education sciences

IMPACT FACTOR 2.6

CITESCORE 5.5

Article

Diverging Paths: How German University Curricula Differ from Computing Education Guidelines

Louis Kobras, Marcus Soll and Jörg Desel

Special Issue
Perspectives on Computer Science Education

Edited by
Dr. Mohammed Farghally

MDPI

https://doi.org/10.3390/educsci15121694

education sciences

Article

Diverging Paths: How German University Curricula Differ from Computing Education Guidelines

Louis Kobras [1,*], Marcus Soll [1,*] and Jörg Desel [2]

[1] Department of Computer Science, NORDAKADEMIE University of Applied Sciences, Köllner Chaussee 11, 25337 Elmshorn, Germany
[2] Fakultät für Mathematik und Informatik, FernUniversität in Hagen, Universitätsstraße 47, 58097 Hagen, Germany; joerg.desel@fernuni-hagen.de
* Correspondence: louis.kobras@nordakademie.de (L.K.); marcus.soll@nordakademie.de (M.S.)

Abstract

Study program recommendations are pivotal for the accreditation of study programs. In Germany, one of the most important recommendations used in the accreditation of computing curricula is published by the Gesellschaft der Informatik e.V. (GI), the largest German professional society of computer science. This work investigates the difference between reality at institutions and the GI recommendations. Systematically gathered syllabi of Northern Germany computing curricula have been coded both deductively and inductively according to Mayring's qualitative content analysis (QCA) method. The descriptions of 197 mandatory courses belonging to 13 program descriptions were analyzed. In addition to the 17 subject areas already described in the GI recommendations, four new subject areas have been identified that can be considered widespread. On the other hand, four subject areas from the GI recommendations could not be found as part of the mandatory curriculum. The study identified a notable divergence between current study programs and the current GI recommendations. However, as only mandatory syllabi were investigated, this study contains some blind spots with regard to electives and study specializations as well as with regard to a regional selection bias. Secondary findings concern the handling of learning outcomes in German syllabus descriptions and the GI recommendations themselves.

Keywords: computing education; computer science education; curriculum analysis; outcome-based education; undergraduate education in Germany

check for updates

Academic Editor: Mohammed Farghally

Received: 30 September 2025
Revised: 5 December 2025
Accepted: 7 December 2025
Published: 15 December 2025

Citation: Kobras, L., Soll, M., & Desel, J. (2025). Diverging Paths: How German University Curricula Differ from Computing Education Guidelines. *Education Sciences*, *15*(12), 1694. https://doi.org/10.3390/educsci15121694

1. Introduction

The so-called Bologna process (Allegre et al., 1998; Ministerial Conference in Bologna, 1999) is the culmination of a movement that started in the late 1990s, aiming to make higher education in Europe more structured and more comparable, which led to the development of the European Higher Education Area[1]. As part of this process, the European Credit Transfer System was created to be able to compare the workload of different learning units (called modules). Additionally, different degrees across different countries were all converted to bachelor's degrees (undergraduate program) and master's degrees (postgraduate program)[2]. The Bologna process also finds its roots in an attempt by the founding countries (cf. Allegre et al. (1998)) to foster (inter)national competitiveness and innovate their higher education systems, especially in competition with the US (Charlier & Croché, 2007).

Going further in the process of implementing those ideas, the *Berlin Communiqué* of 2003 first mentioned the usage of *competencies* and *learning outcomes*, which should be used to specify the different Bachelor/Master degrees (Ministerial Conference in Berlin, 2003, p. 4). While these terms had been around in educational research for a while, this is the first time they formally appeared in publications of the Bologna process, cementing them on the highest political levels in Europe.

While the introduction of competency-based education was mandated in a top-down process by policy in the EU, the process was different in the US. Both universities and courts objected to the mandatory introduction of a single form of education (Nodine, 2016, p. 7); thus, learning outcomes and competencies are just one didactical model of many. However, due to multiple factors—namely, online technologies for education, increased institutional acceptance (of outcome-based approaches), increased effects for direct assessment, political influence, and increased demand from students—there is a pressure for the wide adoption of competency-based education (Nodine, 2016). It is not clear how many students are enrolled in competency-based education programs in the US (Nodine, 2016, p. 9). Internationally, learning outcomes play an important role in many accreditation programs, e.g., ABET (Velázquez-Iturbide (2021, p. 1), Brahimi et al. (2016)), FIBAA (Foundation for International Business Administration Accreditation, 2023), ACBSP[3], and NCAAA (Brahimi et al., 2016), just to mention a few.

In Germany, recommendations for the outcome-based structure of computer science degrees are given by the German Computer Science Society (*Gesellschaft für Informatik, e.V.*, GI)[4]. The current version of the recommendations was published in 2016, and at the time of writing, the recommendations are in the process of revision. However, while the recommendations are developed using several diverse stakeholder perspectives—thus experiencing widespread acceptance as a baseline model—and oftentimes used as a frame of reference for accreditation and curriculum design, they do, in fact, have recommendation status and are not mandatory to implement. Thus, their adoption into study curricula is not ubiquitous. Nonetheless, to ensure coherence and comparability between programs—one of the chief goals of Bologna—clear and common guidelines are needed.

This work aims to measure how well the recommendations are currently integrated into curricula and where the current educational praxis is diverging from those recommendations. The way curricula are created has a huge impact on education. Annala et al. (2015) state that, besides other aspects, being involved with curricula means control of the content being taught (Annala et al., 2015, pp. 177–178) while also outlining the issue with differing nomenclature on the topic (Annala et al., 2015, pp. 172–173). However, they also see curricula as a negotiation between different stakeholders (internal and external) where student autonomy and decision-making are in danger of being overlooked (Annala et al., 2015, pp. 177–178). Since curricula play such an important role, the question arises of how curricula are constructed and how well they are aligned with the GI recommendations. For this, computing program descriptions of institutes in Northern Germany are compared against the current curricula recommendations of the GI. Primarily, the divergence is measured through degree of overlap and by identifying gaps in the recommendations. Secondarily, a light is shed on how program descriptions currently handle the concept of learning outcomes and what common problems of learning outcome formulations can be found in curricula.

2. Theoretical Background

According to Hlebowitsh's foreword to the 2013 version of the Tyler Rationale, the first notions of operationalizing education took root in the form of the seminal work of Ralph Tyler after a long-winded process that had started in the 1930s (Tyler & Hlebowitsh, 2013, p. vii), although Tyler himself had referred to work performed in the last thirty years (Tyler, 1949, p. 3), dating the start to the late 1910s or early 1920s. This work later culminated in the original Bloom's Taxonomy (Bloom et al., 1956), a tool to formalize teaching objectives in the form of learning outcomes. These outcomes follow a common sentence structure (subject-verb-object) where the verb–or process–denotes some observable action. The taxonomy assigns each learning outcome to one of six such *cognitive processes*[5]–*Knowledge*, *Comprehension*, *Application*, *Analysis*, *Synthesis*, and *Evaluation*–which are occasionally portrayed in a strict (pyramidical) hierarchy of complexity (see, e.g., Kennedy (2006, Figure 3.2)).

Having evolved over the years (as intended, see Anderson et al. (2001, pp. xxvii–xviii)), the group around Anderson et al. (2001) created the "revised taxonomy", which shall henceforth also be labeled as "AKT" (from "Anderson–Krathwohl Taxonomy", based on the main editors' names[6]). The revised taxonomy extends the original Bloom's taxonomy by another dimension, mapping learning outcomes not only against cognitive processes but also against different types of knowledge (*factual knowledge*, *conceptual knowledge*, *procedural knowledge*, and *metacognitive knowledge* (Anderson et al., 2001, Ch. 4)), while at the same time shifting the focus from the levels of cognitive processes themselves to the different processes that comprise a taxonomy level (among other things)[7]. The Bloom Taxonomy is still in use today in one form or another, either the original model from 1956 or the revised model from 2001[8].

Learning outcomes are not an idea unique to Bloom or Tyler, and unfortunately also not a precisely delineated concept. In the context of education (research), several related terms[9] appear in the literature. While discussing the different terms in minutiae—albeit a worthwhile effort that would warrant a publication on its own—would extend beyond the scope of this work, we will interpret them all on the shared foundation that they share the same notion of observable change in learner behavior that is also found in Tyler (1949). Since there is this ambiguity, we will use the following terms for this work: *content* describes any topic without regard to specific behaviors or cognitive competences, whereas a *learning outcome* describes content combined with student behavior. We shall thus use the term *learning outcome* as a stand-in for any type of outcome-based formulation. Further terms used include *curriculum* to mean the description of an entire study program, whereas *syllabus* denotes the formal description of a singular course or module that is part of a study program. The terms *course* and *module* will be used interchangeably.

In Germany, recommendations for the content—and, since 2005, learning outcomes—of Computer Science Bachelor and Master programs in higher education are traditionally given by the *Gesellschaft für Informatik e.V.* (GI, *Computer Science Society*). The structure of the model employed by the GI (Zukunft, 2016, Sec. 3.1–3.2) borrows from the AKT (see Figure 1). It forgoes the knowledge dimension, instead introducing a notion of complexity and contextualization[10]. Further dimensions[11] are defined in the model but not used. The GI model's *2a Transfer* denotes a group of cognitive processes that are related to AKT's *Apply* but rely on contextual information and feature a higher complexity than "plain" application. In the same vein, the GI model's *3a Evaluate* is seen as simply a more complex, context-sensitive variant of AKT's *Analyze*. At the same time, the GI model assumes that (revised Bloom's) *Remember* on its own has no relevance for one's own main subject and that (revised Bloom's) *Create* occurs only in capstone projects for bachelor students at the earliest; thus, the GI decided to skip both levels of cognitive process entirely. A mapping between the cognitive processes of AKT and the GI model is shown in Figure 1. Note that there is no corresponding preimage that maps onto the GI model's *2a Transfer* from AKT, and there is no image that AKT's *Remember* maps onto. Level *4 Create* is defined in the GI model only so the same scheme can be used for undergraduate and graduate recommendations.

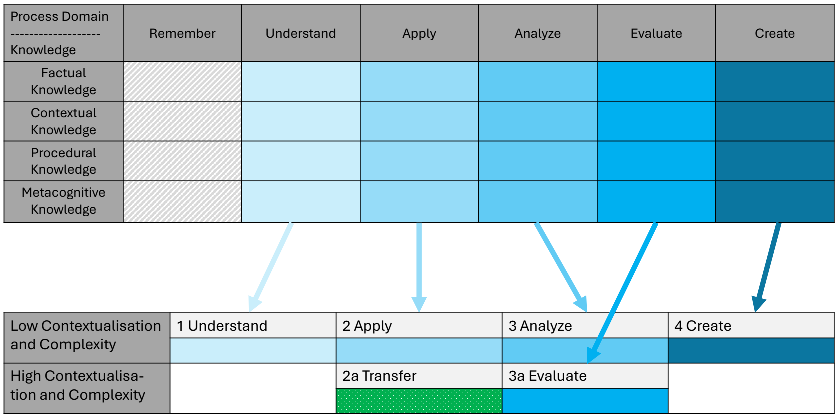

Figure 1. Mapping between the Anderson–Krathwohl Taxonomy (upper matrix) and the GI competence model (lower matrix). Note that the GI model ignores both the knowledge dimension and the cognitive process of *Remember*ing while at the same time introducing a dimension of context/complexity. Further, the GI defines a new cognitive process in *2a Transfer* that possesses no preimage in the Anderson–Krathwohl Taxonomy.

