Theoretische Informatik: Automaten
- Nicht-prozedurale Sprachkonzepte (logisch oder funktional)
- Datenbanken
- Computergrafik

Diese Vertiefungsgebiete (insbesondere die beiden letzten) konnten durchaus auch projektorientiert bearbeitet werden. Die Behandlung eines selbstgewählten Themas (z. B. Technische Informatik) war auf Antrag weiterhin möglich.

## 1.5 Zwischenbilanz

Vor 25 Jahren war der Berliner Rahmenplan Informatik für die SekII gekennzeichnet durch die Betonung des anwendungsorientierten Ansatzes sowie durch ein verpflichtendes Softwareprojekt. Vom zeitlichen Umfang her lag der Schwerpunkt auf der Vermittlung von Algorithmen und Datenstrukturen. Demgegenüber war die Fachsystematik der Kerninformatik (technische, theoretische und praktische Informatik) für den Rahmenplan nur von untergeordneter Bedeutung.

Das ist aus Sicht der Autoren auch gut so, weil eine stark ausgeprägte Orientierung an der Fachsystematik immer die Gefahr impliziert, die Inhalte des Schulfaches als verkleinertes Abbild der Hochschuldisziplin zu verstehen.

Didaktik soll in der Regel das aus den jeweiligen Fachwissenschaften herausfiltern, das von allgemeiner, existentieller Bedeutung für das Leben (Überleben und menschenwürdiges Leben) ist und daher als lehrnotwendig legitimiert gelten kann. (Wolfgang Hilligen 1991). Ein Missverständnis von Didaktik ist die so genannte „Abbilddidaktik“, nach der es nur darum gehe, die Ergebnisse der Fachwissenschaft auf welche Weise auch immer in die Schülerköpfe zu transportieren. Die Fachwissenschaften sind nur ein Bezugspunkt der Didaktik, andere sind die Gesellschaft und die Bedürfnisse des Schülers selbst (<http://de.wikipedia.org/wiki/Didaktik>).

Beispiele für die Abbilddidaktik finden sich in vielen Fächern. Hier soll nur die „Wissenschaftsorientierung“ in der Mathematik“ aus den 60er und 70er Jahren des letzten Jahrhunderts als prominentes Beispiel genannt werden. Wenn man Lehrbücher aus dieser Zeit in die Hand nimmt, erscheinen sie häufiger abstrakter (und abschreckender) als einführende Universitätsliteratur.

Aber auch in der Informatik gibt es vereinzelt Verfechter der Abbilddidaktik. So heißt es bei Christian Maurer:

Unabdingbare Voraussetzungen für die Abbildung wissenschaftlicher Erkenntnisse auf Schulunterricht sind realistische Einschätzungen der Machbarkeit im Kontext des Bildungs- und Erziehungsauftrages der Schule – im Klartext: scharf durchdachte didaktische Reduktionen wesentlicher fachlicher Inhalte. [...] Für den Informatikunterricht in der gymnasialen Oberstufe sind in erster Linie die Kerngebiete eines Grundstudiums relevant [M08].

Hiermit ist offenbar nicht nur die (selbstverständliche) Forderung gemeint, dass schulisches Lernen nicht in Widerspruch zum jeweiligen wissenschaftlichen Erkenntnisstand geraten darf und Begriffsbildungen so angebahnt werden müssen, dass sie kein Lernhindernis für die weiterführende Beschäftigung werden. Insofern ist eine wissenschaftlich solide Lehrerausbildung unverzichtbar. Beim zitierten Ansatz stellt sich allerdings die Frage, ob die Gleichsetzung von schulischem und universitärem Lernen eine optimale Wissenschaftspropädeutik darstellt. Zumindest ist nicht erkennbar, wie die oben erwähnten anderen Bezugspunkte (die Gesellschaft und die Bedürfnisse des Schülers) angemessen berücksichtigt werden.

## **2 Was danach geschah**

Im Jahr 2000 begannen Vorarbeiten für einen neuen Berliner Rahmenplan Informatik für die Sek II. Anlässe waren u. a. die geplante Einführung des Leistungskurses Informatik, neuere fachdidaktische Entwicklungen (insbes. die Bedeutung der Objektorientierung im Informatik-Unterricht) sowie die Auswertung der Erfahrungen mit dem bis dahin gültigen Rahmenplan [S93].

