



UNIVERSITÄT
DUISBURG
ESSEN

Open-Minded

Abschlusspräsentation – Masterarbeit

Konstruktion und Erprobung eines Bausteins zur Kompetenzmessung im Bereich der OOP in den Dimensionen Syntax und Semantik

Mike Barkmin ■ 14. Juni 2017

1. Einleitung
2. Forschungsstand und Forschungsfragen
3. Konzeption und Methodik
4. Durchführung und Auswertung
5. Fazit und Ausblick

Einleitung

- Beherrschen der Syntax einer Programmiersprache ist eine der ersten Herausforderungen
- ExpertInnen scheinen ein schnelleres Auffassungsvermögen von Quelltexten zu haben
- ExpertInnen scheinen auch unabhängig von der Programmiersprache die Semantik eines Quelltextes erschließen zu können



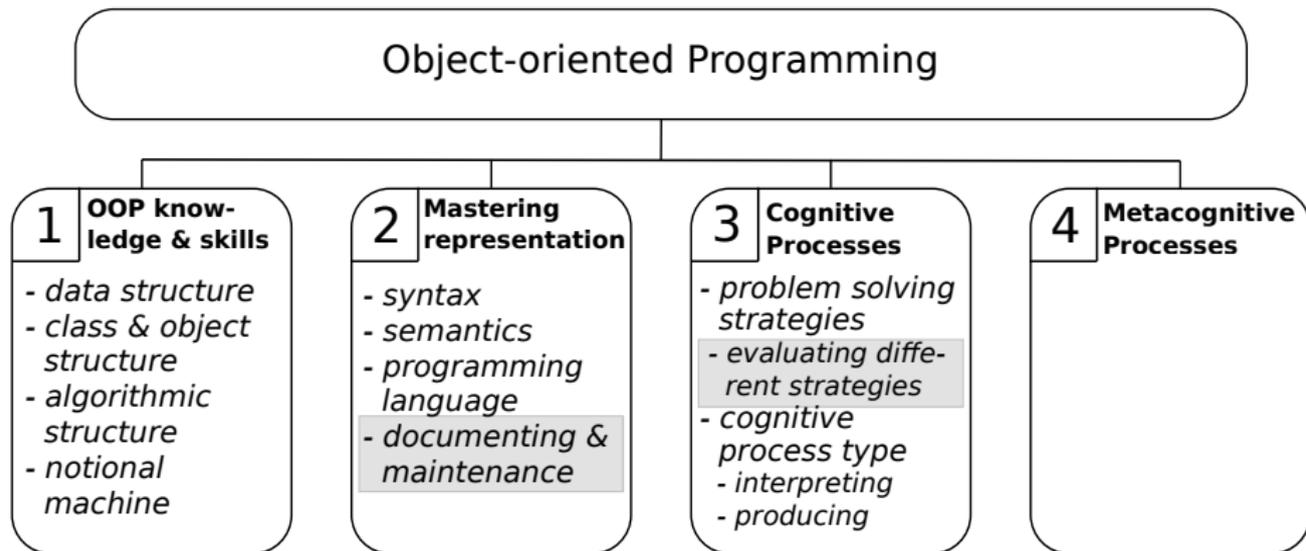
- Beherrschen der Syntax einer Programmiersprache ist eine der ersten Herausforderungen
- ExpertInnen scheinen ein schnelleres Auffassungsvermögen von Quelltexten zu haben
- ExpertInnen scheinen auch unabhängig von der Programmiersprache die Semantik eines Quelltextes erschließen zu können

⇒ Ist dies auf Kompetenzunterschiede in den Dimensionen Syntax und Semantik zurückzuführen?
Spielen andere Kompetenzen oder gar kognitionspsychologische Konstrukte eine Rolle?