On the content level of the GI model, it was actively decided that it would not take influence from international standards like the ACM CS2013 curriculum[12] due to differences between the national educational systems (Desel, 2017, p. 236). However, as both the ACM and the GI model fundamentally describe the same discipline, notable overlaps on the content level are unavoidable. On a structural level, the CS2013 guidelines are defined to a much greater level of detail. Other differences appear in the grouping of content and focus areas. On a superficial comparison[13] between the 17 subject areas from the GI model and the 18 knowledge areas from the CS2013, the CS2013 seems to put more focus on programming and software development. The GI model, meanwhile, yields more space to (technical and theoretical) foundations of computing and mathematical skills. A final difference worth noting here is that the CS2013 differentiates between content and learning outcomes, a structure that can be found in syllabi in Germany but which the GI model does not really possess[14].

Most of the GI recommendations document focuses on the cognitive competences. Besides that, Zukunft (2016, Section 3.4) also mention some non-cognitive competences a student should gain. These are not formulated as learning outcomes and are not directly linked with any subject area but should be learned implicitly. They are also given substantially less space (around 1.5 pages compared to 25 pages of cognitive competences)[15].

These recommendations have a high importance in Germany, as the GI is (according to them) the largest professional society in Germany for computer science[16]. In the foreword of the 2005 recommendations, it is mentioned that the GI actually released the first formal recommendations for study programs in Germany (Gesellschaft für Informatik e.V., 2005, p. 5). Beginning with the recommendations of 2005, they are explicitly aligned with the Bologna process and are built based on learning outcomes (Gesellschaft für Informatik e.V., 2005, p. 6). However, a formal didactic model was only used in the latest iteration of 2016 (Zukunft, 2016, Section 3.1). It is unfortunately hard to determine how far these recommendations have been implemented since the recommendations are not mandatory and there is a lack of publications about study program design in Germany. However, based on personal experience from the authors, discussion with colleagues from different universities, and participation in Germany-wide working groups, the authors can confirm that the GI recommendations have a big influence on the design of German university curricula. This is strengthened by the fact that many stakeholders in universities, namely the "Fakultätentag Informatik"[17] and the

"Fachbereichstag Informatik"[18], were involved during the creation of the recommendations (Desel, 2017, p. 240). In addition, the recommendations for computer science are used as the basis for accreditation of bachelor's and master's program, e.g., see the foreword of the 2016 recommendations (Zukunft, 2016, p. 4), the ASIIN[19] as one of the (international) accreditation agencies (ASIIN e.V., 2018), or accreditation reports like Zentrale Evaluations- und Akkreditierungsagentur (2018, Section 3) and Agentur für Qualitätssicherung durch Akkreditierung von Studiengängen e. V. (2019).

For completeness, it should be mentioned that while the GI recommendations might be widespread and highly relevant in Germany, other recommendations[20] and international contracts (e.g., the *Seoul accord* (2008)[21]) exist. However, since the GI decided to not draw inspiration from the ACM/IEEE curricula for the GI recommendations in 2016 (Desel, 2017, p. 236), and neither the GI nor any accreditation agencies in Germany[22] are signatories of the Seoul Accord, their relevance is low when discussing the GI recommendations. In addition, accreditation agencies might have their own internal guidelines, which are not necessarily public.

3. Method

The latest iteration of the recommendations is from 2016 (Zukunft, 2016) (and thus nine years old at the time of writing), and a revision process of the recommendations commenced earlier this year. As an effort to inform this revision process, the purpose of this study is to identify the divergence between the GI recommendations and current computing study program curricula. To arrive at a response, three research questions (RQs) were derived as follows:

RQ1 Which learning outcomes defined in the guidelines of the GI can still be found in teaching?

RQ2 Which learning outcomes defined in the guidelines of the GI can no longer be found in teaching?

RQ3 Which new learning outcomes are actually taught in higher education but are not yet addressed in the GI recommendations?

To answer those questions, we need a criterion to decide whether a learning outcome is considered *widespread* (and thus relevant to the current discourse). We defined a learning outcome as widespread if it can be found in at least a quarter of the analyzed curricula. The study followed a classic literature survey approach (cf. Cooper (1988)) while applying qualitative content analysis (Mayring, 2022) to extract information from the gathered material. While the original purpose of the study was not to identify issues with curricula formulation, we were able to uncover several common problem areas in how German higher education institutes handle learning outcomes. These emerged during data gathering and analysis, and since we hope that this adds value for educators both nationally and internationally, we wanted to include them in the discussion as secondary findings.

3.1. Literature Survey

In this study, we aimed to take a neutral point of view to provide an objective overview of the content and skills that institutes of higher education promise to teach and thus (legally) bind themselves to. To investigate the educational landscape in Northern Germany, publicly available study program descriptions were downloaded from their respective program websites. Since there is no central register of all study programs in Germany, we decided to use the study finder of the German Federal Employment Agency (*Bundesagentur für Arbeit)*[23], where interested persons can search for study courses. While the search index still might not list all courses of study in Germany, it does feature a large database provided by federal offices and unbiased by financial incentives. The choice was made to analyze module descriptions because that is the publicly available information an institute of higher education chooses to present its program with. As such, that is the definite information about a program's contents and, in extreme cases, the factual basis for legal disputes between students and institutes (in conjunction with applicable law and quasi-legal documents like an examination regulation, in which the module descriptions are not uncommonly included).

The initial search queried any core computer science study programs (vis-à-vis specialized programs like computer engineering or business informatics or mixed programs like bioinformatics or computing education) of any type of higher education institution. The filtering was performed based on the title of the study program. In this case, modifiers like *International* were not grounds for exclusion. Similarly, it also did not differentiate between the German *Informatik* and the English *Informatics* or *Computer Science*; these were all included and interpreted as equivalent for the purpose of this study. It also included any organizational forms of study (full-time, part-time, minor degree, dual studies[24]). The availability of public program descriptions was determined as another exclusion criterion: If a program did not provide a link to the respective website in their entry in the Employment Agency's database and the program description could not be found on the university's website within a reasonable time frame, the program was excluded. The search was conducted at the end of January 2024.

A total of 372 study programs were identified. After applying exclusion criteria, the number of included programs was reduced to 145. Since analyzing these 145 programs was not viable, we decided to sample the study program based on the Federal States of Germany, similar to Pancratz and Grave (2023). We selected Northern Germany (Free and Hanseatic City of Hamburg, Free Hanseatic City of Bremen, Mecklenburg-Vorpommern[25], Schleswig-Holstein) for this study, resulting in 13 study programs (≈9%) from 11 institutions, which were analyzed in detail. To ensure that the investigation would only argue based on a canon of competences that could be reasonably assumed for any and every graduate of a degree to possess, only mandatory courses were investigated[26].

3.2. Qualitative Content Analysis (QCA)

The qualitative content analysis (QCA) as described by Mayring (2022) is a well-established systematic approach to qualitative studies in (German) social sciences[27]. We decided to use QCA since it allows easy analysis both qualitatively (Are there any problems with the way learning outcomes are formulated?) and quantitatively (which learning outcomes are widely included in curricula? Which are not?) while still following a rigorous scientific method. As a side note, this also allows us to measure how "flat" both the GI recommendations and the analyzed curricula are, i.e., how the coded learning outcomes are distributed over the AKT levels. Since the method is well-established and well-tested, we can assume a certain degree of validity.

The QCA as employed here combined deductive and inductive coding. As the goal of the study was to compare actual study descriptions that are currently in use against the GI's recommendations document, the coding scheme was designed deductively based directly on said recommendations document. Over the course of the analysis, the coding scheme was extended by new codes whenever some content could not be applied to existing codes. Investigated parts of the module descriptions included the fields *summary*, *learning outcomes*, *qualification goals*, *content description*, *relation to other courses*, and *comment* (and functionally equivalent fields with different labels) as far as they were available for a certain program's course descriptions. The study was coded by a single coder.

4. Primary (Quantitative) Findings

This study contained 13 curricula from 11 institutes in Northern Germany. Of all identified study programs, only mandatory courses were investigated to determine a common base that all graduates of said program are assumed to have. A total of 197 mandatory courses were analyzed, resulting in a total of 4923 coded text passages distributed over 950 unique codes. All curricula were analyzed based on their current status as of January 2024. The complete coding table will be provided as Supplementary Materials due to the size; a quick overview can be found in Table 1.

Figure 2 depicts the relative differences in competences that were the result of this study. For each of the defined GI subject areas, the amount of both the "removed" and "added" competences relative to the amount of competences already present in the subject area are given. Here, "removed" describes competences that are defined in the GI recommendations but could not be found in sufficiently many study programs, whereas "added" refers to competences that appeared in at least four study programs but were not part of the GI recommendations (and were thus inductively added to the coding scheme). The following noteworthy observations can be made:

1. Every subject area experienced removal of at least half of their defined competences;
2. No competences of the subject areas `VIII` (*Computer Science as a Discipline*), `X` (*IT Security*), `XI` (*Human Computer Interaction*), and `XIV` (*Project and Team Skills*) remained; all were removed in accordance with the study criteria;
3. Subject areas `XIII` (*Programming Languages and Methods*) and `XVI` (*Software Engineering*) experienced the most additions with about 90% and about 140%, respectively.

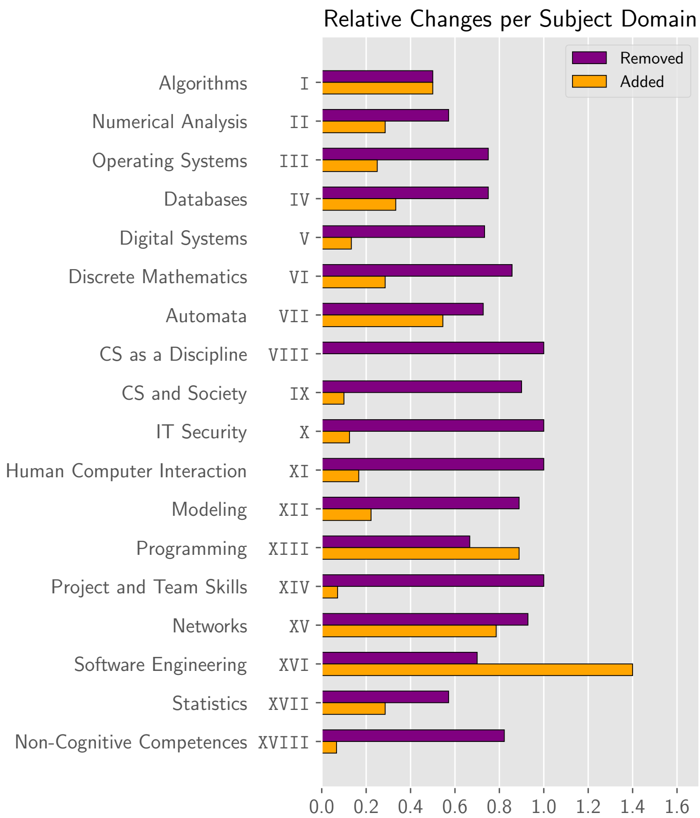

Figure 2. The relative differences for each of the GI subject areas. The flat purple bars denote the percentage of GI competencies that could not be empirically verified. The orange bars denote the new competences that could be identified for each subject area, relative to the number of competences the GI defines for the given subject area. Refer to Table 1 for non-shortened labels.

