



**FernUniversität
in Hagen**

The Recognition of Prior Learning in an Undergraduate Program in Informatics - Insights and Challenges

PROJECT PERMEABILITY

Prof. Dr. Uwe Elsholz, Hoai Nam Huynh, Prof. Dr. Jörg Desel





Agenda

- **Background and Approach of the project**
- **Conduction of the project**
- **Experiences**
- **Insights and Challenges**
- **Discussion**

Lifelong Learning and Permeability in Germany

“The legacy of the past: the segmentation between vocational and higher education” (Baethge/Wolter 2015) –
 „Das deutsche Bildungs-Schisma“ (Baethge 2006)

Table 1 Institutional orders of VET (dual system) and Higher Education

Distinctive criteria of skill formation institutions	Higher education ^a	Dual system of vocational education and training
Guiding principle (main objectives)	Academic professionalism	Vocational competence in firms and labour markets
Normative references for curricula	Representative systematic knowledge in academic disciplines	Labour market and employment structure; economic demand for qualifications
Political governance	<i>Länder</i> governments	Corporatist: (self-)governance by social partners (chambers of trade, industry and commerce), regulated by federal law
Financing	Public by the <i>Länder</i> governments	Mainly by private enterprises
Status of the learners	Students	Apprentices with employment contract
Location/organisation of learning processes	Universities; institutions independent from workplaces	Firms; integrated in work processes

Developments concerning permeability



Elsholz, U. (2019):

Lebenslanges Lernen als Katalysator der Modernisierung des Bildungssystems: Über den Kollateralnutzen neoliberaler Bildungspolitik.

In: Walgenbach, K. (Hrsg.): Bildung und Gesellschaft im 21. Jahrhunderts. Zur neoliberalen Neuordnung von Staat, Ökonomie und Privatsphäre. Frankfurt a.M., S. 161-177.



Facts from Fernuniversität

Students' Sociodemographic characteristics	FernUniversität	Germany*
Currently employed	85 %	No data
Age at study begin	32 years (first semester)	24.7 years (all)
With professional qualification	53 %	22% (14 % Uni)

* 21. Sozialerhebung DSW, 2016; ohne Fernstudierende



Permeability Between Vocational Training and Bachelor's Degree - From IT Specialist to Bachelor's Degree in Computer Science Through Adapted Study Design

Prof. Dr. Jörg Desel, Chair of Software Engineering and Theory of Programming (and Dean of Faculty of Mathematics and Computer Science)

Prof. Dr. Uwe Elsholz, Chair of Lifelong Learning

Staff: Hoai Nam Huynh, Dr. Len Ole Schäfer, Carsten Thorbrügge

Project duration: 01.01.2019-31.12.2021

Located in Research Cluster D²L² „Digitalization, Diversity and Lifelong Learning. Consequences for Higher Education“

Additional funding Stifterverband für die deutsche Wissenschaft (Fellowship-program)



Generals aims of the project „permeability“

- to enable lifelong learning from a learners perspective
- to support the mission of Fernuniversität as an open university/
a university for lifelong learning
- to become more attractive for learners



Central question of the project

How can a university bachelor's degree program be adapted for those who have undergone vocational training and who have work experience, so that they can achieve their educational goals with less effort, taking into account **existing competencies**?

Goal: reduction of time for learners by up to 60 ECTS (from 180 ECTS)



Goal: reduction of time for learners by up to 60 ECTS

- up to 15 ECTS **lump-sum recognition** (formal learning; IT-specialist/Fachinformatiker:in)
- up to 20 ECTS **individual recognition** (non-formal and informal learning/work experience)
- up to 25 ECTS **adaption of learning pathways/ „fast track“** (informal learning)

Lump-sum recognition

Application

Submission of a training certificate
=> uncomplicated recognition

Underlying Idea

- This recognition applies to a **whole group** of people (such as IT specialists)
- The competencies that have to be acquired at higher education institutions can be **replaced** by vocationally acquired competences
- For this purpose, a verification **procedures** must be developed and executed once for each study module