Motivation und Hintergrund

- Kompetenzstrukturmodell der objektorientierten Programmierung nach Kramer, Hubwieser und Brinda (2016)



Ziele

- Entwicklung eines Bausteins zur Kompetenzmessung in den Bereichen Syntax und Semantik
- Entwicklung von Items zur Kompetenzmessung in den Bereichen Syntax und Semantik
- Herausstellen von Kennwerten zur Messung von Kompetenzunterschieden in diesen Bereichen
- Aufklären über die genauere Gestalt der Unterschiede

Forschungsstand und Forschungsfragen

Typisches Studiendesign

Merken Sie sich die folgenden Quelltextzeilen. Jede Quelltextzeile wird für 4 Sekunden eingeblendet.



```
if (vielfaches % zahl == 0)
```



```
{ return true;
```

```
boolean istVielfaches(int zahl, int vielfaches) {
```

```
} else { return false;}}
```

```
boolean istVielfaches(int zahl, int vielfaches) {  
    if(vielfaches % zahl == 0) {  
        return true;  
    } else {  
        return false;}}}
```

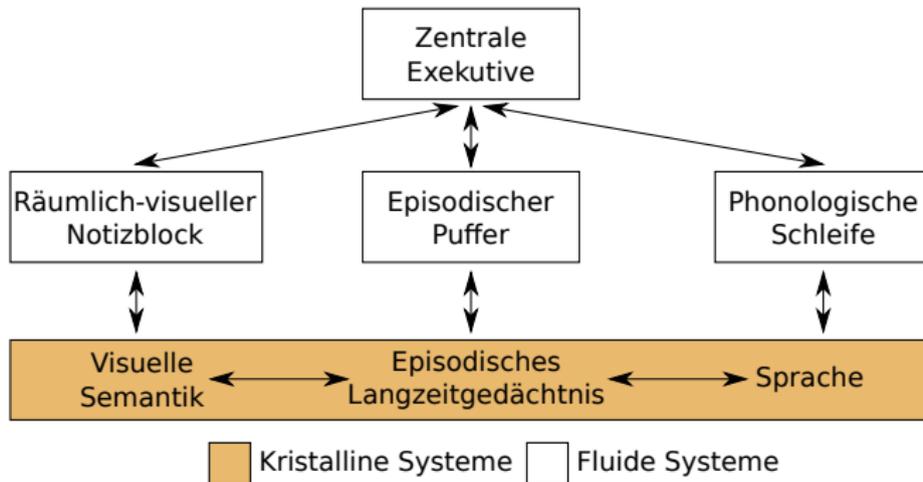
Gedächtnisfähigkeiten von ExpertInnen und NovizInnen der Programmierung

Studie	N	Merkphase	Programmiersp.
Shneiderman (1976)	79	180s	Fortran
Adelson (1981)	10	20s	PPL
Barfield (1986)	221	300s	BASIC
Schmidt (1986)	20	4s	PL/I
Bateson, Alexander und Murphy (1987)	50	180s	Fortran IV
Magliaro und Burtin (1987)	16	120s	BASIC
Guerin und Matthews (1990)	104	600s	COBOL

Tabelle: Übersicht über Studien im Bereich Programmierung zu Unterschieden zwischen ExpertInnen und NovizInnen beim Memorieren und Reproduzieren von Quelltexten



Rolle des Arbeitsgedächtnisses beim Memorieren und Reproduzieren von Quelltexten



Chunking-Theorie nach Chase und Simon (1973)

- Informationen werden zu kleinen Einheiten zusammengefasst
- Können im Langzeitgedächtnis gespeichert werden
- Es wird davon ausgegangen, dass durch mehr Wissen in einer Domäne mehr Chunks im Langzeitgedächtnis entstehen

Template-Theorie nach Gobet und Simon (1996)

- Ist eine Art Vorlage, welche befüllbare Lücken besitzt
- Lücken können mit Informationseinheiten oder Chunks befüllt werden