Table 1. Amount of codes found in curricula by different subject domains and competence levels. Here, "*No Level*" denotes a thematic hit that wasn't formulated as a competence as per the AKT and thus could not be assigned to any competence level. "*Transfer*" denotes an additional level used by the GI, which focuses on the application of skills within a professional context. The GI model used as a basis for the study encompasses the first 17 subject areas as well as a list of non-cognitive competences (denoted as subject area XVIII). Subject areas XIX–XXIV are areas that were identified during the study that are currently missing from the GI model. Subject areas ICVIII and ICIX contain topics of competences that were identified over the course of the study but could not clearly be assigned to any one subject area. Gray text color in a row indicates that no competence in this subject domain appeared in at least four curricula.

Number	Subject Domain	No Level	Know	Understand	Apply	Transfer	Analyze	Evaluate	Create	Total
I	Algorithms and Data Structures	241	43	58	36	23	35	19	10	465
II	Numerical Analysis	106	8	17	19	12	1	5	0	160
III	Operating Systems	151	4	41	14	3	4	9	10	236
IV	Databases and Information Systems	144	21	17	47	6	3	6	7	251
V	Digital Logic, Digital Systems, Computer Architecture	225	31	82	46	1	17	10	7	419
VI	Discrete Structures, Logic, Algebra	181	20	26	50	8	3	1	0	289
VII	Formal Languages and Automata	196	19	26	32	3	12	0	3	291
VIII	Computer Science as a Discipline	17	3	8	2	0	0	2	0	32
IX	Computer Science and Society	66	9	27	12	2	8	24	1	149
X	IT Security	52	15	3	6	2	0	2	3	83
XI	Human Computer Interaction	34	12	1	8	1	1	3	0	60
XII	Modeling	63	9	9	38	4	2	5	0	130
XIII	Programming Languages and Methods	305	57	46	120	10	14	6	8	566
XIV	Project and Team Skills	22	2	2	3	3	1	0	0	33
XV	Computer Networks and Distributed Systems	132	62	25	44	10	5	10	3	291
XVI	Software Engineering	209	43	28	76	5	8	12	9	390
XVII	Probability Theory and Statistics	118	11	8	21	6	5	3	0	172
XVIII	Non-Cognitive Competences †	408								408
XIX	Data Science	15	3	2	4	0	0	0	0	24
XX	Business Administration	53	21	30	21	0	4	6	2	137
XXI	Artificial Intelligence	20	6	2	4	1	0	3	0	36
XXII	Signal Processing	26	0	6	5	1	0	1	0	39
XXIII	Research Methods	12	7	3	10	1	3	5	3	44
XXIV	Robotics	3	0	0	2	0	0	0	0	5
ICVIII	Computer Science (General)	38	1	1	2	1	0	0	0	43
ICIX	Mathematics (General)	41	7	40	52	9	5	7	1	162
Total		2878	414	508	674	112	131	139	67	4923

† As the GI model is based on the AKT for cognitive competences, non-cognitive competences do not fit this scheme and by design cannot be associated with any competence level.

Figure 3 displays the total code frequencies per subject domain. The horizontal blue line marks the global average over all domains. The gray vertical line separates the subject domains defined in the GI recommendations from the subject domains identified over the course of the study. It is evident that several subject domains are less prominent in the mandatory courses, esp. `XIV` *Project and Team Skills* and `VIII` *Computer Science as a Discipline*. On the other hand, it can be seen that a notable amount of codes pertaining to `ICIX` *General Math* competences[28] (appearing in twelve distinct curricula) and `XX` *Business Administration*[29] (appearing in seven distinct curricula) can be found in current education. Also appearing in at least four curricula were the domains `XXIII` *Research Methods* (6 curricula) and `ICVIII` *General Computer Science*[30] (9 curricula).

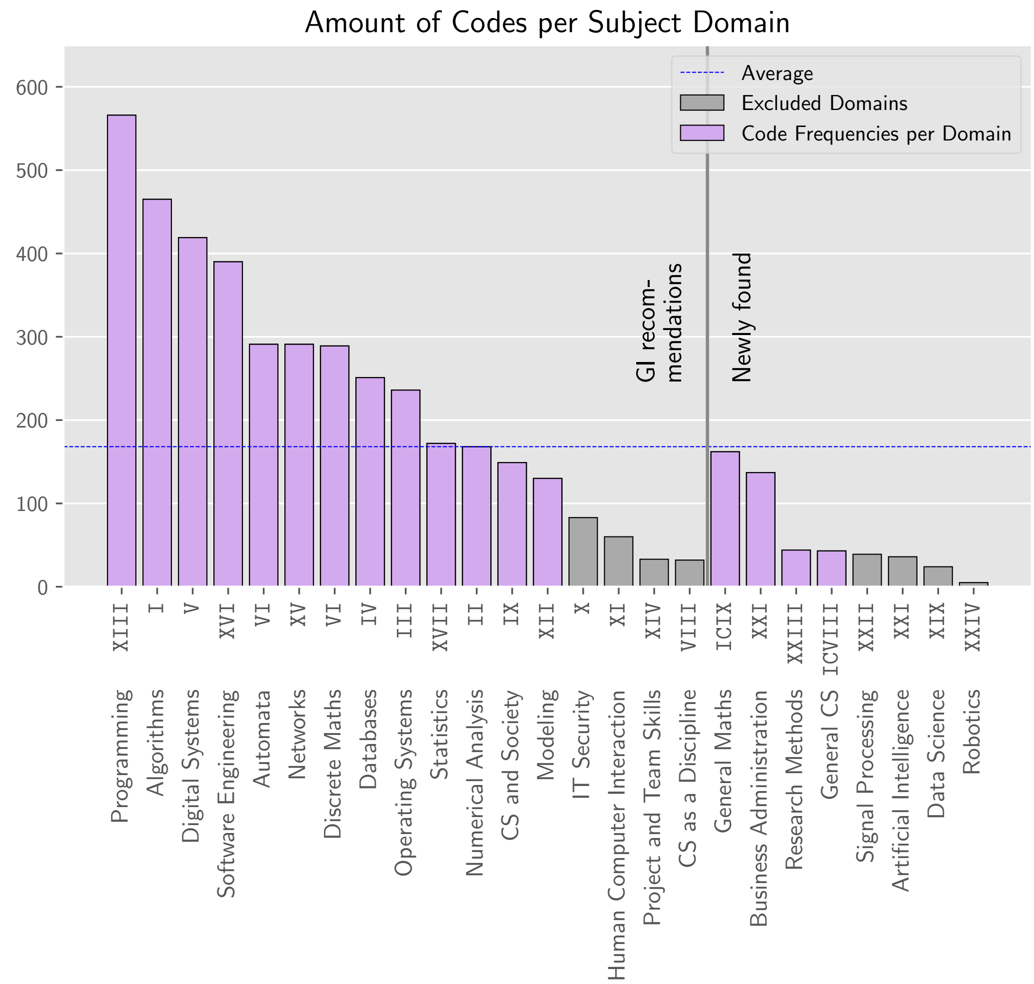

Figure 3. Number of coded learning outcomes per subject domain. The blue line marks the average number of coded learning outcomes over all subject domains. The vertical gray line separates the existing subject domains found in the GI recommendations from the new subject domains found only in curricula. The purple bars indicate subject domains that have been included (as per RQ1&3); the gray bars indicate subject domains with too few coded learning outcomes to be included (as per RQ2). Refer to Table 1 for non-shortened labels.

One question that came up during the study was about *"flatness"* of higher education, i.e., whether there was a tendency for higher education to focus on lower levels of Bloom's cognitive processes. To address this question, we used a linear regression to model the trend in both distributions (GI model and curricula); see Figure 4. For both distributions it can be seen that there is a notable drop in frequency between the *2 Apply* and *2a Transfer* levels[31].

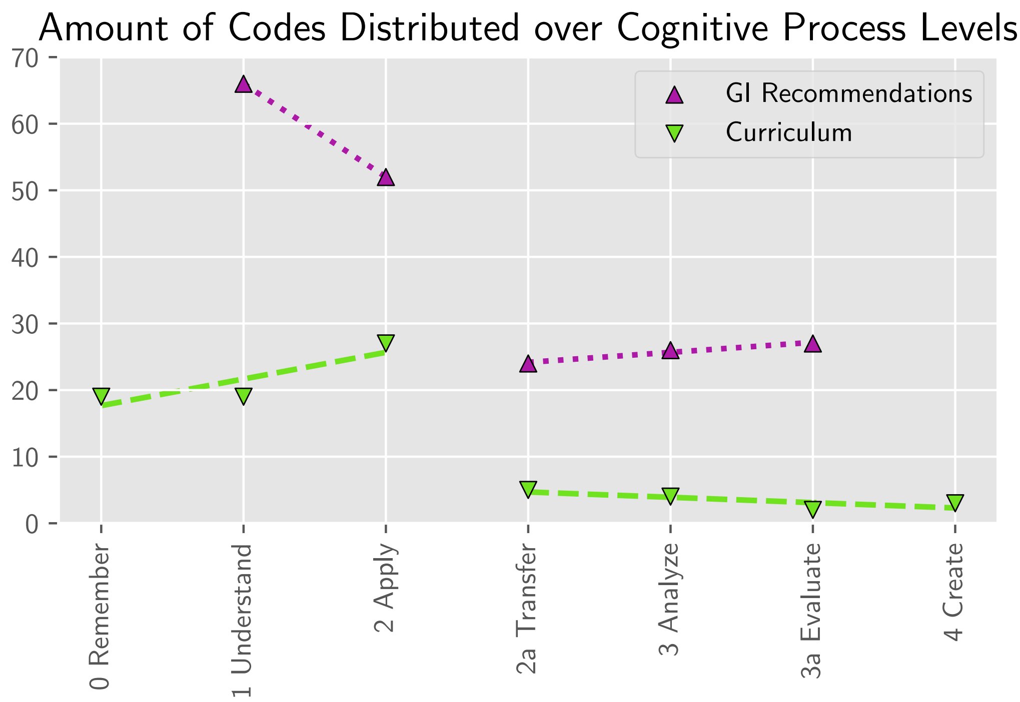

Figure 4. Number of coded passages in each level of the GI model (with added level 0). The purple upward triangles show the number of codes found in the GI recommendations. The green downward triangles show the number of codes found in the analyzed curricula. A linear regression (represented as lines) for both shows a great fit when the models are separated into one bin from *0 Remember* to *2 Apply* and one bin from *2a Transfer* to *4 Create*.