Equivalency assessment - procedure

Step 1: Data preparation

- Identifying competencies
- Paraphrasing into learning outcome-oriented description



Illustration: Alexander Sperl

Step 2: Content comparison (competency synopsis)

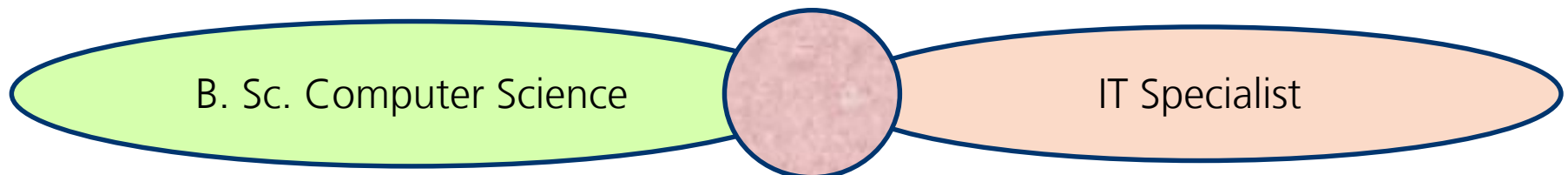
- Comparing competencies according to content areas
- Evaluate content overlap

Step 3: Level comparison (recognition assessment)

- Assigning competencies to the AKT levels

Reference documents for competencies

- *University:*
 - Module manual
 - Exercises
 - Course Material
- *Vocational training:*
 - Rahmenlehrplan
 - Ausbildungsordnung