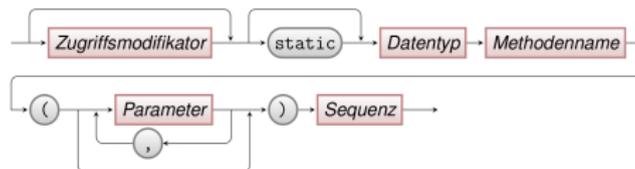


Chunking-Theorie nach Chase und Simon (1973)

- Informationen werden zu kleinen Einheiten zusammengefasst
- Können im Langzeitgedächtnis gespeichert werden
- Es wird davon ausgegangen, dass durch mehr Wissen in einer Domäne mehr Chunks im Langzeitgedächtnis entstehen

Template-Theorie nach Gobet und Simon (1996)

- Ist eine Art Vorlage, welche befüllbare Lücken besitzt
- Lücken können mit Informationseinheiten oder Chunks befüllt werden



- **F1:** Welche Unterschiede zwischen ExpertInnen und NovizInnen der Programmierung sind mit Hilfe des entwickelten Messinstruments bezüglich
 - (a) ihrer Tippgeschwindigkeit,
 - (b) ihrer Fehlerrate,
 - (c) der Gesamtlänge der Merkphasen und
 - (d) der Anzahl der Merkphasen messbar?
- **F2:** Inwiefern erklären die Konstrukte Aufmerksamkeit, Arbeitsgedächtnis und Programmiererfahrung die Fähigkeit beim Memorieren und Reproduzieren von Quelltexten?
- **F3:** Welche Formen der Strukturierung wählen ExpertInnen und NovizInnen der Programmierung beim Übertragen und Memorieren von Quelltexten?



Konzeption und Methodik

Vorstellung der Messinstrumente

- d2-Aufmerksamkeits-Belastungs-Test (Brickenkamp, 1967)
- I-S-T-2000R Merkmodule (Amthauer u. a., 2006)
- Corsi-Block-Tapping-Test (Brunetti, Gatto und Delogu, 2014; Corsi, 1972)
- Memorieren und Übertragen
- Fragebogen zur Erhebung der Programmiererfahrung (Siegmond u. a., 2014)

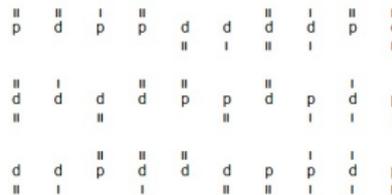


Abbildung: d2-Test

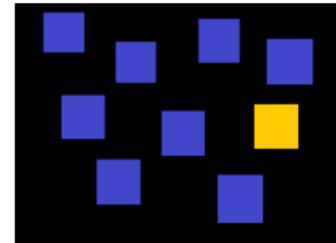


Abbildung: Corsi-Blöcke



- Drei Itemgruppen bestehend aus jeweils drei Items
- Eine Klassenstruktur, ein Algorithmus und ein natürlichsprachlicher Text
- Items innerhalb einer Gruppe vergleichbar in Bezug auf Zeichenanzahl und kognitive Komplexität
- Einordnung der Quelltexte mittels Weighted Class Complexity und Method Complexity (Misra und Akman, 2008)
- Einordnung der natürlichsprachlichen Texte mittels Lesbarkeitsindizes:
Wiener-Sachtext-Formel (Bamberger, 1984) und Flesch-Reading-Ease-DE (Amstad, 1978)