5. Secondary (Qualitative) Findings

While collecting the data, several secondary findings were discovered. These secondary findings mainly relate to how learning outcomes are formulated in the analyzed curricula. This section should not read as an attack on specific curricula or the persons writing them but should exemplify problems presumably with the way higher education is practiced in Germany—that is, that institutes do not work with established pedagogical models. While working without such models does not necessarily mean that the teaching is bad, having a standardized curriculum is helpful for the long-term planning and improvement of teaching (see Tyler (1949, p. 3)). In addition, while professors (who do most of the teaching in Germany) should have pedagogical qualifications, there is no hard requirement for formal training in pedagogy[32] or didactics.

To illustrate these findings, some examples from existing curricula will be presented. Each example will show the original German version as well as a best-effort translation performed by the authors of this paper. If the learning outcome is itself in English, only said English version will be given. The examples also feature a pointer to the module descriptions they were taken from, mentioning both the institute and the course number or description. Where several study programs from the same institute are investigated, the relevant study program is noted in parentheses following the institute[33]; the parentheses are omitted if only one program of that institute was investigated. In addition, we provide suggestions on how these examples could be improved by showing them in alternate wording.

5.1. No Learning Outcomes

There are multiple definitions for learning outcomes, but most focus on observable actions students should be able to perform (see Krathwohl (2002, p. 213), Biggs and Tang

(2011, Chap. 7), Anderson et al. (2001, pp. 12–14)). The GI scheme uses a similar notion that a learning outcome is a cognitive skill that allows students to act in a given situation (Zukunft, 2016, Chap. 3.1).

However, there were multiple descriptions of learning goals that do not adhere to this loose definition of learning outcomes (independent of which exact model you use for learning outcomes).

5.1.1. Example 1

> English original: *an introduction to [...] the programming of database applications is given.*
>
> — Christian-Albrechts-Universität zu Kiel (1-Fach)—infDB01a Database Systems

In this case, only the content of the course is given. There is no action a student can perform. Here, the item should be converted to a concrete (measurable) learning outcome consisting of a subject, a verb and an object. Multiple learning outcomes are most likely needed, some could be as follows:

> *Students can write simple `SELECT` statements containing a `FROM`/`WHERE` clause.*

> *Students can explain the difference between inner join, left join, right join, and outer join.*

> *Students can write a `JOIN` statement using the optimal type of join.*

> *Students can construct an optimized database view.*

5.1.2. Example 2

> German original: *Festgelegte Schnittstellen mit bekannter Spezifikation liefern einen Teil der Daten für das eingebettete System und das System gibt ebenfalls Daten über normierte Schnittstellen heraus.*
> English translation: *Interfaces with known specifications provide a part of the data for the embedded system and the system also sends data through standartized interfaces.*
>
> — HS Bremerhaven—Eingebettete Systeme

Here, the requirements for a student project are defined. One can only speculate that this should be understood as *students are able to specify and use interfaces*. However, based on the wording, it could also be interpreted that students simply are given a function to receive data and do not need deeper knowledge on interfaces. With a lack of action, it is impossible to tell what students should actually do. Therefore, it is impossible to give better examples here without knowing the intent of the instructor.

5.1.3. Example 3

> German original: *Simulation, Reduktion, Vollständigkeit*
> English translation: *Simulation, Reduction, Completeness*
>
> — Uni Lübeck—CS2000 Theoretische Informatik

Here, just three words are written. This is missing details on multiple levels; not only are the actual procedures (and thus details on the level) missing, but there is also no action associated with them (should students *Know*, *Understand*, *Apply* those methods?). Assuming these should be performed on the *Apply* level, learning outcomes could look like the following:

Students can use software programs to simulate the execution of different models.

Students can perform reduction to reduce a given problem to another given problem.

Students can prove the NP-completeness of a given problem.

5.2. Unclear Learning Outcomes

Sometimes, learning outcomes lack certain details which make it hard to impossible to interpret for students what the instructor actually wants them to learn.

5.2.1. Example 1

Coming back to one of the examples from the last section:

> English original: *an introduction to [...] the programming of database applications is given.*
>
> — Christian-Albrechts-Universität zu Kiel (1-Fach)—infDB01a Database Systems

Based on the formulation, one can interpret this as belonging to different levels of the GI model: *No Level* since no clear cognitive process is given; *Know* or *Understand* since students should be able to name/explain basic principles; or *Apply* if students should be able to actually build small applications after the introduction. For all of these levels, the learning outcomes should be formulated differently. Let us show this for one example[34] below:

Know: Students can state the syntax of a `SELECT` *statement.*

Understand: Students can explain a given simple `SELECT` *statement containing a* `FROM/WHERE` *clause.*

Apply: Students can write simple `SELECT` *statements containing a* `FROM/WHERE` *clause based on a textual description.*

5.2.2. Example 2

There are also some terms which can not be assigned to a level of the model, such as *lernen (to learn)*, *beherrschen (to master)* und *gehen um mit (being able to deal with)*. For example,

> German original: *fachliche Kompetenzen: Die Studierenden lernen, wie Daten in relationalen Datenbanken abgelegt und verarbeitet werden*
> English translation: *expertise: Students learn how data is saved and processed in relational databases*
>
> — HS Bremerhaven—Datenbanken 1

The term *lernen (learning)* here is not helpful. Based on content, this could mean *Remember* or *Understand*, but could go as high as *Create* if students should be able to construct a whole database system themselves.

> German original: *-Beherrschung elementarer Beweistechniken und Beweise selbst durchführen können.*
> English translation: *-Master fundamental methods of proof and are able to do proofs themselves.*
>
> — Uni Bremen (Hauptfach)—THI 1

While *beherrschen (to master)* might sound good—who does not want to master a subject area?—it is not possible to tell which level of mastering (e.g., *Understand*? *Apply*? *Create*? All levels at once?) students should actually show. The same problem exists if one uses *gehen um mit (being able to deal with)*:

> German original: *können mit Parametern, Transformationen und graphischen Darstellungen umgehen*
> English translation: *being able to deal with parameters, transformations, and graphical representations*
> — HAW—Analysis und lineare Algebra

For all of these, multiple learning outcomes can be created depending on the desired level of learning. For the first example in this subsection, learning outcomes for the different levels could look as follows:

> *Remember: Students can describe how data is saved in relational databases.*

> *Understand: Students can explain why data is saved in a given data type in a database.*

> *Apply: Students can perform a `SELECT` statement on a given database.*

> *Analyze: Students can compare different schemas for saving data in a database.*

> *Evaluate: Students can argue whether a given schema is appropriate for given data.*

> *Create: Students can create a schema for a set of given data.*

5.3. Lack of Context

For some learning outcomes, context required for understanding is missing.

5.3.1. Example 1

> German original: *Vorgehen bei der Analyse und beim Entwurf von umfangreichen Systemen*
> English translation: *Approach for the analysis and design of large systems.*
> — HSB—Softwaretechnik

Unfortunately, it is still unclear what exactly the object of the analysis is. For example, if certain quality criteria should be analyzed, this could be classified as `XVI` *Software Engineering*. If the structure and the behavior of the system should be analyzed, this would fit more into `XII` *Modeling*. Some examples of learning outcomes with added context could be as follows:

> *`XVI` Software Engineering: Students can analyze whether a system follows given code quality standards.*

> *`XII` Modeling: Students can analyze whether the behavior of a system fits the model of the system.*

5.3.2. Example 2

> German original: - *Methodenwissen für die Analyse von Anwendungskontexten und die Gestaltung von Informatiksystemen.*
> English translation: - *Method knowledge for the analysis of application contexts and the design of information systems*
> — Uni Hamburg—InfB-IKON Informatik im Kontext

For this example, it is unclear what process should be addressed by the learning outcomes. This could be focusing on the *Remember* level for simply knowing the processes. It could also be a combination of *Analyze* and *Create*—and in this case also be an example for the category of *procedural knowledge* (knowledge domain) of the revised Bloom's Taxonomy (Anderson et al., 2001) which is ignored in the GI model.

> *Remembering: Students can enumerate over different methods for the analysis of application contexts and the design of information systems.*

> *Procedural knowledge: Students can determine which method should be applied for a given analysis task of a given information system.*

Since the AKT actually defines learning outcomes as both having a related cognitive process and associated type of knowledge (Anderson et al., 2001, p. 27), it would be even better to combine them into one learning outcome:

> *Procedural Analysis: Students are able to select and execute a fitting method to analyze application contexts.*

5.3.3. Example 3

> German original: *entwerfen, implementieren und analysieren anhand von Anforderungen eigene Algorithmen,*
> English translation: *design, implement and analyze own algorithms based on requirements*
> — BHH—Algorithmen und Datenstrukturen

This learning outcome contains *Anforderungen (requirements)* as a word, but the context is missing on which kind of requirements should be analyzed (complexity? correctness? parallelism?). In addition, the requirements might pertain to the analysis part or to all three steps. Learning outcomes could look like the following:

> *Students can create own algorithms given a problem to solve and a target runtime in Landau-notation.*

> *Students implement own algorithms given run time constraints of a real (physical) system.*

> *Students can analyze whether a self-designed algorithm is correct given a formal specification.*

5.3.4. Example 4

> German original: *Simulation, Reduktion, Vollständigkeit*
> English translation: *Simulation, Reduction, Completeness*
> — Uni Lübeck—CS2000 Theoretische Informatik

Both the actual content (which methods are included) and the actions students should do (knowing? understanding? applying?) is missing. This might be one of the shortest items and the one with the least context. Since so much is missing, it is hard to describe concrete learning outcomes so we omit the rephrasing here.