Equivalency assessment in depth

Content Analysis			
Modul: 63811 Introduction to Imperative Programming			
Study Programme: B. Sc. Informatik			
Vocational Training - IT Specialist		Student B.Sc. Computer Science	
1	FI kennen Werkzeuge zur Modellierung und Strukturierung von Programmen und Datenbanken	Knowl.	
2	FI kennen eine oder mehrere Programmiersprachen	Knowl.	
3	FI kennen wichtige Programmier- und Sprachkonzepte	Knowl.	
4	FI kennen verschiedene Programmierentwicklungsmethoden	Knowl.	
8	FI wenden Programmier- und Programmierentwicklungsmethoden an	Skill	
9	FI wenden wichtige Programmierkonzepte an	Skill	
17	FI führen Anforderungsanalysen zur Modellierung von Software durch	Skill	
1W19	FI Ae führen eine Anforderungsanalyse für eine Software durch	Skill	
2W19	FI Ae führen eine Nutzenanalyse für eine Software durch und bewerten den Nutzen einer solchen Analyse	Skill	
23	FI entwickeln unternehmens- oder kundenspezifische Konzepte zur Entwicklung von Anwendungsprogrammen	Skill	
1W18	FI Ae erstellen den Projektplan zu einem Softwareprojekt	Skill	
1519	FI Ae entwerfen Projektstrukturplan zu einem Softwareprojekt	Skill	
5	FI kennen die Grundregeln der Gestaltung ergonomischer Software	Knowl.	
15	FI setzen die Grundregeln zur Gestaltung ergonomischer Software um	Skill	
11	FI wenden der jeweiligen Programmiersprache angemessene Entwicklungsumgebungen an	Skill	
13	FI setzen ausgearbeitete Konzepte in funktionierende Anwendungen um	Skill	
5W19	FI Ae formulieren Anweisungen und Kontrollstrukturen zur Lösung gegebener Problemstellungen, zum Beispiel zum Bearbeiten	Skill	
3W19	FI Ae entwickeln Algorithmen und Funktionen zur Verarbeitung gegebener Listen	Skill	
4W19	FI Ae erstellen Algorithmen und Funktionen zur Verarbeitung von Datums- und Zeitwerten	Skill	
14	FI setzen angemessene Programmiersprachen ein	Skill	
18	FI analysieren bestehenden Quellcode	Skill	
5519	FI Ae erläutern , warum eine Methode ohne Rückgabewert auf bestimmte Daten zurückgreifen kann	Skill	
19	FI modifizieren Quellcode nach Anforderung	Skill	
2K	FI erweitern Struktogramme mit verschachtelten Kontrollstrukturen zu einem textuell beschriebenen Problem	Skill	
3K	FI verwenden vorgefertigte Funktionen zum Entwickeln eigener Programmabläufe	Skill	
20	FI erweitern bestehende Software um kundenspezifische Funktionen oder Schrittbefehle	Skill	
6519	FI SI erstellen in Pseudocode einen Algorithmus zur Passwortgenerierung	Skill	
2W18	FI Ae definieren eine Methode zu einer betriebswirtschaftlichen Fragestellung	Skill	
3W18	FI Ae entwickeln eine Methode zur Generierung eines AbfCodes	Skill	
2519	FI Ae fertigen ein Programm zur Auswertung einer Log-Datei an	Skill	
3519	FI Ae entwerfen (als Pseudocode...) eine Funktion druckeReport zur Auswertung eines Arrays mit Fehlerreport	Skill	
4519	FI Ae implementieren eine Methode in Pseudocode, die eine Zeichenkette zurückgibt	Skill	
7	FI kennen grundlegende Verfahren des Softwaretests	Knowl.	
10	FI testen Software und Softwarekomponenten nach definierten Testverfahren	Skill	
1K	FI führen zu einem gegebenen Struktogramm mit verschachtelten Kontrollstrukturen einen Schreibtischtest durch .	Skill	
M1	Sie sind mit grundlegenden Imperativen Programmierkonzepten vertraut .	Knowl.	
M2	Sie kennen Richtlinien für guten Programmierstil.	Knowl.	
M4	Sie bearbeiten kleinere Programmieraufgaben.	Skill	
2002	Sie erstellen eine Problemspezifikation zu einem gegebenen Programm (Eingabe, Ausgabe, Nachbedingung)	Skill	
2004	Sie vervollständigen ein gegebenes Programm nach Spezifikation	Skill	
2005	Sie schreiben ein Programm nach Spezifikation [z. B. zur Maximumsuche]	Skill	
2011	Sie schreiben eine iterative Funktion zur Polynomauswertung nach gegebenen Typdefinitionen und weiteren Vorgaben	Skill	
M10	Sie verwenden einfache Datentypen.	Skill	
2001	Sie benennen Typen von Ausdrücken zu einem gegebenen Programm	Skill	
M15	Sie verwenden einfache Anweisungen.	Skill	
M16	Sie verwenden zusammengesetzte Anweisungen.	Skill	
M17	Sie verwenden Schleifen.	Skill	
M20	Sie verwenden Rekursion.	Skill	
2013	Sie vervollständigen eine rekursive Funktion zu einem Listenvergleich nach gegebenen Typdefinitionen und weiteren Vorgaben	Skill	
2014	Sie vervollständigen eine rekursive Funktion zur Binärbaumausgabe nach gegebenen Typdefinitionen und weiteren Vorgaben	Skill	
2015	Sie vervollständigen eine rekursive Funktion zur Tiefenbestimmung eines Binärbaumes nach gegebenen Typdefinitionen und Skill	Skill	
2016	Sie vervollständigen eine rekursive Funktion zur Prüfung auf Strukturidentität zweier Listen nach gegebenen Typdefinitionen	Skill	
2008	Sie identifizieren ein Programmstück zu einer gegebenen Matrix-Transponier-Funktion	Skill	
M19	Sie verwenden Funktionen und Prozeduren.	Skill	
2006	Sie schreiben eine Funktion zur Überprüfung der Türme auf einem Schachfeld nach Spezifikation	Skill	
2007	Sie geben eine Problemspezifikation zu einer neuen gegebenen Funktion an	Skill	
M6	Sie kennen wichtige grundlegende Algorithmen.	Knowl.	
M5	Sie kennen wichtige grundlegende Datenstrukturen.	Knowl.	
M7	Sie wählen die jeweils geeignete Datenstruktur aus .	Skill	
M8	Sie passen die jeweils geeignete Datenstruktur an .	Skill	
M9	Sie analysieren Algorithmen auf Effizienz.	Skill	
M11	Sie verwenden strukturierte Datentypen.	Skill	
M12	Sie verwenden Zeiger.	Skill	
M13	Sie verwenden einfache dynamische Datenstrukturen wie lineare Listen.	Skill	
2009	Sie ermitteln Ergebnisse aus Aufrufen zur Listenmultiplizierung mit gegebenen Typdefinitionen, Funktionen und Listen	Skill	
2010	Sie ermitteln Rückgaben aus Aufrufen zur Listenabschreibung mit gegebenen Typdefinitionen, Prozeduren und Funktionen	Skill	
2012	Sie schreiben eine iterative Funktion zur Listenverarbeitung nach gegebenen Typdefinitionen und weiteren Vorgaben	Skill	
2013	Sie vervollständigen eine rekursive Funktion zu einem Listenvergleich nach gegebenen Typdefinitionen und weiteren Vorgaben	Skill	
2016	Sie vervollständigen eine rekursive Funktion zur Prüfung auf Strukturidentität zweier Listen nach gegebenen Typdefinitionen	Skill	
M14	Sie verwenden einfache dynamische Datenstrukturen wie Binäräume.	Skill	
2014	Sie vervollständigen eine rekursive Funktion zur Binärbaumausgabe nach gegebenen Typdefinitionen und weiteren Vorgaben	Skill	
2015	Sie vervollständigen eine rekursive Funktion zur Tiefenbestimmung eines Binärbaumes nach gegebenen Typdefinitionen und Skill	Skill	
M3	Sie kennen Grundlagen des Testens.	Knowl.	
2007	Sie geben einen kompakten Kontrollflussgraphen zu einer Funktion Summe der natürlichen Zahlen an	Skill	
2018	Sie geben Äquivalenzklassen für einen Black-Box-Test einer Prozedur zur Teilerprüfung von natürlicher Zahlen an	Skill	
2019	Sie verifizieren Knotenfolgen für eine vollständige Anweisungsüberdeckung einer Funktion	Skill	
2020	Sie verifizieren Knotenfolgen für eine vollständige Zweigüberdeckung einer Funktion	Skill	
2021	Sie verifizieren Testdaten für eine vollständige Zweigüberdeckung eines Kontrollflussgraphen einer Funktion	Skill	