Strukturierung der Untersuchung

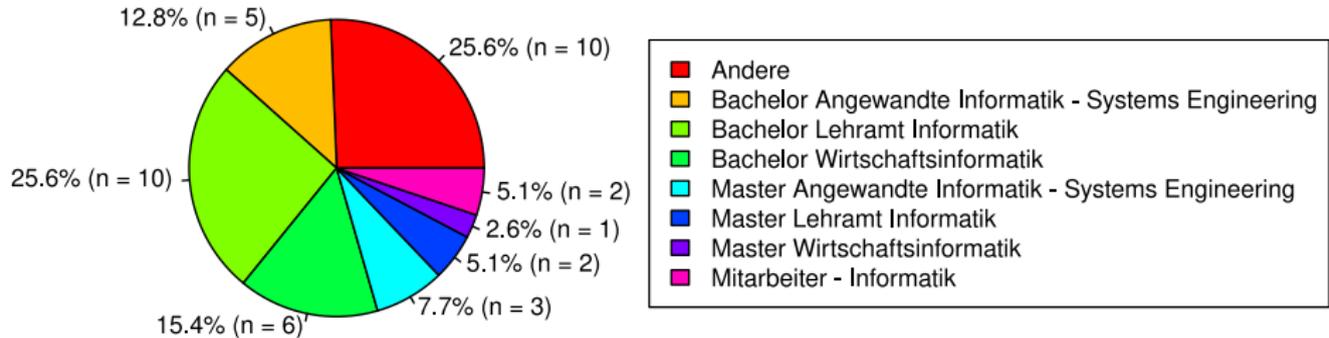
Zeit (in min)	Phase	Medium
2	Begrüßung und Präsentation des Ablaufs der Studie	Präsentation
2	Vorstellung des d2 Aufmerksamkeits-Belastungs-Test	Präsentation
2	d2 Aufmerksamkeits-Belastungs-Test: Übungszeile und Klärung von Verständnisproblemen	Testmodul 1
4:40	d2 Aufmerksamkeits-Belastungs-Test	Testmodul 1
1	Vorstellung des I-S-T 2000-R	Präsentation
1	Einprägen (verbal)	Testmodul 2
2	Reproduktion (verbal)	Testmodul 2
1	Einprägen (figural)	Testmodul 2
3	Reproduktion (figural)	Testmodul 2
2	Vorstellung des Corsi-Block-Tapping-Tests (vorwärts) und Probeitem	Präsentation

Strukturierung der Untersuchung

3	Corsi-Block-Tapping-Test (vorwärts)	Notebook
2	Vorstellung des Corsi-Block-Tapping-Tests (rückwärts) und Probeitem	Präsentation
3	Corsi-Block-Tapping-Test (rückwärts)	Notebook
<hr/>		
2	Vorstellung des Tippgeschwindigkeits-Tests und Probeitem	Präsentation, Notebook
6	Tippgeschwindigkeits-Test	Notebook
2	Vorstellung des Merk-Tests und Probeitem	Präsentation, Notebook
16	Merk-Test	Notebook
<hr/>		
5	Fragebogen	Notebook
1	Verabschiedung	
<hr/>		

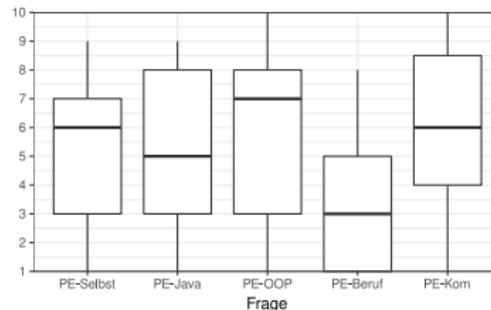
Durchführung und Auswertung

Testpersonen



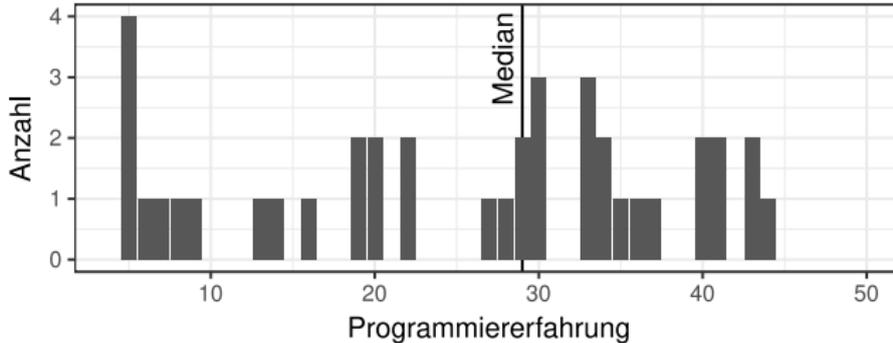
Fragen basierend auf den Ergebnissen von Siegmund u. a. (2014)