5.3.5. Example 5

> German original: *Analyse der Betriebsmittelanforderungen in einer Mehrprozess-Umgebung*
> English translation: *Analysis of resource requirements in a multicore environment*
> — HS Bremen (Dualer Studiengang Informatik)—Betriebssysteme

Here, it is unclear what students should actually analyze. This could, for example, mean that students should analyze under which conditions a deadlock could occur. However, this could also mean that students should analyze which and how much resources are needed to execute a number of processes in parallel—-a task which is much harder than the first interpretation. Without a context, *analyze* here could mean a many different things. Therefore, different learning outcomes are possible here, for example,

> *Students can analyze why a deadlock occurred in a multicore environment.*

> *Students can analyze what conditions must be true in a given multicore environment for deadlock to occur.*

> *Students can analyze which (and how many) resources are needed to execute a given task.*

5.3.6. Example 6

For the last example, an item will be shown which can (with the exact same wording) be found in different modules:

> German original: *In Gruppen Probleme analysieren und gemeinsam Lösungsstrategien entwickeln und präsentieren können*
> English translation: *Work in a group to analyze problems and collaboratively develop and present solutions*[35]
> — Uni Bremen (Hauptfach)—Praktische Informatik 2, Technische Informatik 2, Informatik und Gesellschaft

This learning outcome can be found in multiple modules. For *Technische Informatik 2 (technical computer science 2)*, this outcome comes after multiple competencies from embedded systems and it seems that debugging in teams is addressed. For *Praktische Informatik 2 (practical computer science 2)*, this outcome follows different outcomes of algorithm design and functional programming and thus seems to focus more on the understanding and solving of formal problems. For *Informatik und Gesellschaft (Computer Science and Society)* this item is in a separate area called *General Studies* next to different non-cognitive learning outcomes and thus seems to contain different social skills. Combining all of these perspectives, this item seems to have different meaning based on the context it is placed in. Thus, the learning is that learning outcomes should be formulated in such a way that they stand on their own. In this sense, it would also make sense to separate the soft skills.

> *Technical computer science 2: Students find problems in a given hardware-software-system.*

> *Practical computer science 2: Students can rephrase a formal problem. Students can create a program that solves a formal problem.*

> *Computer Science and Society: [no special learning outcome needed here]*

> *Soft skills (separate section in all three modules): Students can work in a group.*

5.4. Separate Lists for Different Learning Outcomes

On the topic of non-cognitive competencies (see Note 14), the GI recommendations consider students to gain them mostly implicitly (Zukunft, 2016, p. 39). However, these are included in curricula in many different ways.

For example, math modules of the University of Lübeck explicitly mention *fachübergreifende Aspekte (interdisciplinary aspects)*, which are comparable to the non-cognitive competencies of the GI. Other modules do not contain such a section.

The HS Bremen has developed a catalog of *generischen Kompetenzen (general competencies)*, which contains non-cognitive competencies. Each module then has a variable in which it describes which of these general competencies are addressed by the module. For example, the module *5.12 Eingebettete Systeme (embedded systems)* addresses the general competencies *K2: Fähigkeit zu lernen (ability to learn)*, *K3: Lösung von Problemen (solving problems)* und *K4: Fähigkeit, theoretisches Wissen in Praxis umzusetzen (ability to translate theoretical knowledge into practical application)*.

In comparison, the University of Rostock uses a less structured approach. Although each module has variables *fachliche (technical)* competencies, *methodische (methodical)* competencies, *sozial/ethisch/rechtliche (social/ethical/legal)* competencies and *Selbst (self)* competencies, there is no defined catalog of competencies (in contrast to the HS Bremen).

5.5. Further Findings

There are some secondary findings which do not fit into one of the other categories and at the same time are not widespread enough to yield their own category.

For a positive example, the module *B040 Algorithmen und Datenstrukturen (algorithms and data structures)* at the FH Wedel explicitly mentions empirical measurements of complexity—-a learning outcome that is mentioned in the GI recommendation (item *I.3a.a*) but is often overlooked. The only other curriculum containing this learning outcome is from the Christian-Albrechts-Universität zu Kiel (1-Fach).

In the case of Uni Bremen (Hauptfach), the modules *Theoretische Informatik 1 und 2 (theoretical computer science 1 and 2)* have both different topics, but still list the exact same learning outcomes. This even leads to some questionable learning outcomes; for example, *theoretical computer science 2* deals with the topics *Berechenbarkeit und Komplexität (computability and complexity)* while *theoretical computer science 1* deals with *algorithms*. However, *theoretical computer science 2* addresses the learning outcome of *Korrektheit von Algorithmen beweisen und Eigenschaften von Algorithmen analysieren können (prove correctness of algorithms and analyze properties of algorithms)*—without any further mentions of algorithms in this module description.

In the HS Bremen (Dualer Studiengang), the authors seem to be using standardized texts which can be found in multiple modules without adding meaningful content. This has two effects: First, you obtain a high amount of redundancy. This is problematic since it means that formally, students activate the same competences in multiple modules, which might even pose a problem for (re)accreditation. Second, with learning outcomes not being formulated specifically for the module, pedagogical approaches like Constructive Alignment (see Biggs (1996)) are hard to apply.

> German original: *Sie kennen die Einfachheit und Eleganz der Konstruktion von Algorithmen durch das Weglassen von Programmvariablen, Zuweisungen und Schleifen.*
> English translation: *They know the simplicity and elegance of constructing algorithms by foregoing variables, assignments and loops.*
> — FH Wedel—B010 Grundlagen der Funktionalen Programmierung

Another example worth mentioning is from FH Wedel, where it seems that the author has some preconceptions which are coded into the learning outcome. In our opinion, curricula should be neutral to avoid both sounding like advertisements and also allow students to make their own judgments. A neutral formulation could be:

> *Students construct algorithms in the functional programming paradigm without using variables, assignments and loops.*

6. Discussion

In this section, the discussion will be focusing on the primary findings, the secondary findings, as well as the limitations of this study.

6.1. Discussion of Direct Coding Results (RQ1 and RQ2)

The coding results regarding RQs 1 and 2 yield two distinct perspectives: The incidence of subject domains, and evidence against assumptions made in the GI models.

6.1.1. Incidence of Subject Domains

From all competences defined in the GI model, 37 remained in total—from a starting amount of 194 pre-filtering. From these 37, 14 each are assigned to *1 Understand* and *2 Apply*, four each to *2a Transfer* and *3 Analyze*, and a singular one to *3a Evaluate*. This indicates a noticeable mismatch between the expectations that go out from the GI recommendations and what is currently put into mandatory curricula.

From the perspective of subject domains, `I` *Algorithms and Data Structures* observed the least loss both absolutely and relatively, with seven out of 14 learning outcomes remaining. The greatest absolute loss could be observed in `IX` *Computer Science and Society*, which lost 18 out of 20 learning outcomes; however, the greatest relative loss was observed in `VIII` *Computer Science as a Discipline*, `XIV` *Project and Team Skills*, `X` *IT Security*, and `XI` *Human Computer Interaction*, which share the same rate of loss: 100%. In other words, no four curricula from the sampled study programs could agree on a canon for these subject domains in any form (at least with respect to the mandatory syllabi). This yields weak evidence that even though these subject areas may be considered of some importance[36], they are ostensibly not considered part of the "core" of computer science by curriculum planners. Assuming this is not the case, it should be thoroughly investigated what is considered the current state of research and teaching in these topics. This can then be used to include this in the core recommendations to work against this apparent mismatch between theory (the recommendations) and practice (the actual curricula).

6.1.2. Checking for GI Model Assumptions

The GI model makes some assumptions about (revised) Bloom's levels of cognitive processes as follows:

1. Rote *Remember*ing (without *Understand*ing) has no relevance for one's own field of expertise;
2. *Creat*ing only happens in theses or large projects situated late in the study program, at least in the context of Bachelors' degrees;
3. *Evaluat*ion is only a context-specific special case of *Analys*is;
4. Non-cognitive competencies can not be formulated as learning outcomes.

These assumptions have all been invalidated by the data and based on the underlying AKT.

Empirical data gathered during the study shows that, at least on the syllabus level, knowledge of facts is very much a notable part of course contents. Notwithstanding how those learning outcomes may be implemented in a course[37], the frequencies displayed in

Figure 4 demonstrate that *0 Remember* is one of the most common cognitive processes in real-world syllabi. Whether this means that fact retention is actually relevant for current teaching practice or if this is simply an artifact of how syllabi are written was not part of this study.

The study was able to identify some learning outcomes on the *4 Create* level. Although they are few and far between, they do occur, and would presumably do even more so with a larger sample. In fact, Kiesler (2020) demonstrates that there is a plethora of *Creat*ing-level outcomes to be found in programming education alone. Arguably, both the GI understanding of Bachelor's students not being able to *Create* and the comparatively low frequency of relevant learning outcomes stem from the same misconception: that *Create* in the context of Bloom's means to create *something new*. This is discussed by Anderson et al. (2001) to some degree:

> *Create [. . .] also refers to objectives calling for production that all students can and will do. [. . .] in meeting these objectives, many students will create in the sense of producing their own synthesis [. . .] to form a new whole [. . .]*
>
> —Anderson et al. (2001, p. 85; text that was highlighted in italics in the source is underlined here.)

Thus, at least in the revised taxonomy (which the GI refers to), *Creat*ing refers to any process in which a problem is tackled by freely generating a solution space and then implementing a fitting solution[38].

The current structure with *Evaluate* being a subcategory of *Analyze* is problematic in the following two ways:

1. There is a clear distinction between the two levels in both the original *Bloom's Taxonomy* (e.g., see Bloom et al. (1956, p. 144, p. 185)[39]) as well as in the AKT (e.g., see Anderson et al. (2001, p. 68)). Any changes to the established theory should therefore be carefully explained. The GI gives as a reason that *Analyze* is a *sine qua non* for *Evaluate*[40], which is in immediate contradiction to the AKT scheme they use (see Anderson et al. (2001, pp. 267–268), Krathwohl (2002, p. 215)). The only other reason is the GI claiming that high complexity and contextualization are mandatory for evaluation (Zukunft, 2016, p. 68), a claim they give no real explanation for. Imagine an exercise like "Evaluate whether a given pseudo code (e.g., list all permutations of a list with 20 elements) can be run in reasonable time on a common PC.". This fulfills the definition of low context of the GI (level K2) but is clearly an evaluation task;
2. Many learning outcomes by the GI are wrongly sorted based on their wording. For example, in IT Security, the learning outcome "X.3.a Question properties and limitations of security concepts, combine different concepts in a sensible manner, and evaluate the security of complex systems." is sorted as *Analyze* while "X.3a.a Analyze situations in a company setting that pertain to IT security." is sorted as *Evaluate* (highlights made by the authors of the paper).

Thus, it is evident that the current form of the GI model is diverging from the underlying AKT with questionable reasoning and should experience a thorough rework—either in its structure or its justification.

Finally, non-cognitive outcomes are not formulated as learning outcomes. It should be noted that formulating non-cognitive outcomes as learning outcomes would have been feasible, as the original Bloom's Taxonomy had been planned around (Bloom et al., 1956, pp. 7–8) and extended to an Affective domain (Krathwohl et al., 1964) and a Psychomotor domain (e.g., Harrow (1972); Simpson (1966); which were written by different groups), providing the necessary tools to do so within the same (extended) framework as the AKT model. For example, the non-cognitive competence *reliability (Verbindlichkeit)* could be formulated as *Students desire to be reliable with regard to the promises they give.* (based on level 3.1 *Acceptance of a Value* of Krathwohl et al. (1964, pp. 140–149)).