Zur Auswertung der Erfahrungen wurden in den Jahren 2001 und 2002 auf den Landestagungen der GI-Fachgruppe IBBB (Informatik-Bildung Berlin Brandenburg) zwei Workshops zu dem geplanten neuen Rahmenplan durchgeführt.

Im Schuljahr 2003/04 wurde von der Senatsschulverwaltung die Rahmenplankommission Informatik SekII eingesetzt, die bereits am 23. 5. 2004 einen Entwurf des neuen Rahmenplans veröffentlichen und zur Diskussion stellen konnte. Dieser Rahmenplan wurde nach Überarbeitung zum Schuljahr 2005/06 in Kraft gesetzt und erhielt dabei aus Gründen, die hier nicht weiter interessieren, den etwas ungewöhnlichen Namen „Curriculare Vorgaben“ [S05].

### **2.1 Curriculare Vorgaben Informatik**

Was war neu an den curricularen Vorgaben?



Eine wichtige Änderung, die allerdings nicht nur die Informatik betraf, war die Umstellung von den durch den Behaviorismus geprägten Lernzielen auf Kompetenzen, die dem konstruktivistischen Ansatz in der Lerntheorie verpflichtet sind. Kompetenzen werden verstanden als individuelle Disposition zur Lösung von Problemen in variablen Situationen (vgl. [WE01]), während Lernziele punktuelle Verhaltensänderungen in einzelnen Lehr- und Lernprozessen beschreiben. Wenn mitunter operationalisierte Lernziele mit „Kompetenzprofilen“ gleichgesetzt werden (s. [M08]), zeigt dies, dass dieser grundlegende Unterschied offenbar noch nicht überall verstanden wurde.

In den Curricularen Vorgaben wurden erstmals Eingangsvoraussetzungen für die Qualifikationsphase und Abschlusstandards für das Abitur in Informatik formuliert. Standards sind dabei für eine bestimmte Jahrgangsstufe konkretisierte Kompetenzen. Mit der Beschränkung auf Kerninhalte wurden den schulischen Fachkonferenzen vielfältige Möglichkeiten zur selbstgestalteten Schwerpunktbildung eingeräumt.

Veränderungen ergaben sich

- bei der Gestaltung des Einstiegsunterrichts; hier wurden neben dem bisher einzig möglichen Einstieg über die Dekonstruktion eines mäßig komplexen Softwaresystems alternative Zugänge ermöglicht (internetorientierter Zugang, Einstieg über eine Minisprache, [S05], S. 11) sowie
- durch die freie Wahl bei der Anordnung von Projektsemester und Vertiefungsgebiet im dritten Lernjahr. Hierdurch wurden z. B. Projekte mit Datenbanksystemen ermöglicht.

Auf diese Weise wurde den Schulen weitgehende Freiheit bei der Erarbeitung schuleigener Curricula ermöglicht (Einstiegswege, Vertiefungsgebiete, Projektthemen).

Leider galten die Curricularen Vorgaben nur für die zwei Schülerjahrgänge, die 2008 bzw. 2009 das Abitur abgelegt haben. In im Frühjahr 2004 beschlossen die Bundesländer Berlin, Brandenburg und Mecklenburg-Vorpommern, für alle Fächer der Sek II Kerncurricula für die Qualifikationsphase zu entwickeln. Damit waren die Curricularen Vorgaben der letzte eigenständige Berliner Rahmenplan Informatik SekII.

## **2.1 Länderübergreifendes Kerncurriculum Informatik**

Im September 2004 fiel der Startschuss für das Projekt eines 3-Länder-Kerncurriculums. Die Federführung für die einzelnen Curricula wurde auf die drei Bundesländer verteilt, für Informatik fiel sie auf Brandenburg. Aus diesem Grunde konnte Berlin nur einen Vertreter für die länderübergreifende Kommission entsenden.

In Log in berichtete Manfred Vollmost ausführlich über das Kerncurriculum [V05]. In Berlin gilt es als Teil des Rahmenlehrplans für die gymnasiale Oberstufe für alle Schülerinnen und Schüler, die seit dem Schuljahr 2006/07 den Informatikunterricht aufgenommen haben [S06].

Ein länderübergreifendes Kerncurriculum zu schaffen ist ein ambitioniertes Vorhaben, das grundsätzlich zu begrüßen ist, weil es der Zersplitterung des Bildungswesens in der Bundesrepublik Deutschland entgegenwirkt. Aber selbst wenn nur drei Bundesländer beteiligt sind, unterscheiden sie sich doch zum Teil gravierend in den organisatorischen Vorgaben für die gymnasiale Oberstufe, von unterschiedlichen (Unterrichts-)Traditionen einmal ganz abgesehen.