Modelling and Concept Development

Software Ergonomics

Implementing and Programming

Testing

Modelling and Concept Development

Software Ergonomics

Implementing and Programming

Testing

Equivalency assessment in depth

Vocational Training - IT Specialist

1	ITS know tools for modelling and structuring programmes and databases	Knowl.
2	ITS know one or more programming languages	Knowl.
3	ITS know important programming and language concepts	Knowl.
4	ITS know different programme development methods	Knowl.
8	ITS apply programming and programme development methods	Skill
9	ITS apply important programming concepts	Skill
17	ITS carry out requirements analysis to model software	Skill
1W19	ITS-AD carry out a requirements analysis for software	Skill
2W19	ITS-AD conduct a utility analysis for software and evaluate the utility of such an analysis	Skill
23	ITS develop company or customer-speciITSc concepts for the development of application programmes	Skill
1W18	ITS draw up the project plan for a software project	Skill
1S19	ITS draw up a work breakdown structure for a software project	Skill
5	ITS know the basic rules for designing ergonomic software	Knowl.
15	ITS implement the basic rules for the design of ergonomic software	Skill
11	ITS use development environments appropriate to the programming language in question	Skill
13	ITS convert elaborated concepts into functioning applications	Skill
5W19	ITS-AD formulate instructions and control structures to solve given problems, for example to handle state transitions	Skill
3W19	ITS develop algorithms and functions for processing given lists	Skill
4W19	ITS create algorithms and functions for processing date and time values	Skill