6.2. New Learning Outcomes and Domains (RQ3)

In total, 37 new learning outcomes were identified that appeared in at least four distinct curricula. Of these, 19 are assigned *0 Remember* and 10 are assigned *2 Apply*. It thus seems that educators put some importance on students having some basic facts available—this might be being familiar with the relevant glossary terms for any given subject domain—and that a focus is put onto application of learned concepts, tools and methods (instead of a purely academic discussion). This is in line with an argument already made by Johnson and Fuller (2006) who claim that application is the core of computer science.

More interestingly, eight subject domains are identified which are completely new compared to the GI model. These are: `XIX` *Data Science*, `XX` *Business Administration*, `XXI` *Artificial Intelligence*, `XXII` *Signal Processing*, `XXIII` *Research Methods*, `XXIV` *Robotics*, `ICVIII` *General CS*, and `ICIX` *General Maths*. Of these, however, only `XX` *Business Administration*, `XXIII` *Research Methods*, `ICVIII` *General Computer Science*, and `ICIX` *General Maths* appear in at least four curricula. This might be indicative of a blind spot of the GI model: To function properly in an industrial position (that goes beyond being a *"code monkey"*), Computer Scientists need to know their way around company structures and how their field relates to business management. To function properly as actual scientists, a thorough training in general-purpose scientific methods and statistics is required. The outlier is found in `ICIX` *General Maths* which groups together aspects that cannot be clearly assigned to any single one Maths subject, such as proof techniques or formal notation. A similar phenomenon could be observed for `ICVIII` *General Computer Science* which included general-purpose skills like method and tool skills, general computer handling, and computational thinking. Note that `ICVIII` *General CS* and `XXIII` *Research Methods* appear in several curricula even though they have comparatively few codes in Figure 3. This indicates that the topics are widespread and relevant, but only include a comparatively narrow focus.

A special mention should go to `XX` *Business Administration*. This study explicitly only included study programs focusing on core computer science. Related study courses like *information systems* or German *Wirtschaftsinformatik*[41], where one would probably expect business-related learning outcomes, were explicitly excluded. It is surprising, then, that what is in practice a separate study degree in Germany which had decidedly not been part of the investigation had such a large influence on the conducted study still.

The findings concerning "new" subject domains raise a question about the identity of computer science. Keeping in mind that different interpretations of "Computer Science" or "Computing Science" exist between different countries, most computing professionals probably agree that programming is a core tenet of being a computing specialist. However, what would the consensus be regarding robotics or AI? How about economics/business administration? What about research methods and the all-time dreaded question about how much "science" can actually be found in "computer science" (e.g., see Tedre (2011))? The study shows that, even though these subjects are not (currently) part of the core computing recommendations (at least for Germany), they appear often enough in actual (German) computing curricula to be of note[42]. This is another call to the global computing community to reach consensus about our tree of subject families.

6.3. A Note on the Quality of Learning Outcomes (Secondary Findings)

Based on the secondary findings, we can see that writing good learning outcomes is difficult and that there are many errors that can be made writing learning outcomes. While being a good teacher is one thing, writing good learning outcomes is another skill that also must be learned. Based on our secondary findings, we can conclude that work on curricula could profit from more didactical guidance in order to write better learning outcomes across the board.

Another observation made during the curricula analysis is that the quality of learning outcomes has a high variance even between modules within the same curriculum. The easiest solution would be to let professors talk with each other about how they formulate their module descriptions. Other solutions might include improved quality assurance (e.g., peer review of module descriptions) and training of personnel.

This is unfortunate since good learning outcomes can not only communicate to students what is expected of them but also help structure teaching. Evidence suggests that presenting well-crafted learning has a strong positive effect on the actual learning of students; see Feldman (1989, pp. 604–605), Marzano (1998, p. 93), or Auer (2022). As a community, we should strive to improve the quality of written learning outcomes, be it through improved quality control or by helping educators write better learning outcomes. This also shows that while Bologna might be implemented formally, the implementation still is not running smoothly in all areas.

6.4. A Note on the GI Recommendations

While working on the coding, some areas were found where the GI recommendations themselves could be improved. These areas should be documented here in the hopes that they might prove helpful for other recommendations. For example,

> English original: *understanding of the tasks, structure and functionality of an operating system*
> — CAU—1-Fach–infOS-01a Operating Systems

Here, the same learning outcome can be classified as *III.1.a Explain fundamental/theoretical concepts of operating systems* or *III.1.b Explain the general structure of an operating system* (in this case, the learning outcome was coded by both). Something similar can be found for

> German original: *können Softwaresysteme auf verschiedenen Abstraktionsebenen beschreiben*
> English translation: *can describe software systems on different abstraction layers*
> — Uni Lübeck—CS2300-KP06 Software Engineering

Here, the same learning outcome could be interpreted as `XVI` *Software Engineering* or as `XII` *Modellierung*, depending on which context you assume.

There are some domains where this ambiguity is quite obvious. For example, learning outcomes regarding management and quality assurance of software products feature notable overlap between `XVI` *Software Engineering* and `XIV` *Project and Team Skills*. In fact, you could actually eliminate `XIV` *Project and Team Skills* by moving and merging the learning outcomes with `XIII` *Programming Languages and Methods*, `XVI` *Software Engineering*, as well as some non-cognitive competences.

For both future versions of the GI recommendations as well as other recommendations, attention should be paid to all categories being distinct and unambiguous.

6.5. Limitations

Four different limitations were identified in the context of this study: a blind spot concerning the analyzed data, a selection bias during data gathering, data density, and that there was only a single coder.

6.5.1. Blind Spot: Electives and Study Specializations

To identify those and only those competencies *any* graduate of a given study program could reasonably be expected to possess, analysis was confined to only mandatory parts of the curriculum. As mentioned before, this was performed to ensure that only modules that any and every graduate of a degree has taken are included. This, by design, excluded elective courses and study specializations, but also course catalogs where students could choose one course from several offers[43]. In cases where all these offered variants contain a common subset of learning outcomes, like it is often found in seminars and practical-oriented election modules (like projects), one could argue that this subset should be treated like a required module for the purpose of this study.

The exclusion of elective courses and study specializations therefore might lead to a relative under-representation of certain subject domains that are outside the "core" of computer science. However, the insight into which subject domains are considered "outside the core" is in itself relevant, as it shows where the discipline currently sees its main focus. Even where computing education might diverge from social perception of relevance—e.g., *IT Security* was only rarely part of the mandatory curriculum; however, cyberattacks happen increasingly often and regularly make prime-time news.

6.5.2. Regional Selection Bias

For this study, only a limited number of curricula were analyzed, namely from Northern Germany. The analysis of these already took a substantial amount of effort and time, and including more was not feasible for the study.

There are multiple sampling strategies. Choosing such a regional sampling strategy means that cultural biases (or other influences) of one region could influence all selected curricula and thus add a systematic bias to our analysis. This could be prevented by sampling the curricula based on quota (here, a fixed number of samples from each region in Germany). However, such a (region-based) approach of sampling has been used in related work, for example, by Pancratz and Grave (2023); thus, we assume it as a valid means of selection.

6.5.3. Bias Caused by Syllabus Descriptions

As was stated in Section 3.2, several different sections of module descriptions were analyzed. These included both the stated learning outcomes and the topic list. We found that often module descriptions in Germany give both a list of topics of a course and the intended learning outcomes; however, the topic list is usually given as a simple collection of keywords or a list of (textbook) chapters. This in turn influenced how many codes of *No Level* in relation to the total amount of codes were found in the curricula, causing a possible over-representation[44]. However, since we were interested in both the competence level as well as the (pure) content of the curricula[45], we decided to include all mentioned sections.

6.5.4. Data Density

The amount of curricula analyzed in this study can still be considered low, especially taking into consideration the large numbers of codes. A total of 950 codes were used, which means that with 13 curricula (≈9% of study programs in Germany), many codes have been found only a few times in the analyzed curricula. Therefore, there is a risk that both some blind spots still exist and some of the codes are overrepresented and would not have that much of an impact if more curricula were to be analyzed.

Please keep in mind that although the data density as well as the number of curricula is low, there is still a lot of work put together. The analysis of the 13 curricula consists of around 490 pages of module descriptions and 4923 coded text passages.

6.5.5. Single Coder

The coding itself in this study was performed by a single coder over a period of around five months. Although some passages that were hard to interpret were discussed with other persons, this bears the risk of adding the biases of the coder to this study. Please note that the QCA used here does not require multiple coders (see Section 3.2). These biases could be mitigated (but not eliminated) by using multiple coders and comparing their results (e.g., calculating inter-rater reliability); however, this would multiply the resources required for this study, which was not feasible for us.

Adding to the problem is the fact that both the GI model and the revised Bloom's taxonomy make it hard to clearly distinguish different levels of cognitive competences—a fact that is not only noted by Anderson et al. (2001, see p. 34)[46], but is also discussed by Velázquez-Iturbide (2021, pp. 4–5). Using multiple coders here could have reduced the risk of wrong classification due to including multiple perspectives.

7. Conclusions

Formalization of learning outcomes has been around for a while now (cf. (Bloom et al., 1956; Tyler, 1949)) and is coded into EU policy since 2003 (cf. (Ministerial Conference in Berlin, 2003)). For German computer science curricula, the German *Gesellschaft für Informatik e.V. (GI)* (Society for Informatics) has produced a catalog of recommended competences a bachelor graduate should achieve, separated into 17 different subject areas (Zukunft, 2016). This study took these GI recommendations and compared them against the syllabi of mandatory courses of 13 computer science study programs throughout Northern Germany using qualitative content analysis (Mayring, 2022).

In general, we can see some diverging paths between what the GI recommendations contain and what is actually taught: A vast difference was identified concerning several subject areas from the recommendations that are not regularly part of the mandatory curriculum (namely `X` *IT Security*, `XI` *Human Computer Interaction*, `VIII` *Computer Science as a Discipline*, and `XIV` *Project and Team Skills*). On the other hand, four subject areas have been identified that are relevant to current teaching discourse but lacking any formalization in the GI recommendations: `XX` *Business Administration*, `XXIII` *Reseach Methods*, `ICVIII` *General Computer Science*, and `ICIX` *General Maths*.

Furthermore, secondary findings show that the quality of learning outcomes in curricula (which are demanded by the Bologna process; cf. Ministerial Conference in Berlin (2003, p. 4)) is not up to the standard we hoped for. This should be addressed both on an institutional level (e.g., improved quality assurance, training of professors and other personnel) and on a national level (e.g., helpful guidelines, grants for improving curricula) to ensure both a high quality of teaching and an adherence to the Bologna process.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/educsci15121694/s1, Table S1: Competences remaining in the GI catalog as per RQ1; Table S2: Competences from the GI model which were identified in less than 4 module catalogs and are thus not seen as relevant; Table S3: Newly identified competences which were found in at least four distinct curricula; Table S4: Newly identified subject areas.