Für die Qualifikationsphase werden im Kerncurriculum sechs fachbezogene Kompetenzbereiche definiert:

Neben den Kompetenzen, die sich eher auf den Erwerb und die Anwendung von Inhalten beziehen (*Informatiksysteme verstehen – mit Informationen umgehen – Wechselwirkung zwischen Informatiksystemen, Menschen und Gesellschaft beurteilen*), gibt es Kompetenzen, die verstärkt prozessorientiert ausgerichtet sind (*Problemlösen – Kommunizieren und Kooperieren*) und die im Informatikunterricht eine besondere Ausprägung erfahren. Die Kompetenz des *informatischen Modellierens* umfasst sowohl inhalts- als auch prozessbezogene Aspekte ([S06], S. 10, Hervorhebungen durch die Autoren).

Für den Unterricht der Qualifikationsphase werden für die genannten Kompetenzbereiche sowohl Eingangsvoraussetzungen als auch die angestrebten abschlussorientierten Standards für Grund- und Leistungskursniveaus benannt. Danach werden fünf inhaltlich orientierte Themenfelder für den Unterricht beschrieben: Datenbanken – Rechner und Netze – Softwareentwicklung – Sprachen und Automaten – Informatik, Mensch und Gesellschaft.

Das Kerncurriculum lässt durch eine gewisse Offenheit alternative Realisierungen durch schulinterne Curricula zu, droht aber durch die separate Auflistung der Inhalte das eigentlich neue, nämlich den Kompetenzansatz aus den Augen zu verlieren. Ein weiteres Problem ist die Stofffülle. Der Anspruch, ein **Kerncurriculum** zu sein, das nur 60% der zur Verfügung stehenden Unterrichtszeit abdeckt, kann damit sicherlich nicht erfüllt werden.

Aus Berliner Sicht kamen mit dem Kerncurriculum folgende verpflichtende Inhalte neu hinzu: Datenbanken (bisher nur propädeutisch in der Einführungsphase bzw. als Vertiefungsgebiet), Sprache und Automaten für den Grundkurs sowie für den Leistungskurs zusätzlich Grammatiken, formale Sprachen und Turingmaschine oder Registermaschine.

Im Kerncurriculum werden keinerlei Kurssequenzen angegeben. Dies ist konsequent im Sinne eines kompetenzorientierten Ansatzes, der nur die zu erreichenden Abschlussstandards beschreibt, den Weg dahin aber offen lässt. Leider wurde dies von den Bildungsverwaltungen als mögliche Überforderung der Lehrenden angesehen, so dass in Berlin und Brandenburg im länderspezifischen Teil noch jeweils unterschiedliche Kursfolgen formuliert wurden. In Mecklenburg-Vorpommern gibt es auch für Informatik ein Zentralabitur, dort wird eine detailliertere Steuerung durch abiturbezogene Vorgaben und Beispielaufgaben realisiert.

Für Berlin setzen die je vier Grund- und Leistungskurse gleiche Schwerpunkte und haben gleiche Titel: Datenbanken und Softwareentwicklung I – Datenbanken und Softwareentwicklung II – Grundlagen der Informatik und Vertiefungsgebiet – Softwareprojekt. Die Leistungskurse sind eine inhaltliche Obermenge der Grundkurse. Diese Anordnung würde grundsätzlich die für viele Schulen attraktive Lösung bieten, den Leistungskurs durch Grundkurs und einen Ergänzungskurs zu realisieren. Hiervon kann jedoch aufgrund der Vorgaben durch den Schulsenat z. Zt. (leider) noch kein Gebrauch gemacht werden.

Die Beschreibung der Kurse der Qualifikationsphase beschränkt sich auf die jeweils zu behandelnden Inhalte. Durch die ausschließliche Betonung der Inhaltsbeschreibungen, die am ehesten bei der konkreten Unterrichtsplanung als Referenz zu Grunde gelegt werden, wird abermals die Kompetenzorientierung konterkariert, stattdessen werden eher die schon totgeglaubten Stoffverteilungspläne wiederbelebt. Offenbar wird dem kompetenzbasierten Ansatz nicht so recht getraut.