Modelling
and Concept
Develop-
ment

Software
Ergonomics

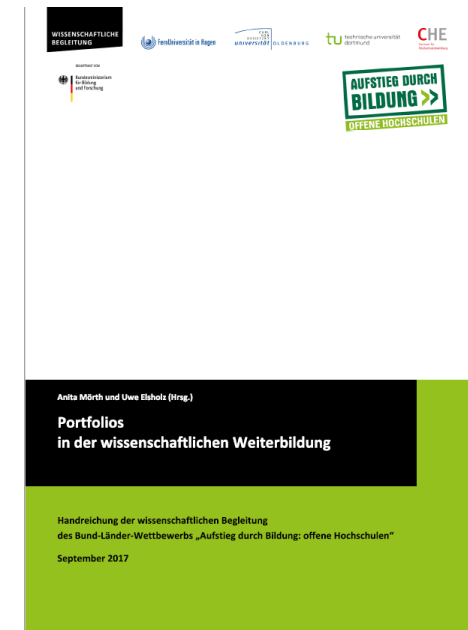
Current Status

- Recognition of the module „*Imperative Programmierung*“ for all IT specialists
- Recognition of the module „*Grundpraktikum der Programmierung*“ for IT specialists for application development (but not for systemintegration)



Individual Recognition of Modules

- Portfolio as a national and international standard for purposes of recognition of prior learning, e.g. France and Swiss (Bernhard 2015)
- Checked on case-by-case basis
- Suitable for reflecting informal and non-formal learning



Mörth/Elsholz 2017

Individual Recognition - procedure

1.

- Providing information
- Counselling and giving guidance
- Submission of application of recognition by the student via portfolio



2.

- Formal, content-related and level-related verification by examination office.
- If decision unclear: verification by person responsible of the module + expert discussion (optional)



3.

- Notification about the decision by examination office.
- If negative decision: opposition proceedings possible for student.



Individual Recognition - procedure

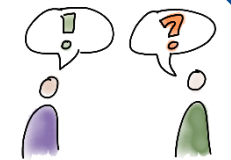
1.

- Providing information
- Counselling and giving guidance
- Submission of application of recognition by the student via portfolio



2.

- Formal, content-related and level-related verification by examination office.
- If decision unclear: verification by person responsible of the module + expert discussion (optional)


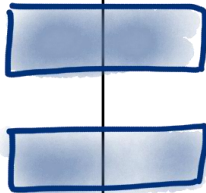



3.

- Notification about the decision by examination office.
- If negative decision: opposition proceedings possible for student.



Step 1 – Filling out portfolio

Academic learning outcomes	Non-academic learning outcomes
<ul style="list-style-type: none"> • Module descriptions • Course material 	<ul style="list-style-type: none"> • Vocational training • Further training • Professional experience • Volunteer work • ...  

Support: Guideline + assistance through advice

Step 1 - Example



Instead of empty form: filled out **module content** and **learning outcomes of module**

Matrikel-Nr.: █████	Name: █████	Vorname: █████
Modulnummer: 63811		Modultitel: Einführung in die imperative Programmierung

Hinweis: Bitte beachten Sie beim Ausfüllen der Tabelle, dass die Inhalte, die in der nachstehenden Box aufgeführt sind, zwingend weitestgehend abgedeckt sein müssen. Ansonsten ist eine Anerkennung des Moduls nicht möglich.

Inhalte des Moduls „Einführung in die imperative Programmierung“
<p>Kurs 01613 bildet den Einstieg in die Programmierausbildung und stellt grundlegende imperative Programmierkonzepte sowie ihre typische Anwendung vor, um kleine Programme zu entwickeln. So werden u.a. einfache und strukturierte Datentypen behandelt. Des Weiteren wird sich mit einfachen und zusammengesetzten Anweisungen und Konstrukten wie Schleifen und Funktionen befasst. Darauf aufbauend werden weitere Techniken wie z.B. Rekursion eingeführt und einfache dynamische Datenstrukturen implementiert. Zur praktischen Erläuterung und Umsetzung dieser Konzepte wird eine typisierte imperative Programmiersprache verwendet. Die in dem Kurs vermittelten imperativen Konzepte bilden auch eine wichtige Grundlage der objektorientierten Programmierung. Im Kurs wird von Beginn an Wert auf guten Programmierstil gelegt, um auf diese Weise die Erstellung von leicht lesbarem und zuverlässigem Quellcode zu fördern.</p> <p>Workload: 150 Stunden</p>