Author Contributions: Conceptualization, L.K., M.S. and J.D.; methodology, L.K., M.S. and J.D.; formal analysis, L.K.; investigation, L.K.; data curation, L.K.; writing—original draft preparation, L.K. and M.S.; writing—review and editing, L.K., M.S. and J.D.; visualization, L.K. and M.S.; supervision, J.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the *Stiftung Innovation in der Hochschullehre*, Germany, as part of project *Flexibel kombinierbare Cross-Reality Labore in der Hochschullehre: zukunftsfähige Kompetenzentwicklung für ein Lernen und Arbeiten 4.0* (CrossLab) (Aubel et al., 2022).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The full coding table is included in the Supplementary Material. Further inquiries can be directed to the corresponding authors.

Conflicts of Interest: The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

AKT	Anderson–Krathwohl Taxonomy
GI	Gesellschaft für Informatik, e.V. (German Computer Science Society)
RQ	Research Question
QCA	Qualitative Content Analysis

Notes

1 See https://ehea.info/, accessed on 10 November 2025.

2 See Ministerial Conference in Bologna (1999) for the original declaration.

3 https://acbsp.org/page/outcomes_assessment, accessed on 11 November 2025.

4 See https://gi.de/, accessed on 10 November 2025.

5 Originally, the Bloom working group postulated three different taxonomies with different areas (or domains) of application: The Cognitive domain, the Affective domain, and the Psychomotor domain. In current discourse, only the Cognitive domain receives much attention; while a definition for the Affective domain exists, most efforts for formal education focus on cognitive outcomes. The Psychomotor domain hadn't even been defined by the original working group, although works from different authors that do define it exist (see Anderson et al. (2001, p. xxvii)).

6 In addition, the name might lean on the two notable works that present and describe the revision to the scientific and educational communities: Anderson et al. (2001) and Krathwohl (2002).

7 Sometimes, the revised model is used, but the newer *knowledge domain* is ignored, leading to learning outcomes whose wordings align with the process domain of the AKT (rather than the original Bloom taxonomy), e.g., by using *Create* instead of *Synthesis*, but are not sorted in the second dimension.

8 Calling back to the notion of hierarchy, most readers are probably familiar with the aforementioned "Bloom Pyramid". This pyramid form cannot actually be found in the primary works on the taxonomy. Furthermore, for the revised model, the notion of the pyramid is indeed factually wrong since *"[...] [in the revision] the requirement of a strict hierarchy has been relaxed to allow the categories to overlap one another."* (Krathwohl, 2002, p. 215) .

9 Some examples are learning outcome, learning objective, competence, competency, mastery, outcome-based education vs. outcome-based teaching and learning, and outcome-based education vs. competency-based education. These terms are sometimes used interchangeably, although they appear to have slightly different meanings. For more discussion, see Hartel and Foegeding (2004), Harden (2002), Holmes et al. (2021); see also Nodine (2016, p. 10). Other examples may be *content*, *curriculum* and *syllabus*, which all denote some form of course or program description; *course* and *program*, which are both used to label an entire degree; or *course* and *module*, which both can be used to label a(n atomic) unit (or group of units) of learning. The label *course* is especially egregious: a *course of study* vs. a *course on programming*.

10 The GI model actually defines five levels of complexity and contextualization (Zukunft, 2016, p. 10): *K1* no contextualization; *K2* small examples; *K3* more complex examples; *K4* internal projects; and *K5* company projects. However, in the actual competence matrix, the first two levels and the second two levels are combined, respectively, into the rows for "low" and "high" contextualization and complexity. The fifth level receives no further attention in the GI model.

11 In addition to *K* for complexity (*Komplexität*), the recommendations also define *W* for the type of knowledge according to Anderson et al. (2001) (*Wissen*) and *T* for the type of scientific work. Notably, T is not independent of the dimensions of cognitive processes, complexity (K), and type of knowledge (W) in the GI model.

12 The Joint Task Force on Computing Curricula Association for Computing Machinery (ACM) IEEE Computer Society (2013), which had been the current version during the time in which the current GI model had been designed.

13 An in-depth comparison would be a research project in itself and shall thus be omitted here.

14 There is a content section in the GI recommendation in Zukunft (2016, Chap. 2), but it only relates loosely to the actual learning outcomes.

15 The non-cognitive competences are comparable to the *Professional Practices* of the CS2013 recommendations, although the *Professional Practices* are more thoroughly integrated in CS2013 (The Joint Task Force on Computing Curricula Association for Computing Machinery (ACM) IEEE Computer Society, 2013, pp. 15–16) up to being part of their own subject area, *Social Issues and Professional Practice* (The Joint Task Force on Computing Curricula Association for Computing Machinery (ACM) IEEE Computer Society, 2013, pp. 192–203).

16 see https://gi.de/ueber-uns, accessed on 17 July 2025, only available in German.

17 Representing all universities that offer study programs in computer science.

18 Representing all other higher education institutes that offer study programs in computer science.

19 https://www.asiin.de/en/home.html, accessed on 23 September 2025.

20 Internationally recognized might be the one by the ACM/IEEE/AAAI (Kumar et al., 2024)

21 Unfortunately, the accord does not define what computer science/computing actually is. It defines multiple professional levels (*Seoul accord*, 2008, Sec. D.2); however, they are defined as "*Apply knowledge of computing fundamentals, knowledge of a computing specialization, and mathematics, science, and domain knowledge appropriate for the computing specialization...*" (*Seoul accord*, 2008, D.2.3) and neither *knowledge of computing fundamentals* nor *knowledge of a computing specialization* are actually defined. Therefore, it is impossible to derive learning outcomes on a content level from it.

22 According to https://www.seoulaccord.org/signatories.php, accessed on 23 September 2025. A list of accreditation agencies that are officially recognized in Germany can be found at https://www.akkreditierungsrat.de/en/accreditation-system/agencies/agencies, accessed on 23 September 2025.

23 Available at https://web.arbeitsagentur.de/studiensuche/, accessed on 23 September 2025 (only available in German).

24 In German education, *dual study* or *work-study* programs combine an academic degree with related vocational training in a company.

25 In English, it is also known as Mecklenburg-Western Pomerania.

26 Most universities in Germany offer areas where you can choose between different courses and sometimes even include courses from different disciplines. Since it is not clear if every graduate has gained the competences listed in these non-mandatory courses, we did not include them in the study.

27 While the study used the German book (Mayring, 2022) as a foundation, an English version also exists and is provided here as reference Mayring (2021). Note that the English version prescribes a strict eight-step process, which is only implied in the German version.

28 That is, math competences that could not be (clearly) assigned to one of the already defined math subject domains. These deal with, for example, proofs and formal notation, which are relevant for all fields of math.

29 Notable mentions for content are: types of companies, company structures, and the interaction between CS/ICT and business management.

30 Analogously to Note 7, *General Computer Science* describes topics and competences related to computer science that could not clearly be assigned to any single subject domain, such as general skills in handling computers, computational thinking, or tool skills like LATEX.

31 There could be an argument made about whether *2a Transfer*, at least for the GI recommendations, should belong to the first or the second bin. The quality of the fit does not vary greatly. However, as it is immediately obvious where the border between bins for the curricula is situated, it was chosen to keep the models for both distributions within the same constraints.

32 For example, the Hamburg Law for Higher Education (*Hamburgisches Hochschulgesetz*) demands "*die pädagogische Eignung für die Lehre an der Hochschule,*" (HmbHG, 2025, § 15 (1) 2.) (translation: "the pedagogical qualification for teaching at an institute of higher education"). However, the qualification is only suggested to be certified by displaying relevant accomplishments during the *Juniorprofessur* (HmbHG, 2025, §15 (2)) (a Juniorprofessur is an assistant professor without tenure intended to learn the job of "professor" and qualify themselves for tenure positions instead of getting a traditional qualification through habilitation). This often amounts to having taught courses, not possessing any formal training.

33 For example, *Christian-Albrechts-Universität zu Kiel (1-Fach)—infDB01a Database Systems* refers to the course "infDB01a Database Systems" as offered in the study program "1-Fach" at the institute "Christian-Albrechts-Universität zu Kiel".

34 Since *No Level* is not desirable, it is not shown as an example.

35 Another perfectly valid translation might be "*Work in a group to analyze problems, develop solutions together and use presentation methods.*"—a translation suggested by a different author—which is further evidence for how unclear this learning outcome actually is, as it differs somewhat in how the group work is scoped: The first translation requires group presentations whereas the second does not.

36 Any personal positions of the authors notwithstanding, the authors do not want to make a claim concerning the importance of any one subject area over any other.

37 And in fact, Anderson et al. (2001, p. 34) mentioned that to clearly assign learning outcomes to a level, it is important to include other information such as observation of the classroom, analysis of assessment items, or the instructors' intend.

38 Not unlike what "modern" vernacular refers to as *Problem-Based Learning*.

39 *"Analysis emphasizes the breakdown of the material into its constituent parts and detection of the relationships of the parts and of the way they are organized."* (Bloom et al., 1956, p. 144) vs. *"Evaluation is defined as the making of judgments about the value, for some purpose, of ideas, works, solutions, methods, material, etc."* (Bloom et al., 1956, p. 185)

40 The exact quote is *"Da außerdem das Analysieren unabdingbare Voraussetzung für das Bewerten ist, [. . .]"* (Zukunft, 2016, p. 68) , which translates to "Since Analysis is an indispensable requirement for Evaluation, [. . .]"

41 Which combines economics, business administration, as well as computer science and is similar to "information systems" in English. For an example, see https://www.haw-hamburg.de/en/bachelor-business-information-systems/, accessed on 30 September 2025.

42 With the exception of `XXI` *Artificial Intelligence*, `XXII` *Signal Processing*, and `ICIX` *General Maths*, the new topics can be found in the ACM/IEEE/AAAI recommendations (Kumar et al., 2024) but are fringe topics subsumed under larger topics.

43 E.g., computer science at the University of Hamburg contains a module *Seminar* which is mandatory and teaches research methods; however, students can choose one from several different courses to fill that module.

44 On the other hand, one could argue that the coding actually represents the reality well and that those sections containing only keywords/chapters should be considered violating the outcome-based education approach.

45 For a hypothetical example, for databases it is mentioned that "Students perform queries on common database models", and a specific database like MongoDB is (only) mentioned in the keywords of all universities; that discovery would have been a valuable insight for the question of the diverging paths. In this hypothetical scenario, the discovery could be that universities do not teach relational databases but all teach the same NoSQL database.