- Wie schätzen Sie Ihre Programmiererfahrung ein?
- Wie erfahren sind Sie mit der Programmiersprache Java?
- Wie erfahren sind Sie mit dem Paradigma Objektorientierte Programmierung?
- Wie schätzen Sie Ihre Programmiererfahrung verglichen mit ExpertInnen mit 20-jähriger Berufserfahrung ein?
- Wie schätzen Sie Ihre Programmiererfahrung verglichen mit Ihren KommilitonInnen ein?



Programmiererfahrung Testpersonen

- Cronbachs Alpha = 0.96
- Programmiererfahrung = Summierte Scores
- Aufteilen der Gruppe am Median (29)



Ergebnisse zur ersten Forschungsfrage

	M _N	M _E	W	p-Wert
Klasse1	1.25	0.52	245.00	0.02
Algo1	0.83	0.01	246.00	0.02
Klasse2	0.38	0.04	221.00	0.12
Algo2	0.96	0.86	184.00	0.68
Klasse3	0.81	0.70	176.00	0.87
Algo3	1.12	0.97	188.00	0.60

Tabelle: Gruppenunterschiede zwischen ExpertInnen und NovizInnen im Bezug auf die Tippgeschwindigkeitsdifferenz (gemessen in Tastenanschläge pro Sekunde)

	M _N	M _E	W	p-Wert
Klasse1	0.008	-0.003	197	0.424
Algo1	0.013	0.014	145	0.460
Klasse2	0.007	0.016	150	0.557
Algo2	0.007	-0.000	199	0.390
Klasse3	0.007	0.004	183	0.707
Algo3	-0.002	-0.015	212	0.209

Tabelle: Gruppenunterschiede zwischen ExpertInnen und NovizInnen im Bezug auf die Differenz der Fehlerraten (gemessen in Form der Anzahl gelöschter Zeichen pro Gesamtzeichenanzahl)



Ergebnisse zur ersten Forschungsfrage

	M _N	M _E	W	p-Wert
Klasse1	-6.296	2.009	73	0.00
Algo1	-7.893	3.051	42	0.00
Klasse2	0.295	8.644	116	0.10
Algo2	-12.825	-7.542	140	0.37
Klasse3	-9.603	-4.643	96	0.02
Algo3	-18.450	-8.577	120	0.13

Tabelle: Gruppenunterschiede zwischen ExpertInnen und NovizInnen im Bezug auf die zeitliche Differenz der Merkphasen (gemessen in Sekunden)

	M _N	M _E	W	p-Wert
Klasse1	-2.47	-0.40	91.00	0.015
Algo1	-3.24	0.40	61.00	0.001
Klasse2	-0.71	0.75	154.50	0.643
Algo2	-4.29	-4.10	159.00	0.748
Klasse3	-4.24	-2.40	104.00	0.045
Algo3	-8.12	-4.80	133.00	0.265

Tabelle: Gruppenunterschiede zwischen ExpertInnen und NovizInnen im Bezug auf die Differenz der Anzahl der Merkphasen