Lernergebnisse des Moduls	Eigene Lernergebnisse („Ich bin in der Lage...“, „Ich kann...“)	Nachweise
<p>Nach erfolgreicher Teilnahme sind die Studierenden mit grundlegenden imperativen Programmierkonzepten vertraut.</p>	<p>In meiner Ausbildung lernte ich die imperative Programmierung mit der Programmiersprache C kennen. Inhalte der Ausbildung waren das Arbeiten mit Zahlen, hierbei die Unterschiede von ganzen Zahlen und Fließkommazahlen und das Rechnen mit diesen. Weiterhin wurden die verschiedenen Basis Datentypen und deren Variationen vermittelt, sowie die Umwandlung von einem Datentyp in einen anderen. Auch kom-</p>	<p>Abschlusszeugnis technischer Assistent für Informatik</p>



Step 1 - Example

	<p>und While Schleifen und die Verschachtelung von Schleifen. Bestandteil war auch das Prinzip von Pointern und Adressen, darauf Aufbauend Konzepte wie Listen, Bäume und Graphen sowie dynamische Speicherverwaltung. Auch komplexere Konzepte wie Rekursion und Multithreading wurden behandelt. Inhalt waren auch Methoden zur Strukturierung und Modularisierung des Quellcodes mit Hilfe von Funktionen und Bibliotheken.</p>	
<p>Die praktische Anwendung sämtlicher Lerninhalte beherrschen sie im Rahmen von kleineren Programmieraufgaben.</p>	<p>Schon in der Ausbildung wurde das Wissen anhand von Beispielen und praktischen Programmieraufgaben vermittelt. Im Rahmen meiner ersten mehrjährigen Tätigkeit als Softwareentwickler vertiefte ich besonders meine Kenntnisse in der imperativen Programmierung und der Programmiersprache C durch die Konzeptionierung und Erstellung von Programmen für Mikrocontroller und Eingebettete Systeme. Besonderes Augenmerk lag hierbei auf Optimierung der Programme im Bezug auf Speicher- und Ressourcenverbrauch, da solche Systeme hier sehr limitiert sind. Dabei halfen mir besonders die Konzepte wie Zeiger und dynamischer Speicherverwaltung, aber auch alle anderen erlangten Fähigkeiten und Techniken fanden und finden immer noch Anwendung in meinen Tätigkeiten als Softwareentwickler. Weiterhin übertrug ich diese Konzepte auf andere Programmiersprachen und auch in die objektorientierte Programmierung mit C#. Hierbei realisierte ich mehrere große Projekte unter anderem von Unternehmensanwendungen, die mit Datenbanken oder ebenfalls von mir erstellten Diensten über Webschnittstellen kommunizier-</p>	<p>Arbeitszeugnis [REDACTED]</p> <p>Arbeitszeugnis [REDACTED]</p>

Individual Recognition - procedure

1.

- Providing information
- Counselling and giving guidance
- Submission of application of recognition by the student via portfolio



2.

- Formal, content-related and level-related assessment by examination office.
- If decision unclear: verification by person responsible of the module + expert discussion (optional)