Die aufgegriffenen Inhalte stellen eine der Entwicklung der Informatik und ihrer Anwendungen entsprechende Aktualisierung dar. Allerdings müsste aus Sicht der Autoren der Umfang der auch im Grundkurs verpflichtenden Themen der theoretischen Informatik hinterfragt werden.

Positiv im Berliner Rahmenlehrplan ist aus Sicht der Autoren das Weiterbestehen des Projektsemesters zu vermerken. Das Softwareprojekt ist aber nicht Bestandteil des Kerncurriculums und in Brandenburg und Mecklenburg-Vorpommern leider auch kein verpflichtender Teil des Informatikunterrichts der Sek II.

Für den der Qualifikationsphase vorausgehenden Unterricht (also Basis- und Profilkurs), der im Kerncurriculums nicht beschrieben wird, werden für Berlin im wesentlichen die curricularen Vorgaben übernommen.

### **3 Fazit**

Was hat sich also im Kern in den letzten 25 Jahren geändert? Die schon positiv vermerkte inhaltliche Modernisierung im Rahmenlehrplan (Datenbanken, Objektorientierung) sollte eigentlich selbstverständlich sein. Die Änderung von Lernzielen zu Kompetenzen betrifft alle Fächer, nicht nur die Informatik. Allerdings erfolgte diese Umstellung, wie bereits erwähnt, nur halbherzig.

Die Idee der leistungs- und wissenshomogenen Lerngruppe scheint immer noch Grundlage bzw. Vorgabe der Lehrplankonstruktion zu sein:

Im 1. Halbjahr der Einführungsphase sind ggf. unterschiedliche Vorerfahrungen bei den Schülerinnen und Schülern durch geeignete Maßnahmen der Binnendifferenzierung zu berücksichtigen. [...] Damit soll sichergestellt werden, dass sie am Unterricht in der Qualifikationsphase erfolgreich teilnehmen können, ([S06], S.V)

Das Thema „Anwendungen und Auswirkungen“ findet sich – zumindest teilweise – im Kompetenzbereich „Informatik, Mensch und Gesellschaft“ wieder. Eine zeitgemäße Weiterentwicklung der Anwendungsorientierung stellt das Projekt „Informatik im Kontext (IniK)“ dar (vgl. <http://www.informatik-im-kontext.de/>).

Der Themenbereich Algorithmik wird inzwischen eingebettet in die objektorientierte Modellierung behandelt. Die schwierige fachdidaktische Frage, ob zuerst in die „kleine“ Algorithmik oder in die objektorientierte Modellierung eingeführt werden soll oder ob beides integriert behandelt werden sollte, wird durch den Rahmenlehrplan nicht vorentschieden.

An der Wertigkeit von Rechnerorganisation und Rechnerbedienung hat sich in den letzten 25 Jahren nicht allzuviel geändert, auch im Kerncurriculum hat die technische Informatik keinen herausgehobenen Stellenwert.

Das verpflichtende Softwareprojekt hat sich als Konstante der Berliner Rahmen(lehr)pläne erhalten, im zeitlichen Umfang hat es allerdings deutlich abgenommen. Durch die Stofffülle, die auch in der Informatik inzwischen erheblich zugenommen hat, ist das oben zitierte „Korsett“ des Rahmenlehrplans spürbar enger geworden.

Leider ist die Informatik auch in einem weiteren Punkt den traditionellen Fächern gefolgt: Die Schülerin, der Schüler wird immer noch als Objekt der Unterrichtsbemühungen des Lehrers gesehen, nicht als Subjekt ihres bzw. seines Lernprozesses. Die Frage „Wirst du noch unterrichtet oder lernst du schon?“ muss daher aus Schülersicht weitestgehend zu Ungunsten des selbstgesteuerten Lernens beantwortet werden.