46 Strictly speaking, Anderson et al. (2001, see p. 34) does not speak of coders but of instructors having the task to classify teaching material. Therefore, Anderson et al. (2001) suggests not only looking at formal descriptions but also observing the course itself, looking at the examination, and talking to the instructors. Since this study—by design—only looked at the curricula, such an extensive approach was not possible

References

Agentur für Qualitätssicherung durch Akkreditierung von Studiengängen e. V. (2019). *Gutachten zur Akkreditierung der Studiengänge · "Informatik" (B.Sc.) · "Informatik" (M.Sc.) · "Praktische Informatik" (M.Sc.) an der FernUniversität in Hagen (No. 449-xx–3)*. Hannover. Available online: https://antrag.akkreditierungsrat.de/dokument/c73b8c84-5b9c-469a-9530-3fc6d9443165 (accessed on 23 September 2025).

Allegre, C., Berlinguer, L., Blackstone, T., & Rüttgers, J. (1998). *Sorbonne Joint Declaration. Joint declaration on harmonisation of the architecture of the European higher education system*. Available online: https://ehea.info/media.ehea.info/file/1998_Sorbonne/61/2/1998_Sorbonne_Declaration_English_552612.pdf (accessed on 2 July 2025).

Anderson, L. W., Krathwohl, D. R., Airasian, P. W., Cruikshank, K. A., Mayer, R. E., Pintrich, P. R., Raths, J., & Wittrock, M. C. (Eds.). (2001). *A taxonomy for learning, teaching, and assessing: A revision of bloom's taxonomy of educational objectives (abridged edition)*. Addison Wesley Longman, Inc.

Annala, J., Lindén, J., & Mäkinen, M. (2015). Curriculum in higher education research. In J. M. Case, & J. Huisman (Eds.), *Researching higher education—International perspectives on theory, policy and practice* (pp. 171–189). Taylor & Francis Group. [CrossRef]

ASIIN e.V. (2018). *Fachspezifisch ergänzende hinweise des fachausschusses 4—informatik zur akkreditierung von bachelor- und masterstudiengängen der informatik (verabschiedet: 29. März 2018)*. Available online: https://www.asiin.de/files/content/kriterien/ASIIN_FEH_04_Informatik_2018-03-29.pdf (accessed on 23 September 2025).

Aubel, I., Zug, S., Dietrich, A., Nau, J., Henke, K., Helbing, P., Streitferdt, D., Terkowsky, C., Boettcher, K., Ortelt, T. R., Schade, M., Kockmann, N., Haertel, T., Wilkesmann, U., Finck, M., Haase, J., Herrmann, F., Kobras, L., Meussen, B., ..., & Versick, D. (2022). Adaptable Digital Labs - Motivation and Vision of the CrossLab Project. In *2022 IEEE German Education Conference (GeCon)* (pp. 1–6). IEEE. [CrossRef]

Auer, T. (2022). Die Wirksamkeit von Lernzielen für Studienleistungen–eine experimentelle Studie. *Die Hochschullehre*, *8*, 662–675. [CrossRef]

Biggs, J. (1996). Enhancing teaching through constructive alignment. *Higher Education*, *32*(3), 347–364. [CrossRef]

Biggs, J., & Tang, C. (2011). *Teaching for quality learning at university: What the student does* (4th ed.). Society for Research into Higher Education & Open University Press.

Bloom, B. S., Engelhart, M. D., Furst, E. J., Hill, W. H., & Krathwohl, D. R. (1956). *Taxonomy of educational objectives: The classification of educational goals. handbook 1: Cognitive domain*. McKay New York.

Brahimi, T., Sarirete, A., & Ibrahim, R. M. (2016). The impact of accreditation on student learning outcomes. *International Journal of Knowledge Society Research*, *7*(4), 51–62. [CrossRef]

Charlier, J.-É., & Croché, S. (2007). The Bologna process: The outcome of competition between Europe and the USA and now a stimulus to this competition. *European Education*, *4*, 10–26. Available online: https://dial.uclouvain.be/pr/boreal/en/object/boreal%3A180588/datastream/PDF_01/view (accessed on 11 November 2025).

Cooper, H. M. (1988). Organizing knowledge syntheses: A taxonomy of literature reviews. *Knowledge in Society*, *1*(1), 104–126. [CrossRef]

Desel, J. (2017). Die Entwicklung neuer GI-empfehlungen für informatik-studiengänge. In *INFORMATIK 2017* (pp. 235–240). Gesellschaft für Informatik. [CrossRef]

Feldman, K. A. (1989). The association between student ratings of specific instructional dimensions and student achievement: Refining and extending the synthesis of data from multisection validity studies. *Research in Higher Education*, *30*(6), 583–645. [CrossRef]

Foundation for International Business Administration Accreditation. (2023). *Assessment guide for the accreditation of bachelor and master programmes by FIBAA*. Available online: https://www.fibaa.org/fileadmin/redakteur/pdf/PROG/Handreichungen_und_Vorlagen/231006_AG_PROG_Bachelor_Master_en.pdf (accessed on 11 November 2025).

Gesellschaft für Informatik e.V. (2005). *Empfehlungen für bachelor- und masterprogramme im studienfach informatik an hochschulen*. Gesellschaft für Informatik e.V. Available online: https://dl.gi.de/bitstreams/dc7ba3f3-979d-46bf-b69b-7643ff715d55/download (accessed on 23 September 2025).

Hamburgisches Hochschulgesetz (HmbHG) Vom 18. Juli 2001 zuletzt geändert durch Gesetz vom 19. Februar 2025, HmbGVBl. S. 241. (2025). Available online: https://www.landesrecht-hamburg.de/bsha/document/jlr-HSchulGHArahmen (accessed on 23 September 2025).

Harden, R. M. (2002). Learning outcomes and instructional objectives: Is there a difference? *Medical Teacher*, *24*(2), 151–155. [CrossRef] [PubMed]

Harrow, A. J. (1972). *A taxonomy of the psychomotor domain: A guide for developing behavioral objectives*. Longman.

Hartel, R., & Foegeding, E. (2004). Learning: Objectives, competencies, or outcomes? *Journal of Food Science Education*, *3*(4), 69–70. [CrossRef]

Holmes, A. G. D., Tuin, M. P., & Turner, S. L. (2021). Competence and competency in higher education, simple terms yet with complex meanings: Theoretical and practical issues for university teachers and assessors implementing Competency-Based Education (CBE). *Educational Process International Journal*, *10*(3), 39–52. [CrossRef]

Johnson, C. G., & Fuller, U. (2006). Is Bloom's taxonomy appropriate for computer science? In *Proceedings of the 6th baltic sea conference on computing education research: Koli calling 2006* (pp. 120–123). Association for Computing Machinery. [CrossRef]

Kennedy, D. (2006). *Writing and using learning outcomes: A practical guide*. University College Cork.

Kiesler, N. (2020). Towards a competence model for the novice programmer using bloom's revised taxonomy—An Empirical Approach. In *Proceedings of the 2020 acm conference on innovation and technology in computer science education* (pp. 459–465). Association for Computing Machinery. [CrossRef]

Krathwohl, D. R. (2002). A revision of bloom's taxonomy: An overview. *Theory Into Practice*, *41*(4), 212–218. Available online: http://www.jstor.org/stable/1477405 (accessed on 7 January 2024). [CrossRef]

Krathwohl, D. R., Bloom, B. S., & Masia, B. B. (1964). *Taxonomy of educational objectives, the classification of educational goals; handbook II: The afffective domain*. Dacid KcMay.

Kumar, A. N., Raj, R. K., Aly, S. G., Anderson, M. D., Becker, B. A., Blumenthal, R. L., Eaton, E., Epstein, S. L., Goldweber, M., Jalote, P., Lea, D., Oudshoorn, M., Pias, M., Reiser, S., Servin, C., Simha, R., Winters, T., & Xiang, Q. (2024). *Computer science curricula 2023*. Association for Computing Machinery.

Marzano, R. J. (1998). *A theory-based meta-analysis of research on instruction*. Mid-Continent Regional Educational Lab.

Mayring, P. (2021). *Qualitative content analysis: A step-by-step guide*. SAGE Publications.

Mayring, P. (2022). *Qualitative Inhaltsanalyse: Grundlagen und Techniken* (13th ed.). BELTZ.

Ministerial Conference in Berlin. (2003). *"Realising the european higher education area". Communiqué of the conference of ministers responsible for higher education in Berlin on 19 September 2003*. Available online: https://ehea.info/media.ehea.info/file/2003_Berlin/28/4/2003_Berlin_Communique_English_577284.pdf (accessed on 2 July 2025).

Ministerial Conference in Bologna. (1999). *The bologna declaration of 19 june 1999. Joint declaration of the european ministers of education*. Available online: https://ehea.info/media.ehea.info/file/Ministerial_conferences/02/8/1999_Bologna_Declaration_English_553028.pdf (accessed on 2 July 2025).

Nodine, T. (2016). How did we get here? A brief history of competency-based higher education in the United States. *The Journal of Competency-Based Education*, *1*(1), 5–11. [CrossRef]

Pancratz, N., & Grave, V. (2023, September 13–14). *"Informatik und gesellschaft" in der hochschullehre: Eine untersuchung von modulhandbüchern verschiedener informatikstudiengänge an norddeutschen universitäten*. 10. Fachtatung Hochschuldidaktik Informatik (HDI 2023), Aachen, Germany.

Seoul accord. (2008). Taipei. Available online: https://www.seoulaccord.org/ (accessed on 19 September 2025).

Simpson, B. J. (1966). The classification of educational objectives: Psychomotor domain. *Illinois Journal of Home Economics*, *10*(4), 110–114.

Tedre, M. (2011). Computing as a science: A survey of competing viewpoints. *Minds and Machines*, *21*(3), 361–387. [CrossRef]

The Joint Task Force on Computing Curricula Association for Computing Machinery (ACM) IEEE Computer Society. (2013). *Computer science curricula 2013: Curriculum guidelines for undergraduate degree programs in computer science*. ACM, Inc. [CrossRef]

Tyler, R. W. (1949). *Basic principles of curriculum and instruction*. University of Chicago Press.

Tyler, R. W., & Hlebowitsh, P. S. (2013). *Basic principles of curriculum and instruction*. University of Chicago Press.

Velázquez-Iturbide, J. A. (2021). An analysis of the formal properties of bloom's taxonomy and its implications for computing education. In *Proceedings of the 21st koli calling international conference on computing education research* (pp. 1–7). Association for Computing Machinery. [CrossRef]

Zentrale Evaluations- und Akkreditierungsagentur. (2018). *Akkreditierungsbericht zum akkreditierungsantrag der FHDW hannover abteilung betriebswirtschaftslehre 449-xx-3 (No. 449-xx–3)*. Hannover. Available online: https://hub.zeva.org/AkkreditierteStudiengaengeAltesSchema/Downloads/Download/176 (accessed on 23 September 2025).

Zukunft, O. (2016). *Empfehlungen für bachelor- und masterprogramme im studienfach informatik an hochschulen (Juli 2016)*. Gesellschaft für Informatik e.V. Available online: https://dl.gi.de/handle/20.500.12116/2332 (accessed on 23 September 2025).