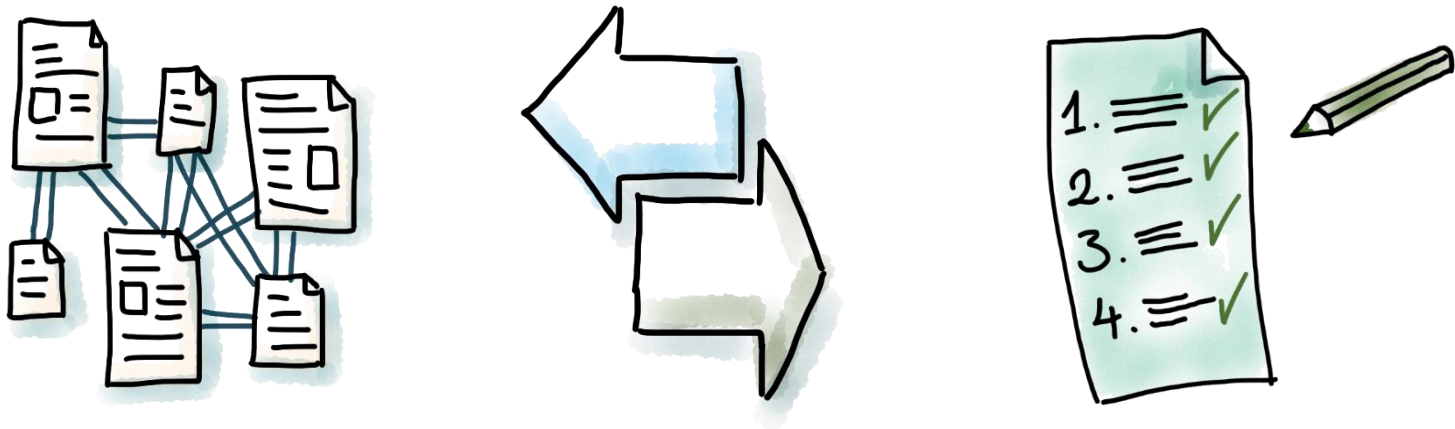


3.

- Notification about the decision by examination office.
- If negative decision: opposition proceedings possible for student.



Step 2 – Portfolio Assessment



1. Equivalency of content (Examination office, lecturer)
2. Equivalency of level (Anderson and Krathwohl)
3. Optional: Expert discussion based on portfolio

Current status

- Guidelines for students
- Guidelines for university members
- Counselling offer in addition to examinations office
 - Online form for more efficiency



FernUni / Formulare / Forschung / Durchlässigkeit / Individuelle Anerkennung beruf...

Individuelle Anerkennung beruflicher Leistungen - Beratungsangebot

Herzlich willkommen! Sie bringen Berufserfahrungen und/oder fachbezogene Ausbildungen mit? Sie möchten erfahren, ob und in welchem Umfang Ihre beruflichen Leistungen anerkannt werden können?

Wir unterstützen Sie dabei, herauszufinden, wo Anerkennungen möglich sind. Auf Wunsch unterstützen wir Sie individuell bei der Beantragung.

Um die Beratung zu erleichtern, geben Sie bitte im Folgenden Ihre Kontaktdaten und Informationen zu Ihrem Anerkennungsanliegen an (soweit diese bekannt sind). Wir werden uns dann bei Ihnen zurückmelden und mit Ihnen die nächsten Schritte besprechen.

Anmeldeformular

Kontaktangaben

Titel	<input type="text" value="optional"/>
Vorname	<input type="text" value="erforderlich"/>
Nachname	<input type="text" value="erforderlich"/>
E-Mail	<input type="text" value="erforderlich"/>

Was bringen Sie mit?

Bitte kreuzen Sie an, welche Leistungen Sie bereits mitbringen. Wichtig: Es muss sich um Leistungen handeln, die nicht aus einem hochschulischen Kontext stammen (z.B. Akademiestudium, Leistungen von anderen Hochschulen).

- Leistungen, die bei Bildungsträgern erworben wurden (z.B. Berufsausbildungen, IHK-Aufstiegsfortbildungen)
- Leistungen, die im beruflichen Kontext erworben wurden (z.B. einschlägige Berufserfahrung, Fortbildungen)

Current status

- Guidelines for students
- Guidelines for university members
- Counselling offer in addition to examinations office
 - Online form for more efficiency

Beschreiben Sie im Freitext-Feld kurz Ihre Leistungen (z.B. Ausbildung zum/zur Fachinformatiker/in Schwerpunkt Anwendungsentwicklung, mehrjährige Berufserfahrung in der Softwareentwicklung).

Leistungsbeschreibung

Wenn bekannt: Bitte wählen Sie die Module aus, für die eine Anerkennung beantragt werden soll.
Mehrfachnennungen sind möglich (über STRG + Linke Maustaste).