## Literaturverzeichnis

- [A78] Arlt, Wolfgang (Hrsg.): EDV-Einsatz in Schule und Ausbildung. München: Oldenbourg, 1978.
- [AH84] Arlt, Wolfgang; Haefner, Klaus (Hrsg.): Informatik als Herausforderung an Schule und Ausbildung. GI-Fachtagung Berlin, Oktober 1984. Berlin, Heidelberg (Springer), 1984.
- [B96] Baumann, Rüdiger: Didaktik der Informatik. Stuttgart (Klett), 1996.
- [BZ92] Bosler, Ulrich; Ziebarth, Wolfgang (Hrsg.): Schul/Computer/Jahrbuch '93/94. Hannover (Metzler) und Stuttgart (Teubner), 1992.
- [GR76] Gruppe Informatik in Berliner Schulen (Beer/Koerber/Reker/Sack/Schulz): Stand der Curriculum-Entwicklung für den Informatik-Unterricht in Berlin. Berlin, 1976.
- [HMS99] Hampel, Thorsten; Magenheimer, Johannes; Schulte, Carsten: Dekonstruktion von Informatiksystemen als Unterrichtsmethode – Zugang zu objektorientierten Sichtweisen im Informatikunterricht. In: [S99], S. 149-164.
- [JKS83] Jänichen, Stefan; Koch, Wilfried; Schürmann, Gerd: Software Engineering und Lehrerbildung im Fach Informatik. In: LOG IN 3 (1983), Heft 2, S. 25-29.
- [KS84] Koch, Wilfried; Sack, Lothar: Konzepte und Inhalte für einen Lehrplan „Informatik“ in der gymnasialen Oberstufe. In: [AH84], S. 133-138.

- [L08] Leiser, Shenja: Adaption von Aspekten des Extreme Programming zur Steigerung der Schüleraktivität bei der Entwicklung eines Spiels mithilfe der Entwicklungsumgebung *Greenfoot*. Berlin (Staatsexamensarbeit), 2008.  
[http://bscw.schule.de/pub/bscw.cgi/d567998/Examensarbeit%20Hr.%20Leiser%20\(2008\)](http://bscw.schule.de/pub/bscw.cgi/d567998/Examensarbeit%20Hr.%20Leiser%20(2008)).
- [M08] Maurer, Christian: Lerninhalte und Kompetenzprofile im Informatikunterricht der gymnasialen Oberstufe. <http://wb.mi.fu-berlin.de/inf/mix/kompetenzen/index.html>
- [S72] Senator für Schulwesen: Vorläufiges Grundprogramm für das Fach Informatik. Berlin 1972.
- [S77] Senator für Schulwesen: Vorläufiger Rahmenplan für Unterricht und Erziehung in der Berliner Schule. Gymnasiale Oberstufe. Informatik. Berlin 1977.
- [S78] Sack, Lothar: Ein Modellversuch zum Informatikunterricht in der Sekundarstufe II. In: [A78], S. 56-64.
- [S85] Der Senator für Schulwesen, Berufsausbildung und Sport: Vorläufiger Rahmenplan für Unterricht und Erziehung in der Berliner Schule. Gymnasiale Oberstufe. Informatik. Berlin 1985.
- [S93] Senatsverwaltung für Schule, Berufsausbildung und Sport: Vorläufiger Rahmenplan für Unterricht und Erziehung in der Berliner Schule. Gymnasiale Oberstufe. Informatik. Berlin 1993.
- [S99] Schwill, Andreas (Hrsg.): Informatik und Schule – Fachspezifische und fachübergreifende didaktische Konzepte. Tagungsband der Infos'99, Berlin, Heidelberg (Springer), 1999.
- [S05] Senatsverwaltung für Bildung, Jugend und Sport Berlin: Curriculare Vorgaben für die gymnasiale Oberstufe – Informatik. Berlin 2005.  
[http://bscw.schule.de/bscw/bscw.cgi/d610953/Curriculare\\_Vorgaben\\_Informatik\\_SekII](http://bscw.schule.de/bscw/bscw.cgi/d610953/Curriculare_Vorgaben_Informatik_SekII)
- [S06] Senatsverwaltung für Bildung, Jugend und Sport Berlin: Rahmenlehrplan für die gymnasiale Oberstufe – Informatik. Berlin 2006. [http://www.berlin.de/imperia/md/content/sen-bildung/schulorganisation/lehrplaene/sek2\\_informatik.pdf](http://www.berlin.de/imperia/md/content/sen-bildung/schulorganisation/lehrplaene/sek2_informatik.pdf)
- [SZ78] Schulz-Zander, Renate: Analyse curricularer Ansätze für das Schulfach Informatik (in [A78], S. 40-49).
- [V05] Vollmost, Manfred: Ein Kerncurriculum Informatik – Zur Diskussion gestellt. In: LOG IN 135 (2005), S. 54-22.
- [WE01] Weinert, F. E. (2001). Vergleichende Leistungsmessung in Schulen – eine umstrittene Selbstverständlichkeit. In: F. E. Weinert (Hg.): Leistungsmessungen in Schulen. Weinheim/ Basel, S. 17-31.