Module

Bitte auswählen

63013 Computersysteme
63113 Datenstrukturen und Algorithmen
63611 Einführung in die objektorientierte Programmierung
63113 Einführung in die imperative Programmierung
63012 Softwaresysteme
63081 Grundpraktikum Programmierung
63512 Sicherheit im Internet
63812 Software Engineering

Weitere Module im Studiengang Informatik B.Sc. etwa aus den Katalogen B und N können ebenfalls für eine Anerkennung in Frage kommen. Bitte geben Sie im folgenden Freitextfeld Modulnummer und Modultitel an (optional).

Weitere Module

Haben Sie weitere Fragen? Bitte nutzen Sie hierfür das folgende Freitextfeld (optional).

Ich möchte die Forschung zur individuellen Anerkennung unterstützen und an einer Evaluation teilnehmen.



Adaptation of Learning Pathways



Underlying Idea and current status

IT specialists bring previous knowledge with them from their vocational training



But they could save time, if they only had to work on the unknown topics of a module



Sometimes, however, it is not enough to credit a whole module for it

Solution:

Appreciation by offering adapted course material for IT specialists



Experience



- Lack of competence-oriented module descriptions make recognitions difficult
- Misunderstandings (f.e. „Anrechnung“ not anchored in the law) not in examination office nor in department of law
- Hesitating of staff
- Hesitating of students



Individual recognition: Many students seem to be interested but only few complete the process



Insights and new challenges

- Who should be **responsible** for recognition of prior learning? (professors, teaching staff, examination office, programm coordinator...)
- How to **build** more **competence** in the field of RPL within the institution? (capacity building)
- How to create a **comprehensive process** (with information and guidance for students from the beginning)?
- When and how to **inform/involve students**? How to activate them?
- How to establish a **positive narrative** of recognition?

References

- Anderson, L. W./Krathwohl, D. R. (Hrsg.) (2001): A taxonomy for learning, teaching, and assessing: a revision of Bloom's taxonomy of educational objectives. New York: Longman.
- Baethge, M. (2006): Das deutsche Bildungs-Schisma: Welche Probleme ein vorindustrielles Bildungssystem in einer nachindustriellen Gesellschaft hat. In: SoFi-Mitteilungen 34, S. 13-27. Online verfügbar: http://www.sofi-goettingen.de/fileadmin/SOFI-Mitteilungen/Nr._34/Baethge.pdf (Abruf: 02.09.2017).
- Baethge, M.; Wolter, A. (2015).. The German skill formation model in transition: from dual system of VET to higher education? *Journal for Labour Market Research* 48 (2): 97–112. doi: 10.1007/s12651-015-0181-x.
- Bernhard, N. (2017). Durch Europäisierung zu mehr Durchlässigkeit? Veränderungsdynamiken des Verhältnisses von beruflicher Bildung zur Hochschulbildung in Deutschland und Frankreich. Opladen
- Elsholz, U. (2019): Lebenslanges Lernen als Katalysator der Modernisierung des Bildungssystems: Über den Kollateralnutzen neoliberaler Bildungspolitik. In: Walgenbach, K. (Hrsg.): *Bildung und Gesellschaft im 21. Jahrhunderts. Zur neoliberalen Neuordnung von Staat, Ökonomie und Privatsphäre.* Frankfurt a.M., S. 161-177.

References

- Mörth, A./Elsholz, U. (Hrsg.) (2017): Portfolios in der wissenschaftlichen Weiterbildung. Handreichung der wissenschaftlichen Begleitung des Bund-Länder-Wettbewerbs “Aufstieg durch Bildung: offene Hochschulen”
- Müskens, W./Eilers-Schoof, A. (2013): Application of the Module Level Indicator (MLI) in the context of transnational comparisons of qualifications. In: Tutschner, R./Wittig, Wolfgang (Hrsg.): Level Assessments of Learning Outcomes in Health Care and Nursing. Bremen: Institut für Technik und Bildung (ITB).
- Opel, S. (2020). Entwicklung eines arbeitsprozessorientierten Kompetenzstrukturmodells für die Ausbildung zum Fachinformatiker bzw. zur Fachinformatikerin. Dissertation.
<https://doi.org/10.17185/dupublico/72913>