Ergebnisse zur zweiten Forschungsfrage: Strukturgleichungsmodell

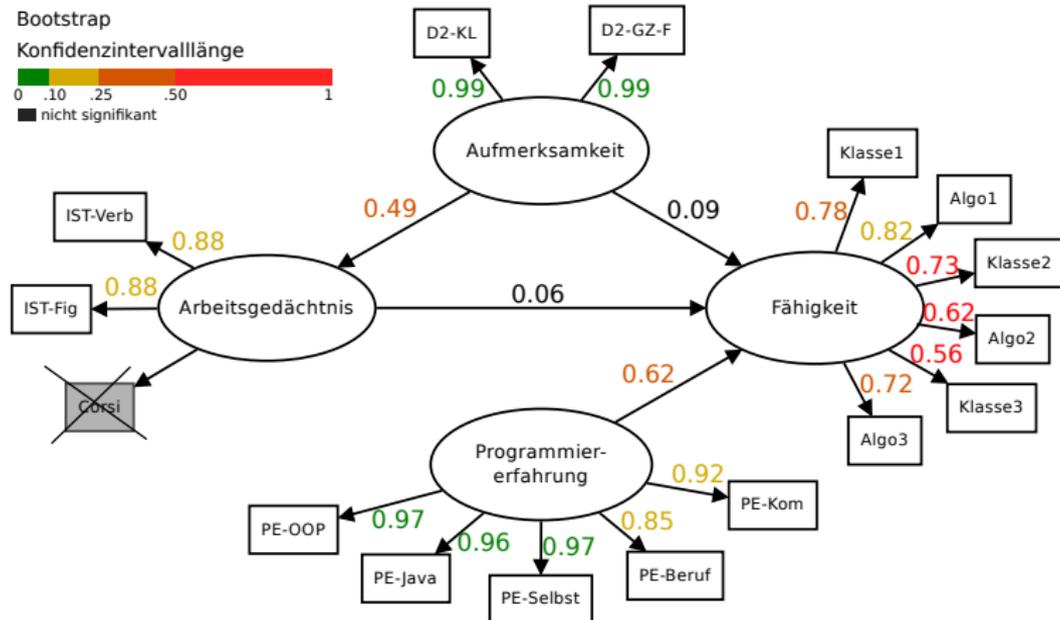


Abbildung: Strukturgleichungsmodell mit Ergebnissen der Bootstrapping-Methode (5000 zufällige Stichproben)

Ergebnisse zur zweiten Forschungsfrage: Gruppenunterschiede

	global	E	N	diff.abs	p.value	sig.05
A->AG	0.49	0.56	0.50	0.06	0.82	nein
A->F	0.06	0.02	-0.01	0.02	0.93	nein
AG->F	0.08	0.40	-0.07	0.46	0.15	nein
PE->F	0.62	0.29	0.83	0.54	0.02	ja

Tabelle: Gruppenvergleich der Pfadkoeffizienten

Ergebnisse zur dritten Forschungsfrage

Blöcke Mehrere Programmzeilen werden nach einer Merkphase zusammen reproduziert.

Ableiten Pausen werden wahrscheinlich zur inhaltlichen Reflexion genutzt. Aus semantischen Überlegungen werden anschließende Anweisungen abgeleitet.

Klammern Klammern zur Vorstrukturierung des Quelltext.

Konventionen Es werden Entscheidungen auf Grundlage von Konventionen getroffen.

Ergebnisse zur dritten Forschungsfrage: Blöcke

Phase	Zeit in s	LS	G	Inhalt	
1	M	4.12			
2	S	12.23	-56	1	<code>public class Haus{ private int nummer private String farbe</code>
		12.23 (0.00)	-56	1	
3	M	1.41			
4	S	4.52	-1	1	<code>public class Haus{ private int nummer private sString farbe;</code>
		4.52 (0.00)	-1	1	
5	M	3.78			
6	S	14.02	-51	2	<code>public class Haus{ private int nummer private String farbe; public void streiche (String farbr){ this.farbe=farbe;</code>
		14.02 (0.00)	-51	2	

Ergebnisse zur dritten Forschungsfrage: Blöcke

Phase	Zeit in s	LS	G	Inhalt	
1	M	3.90			
2	S	13.44	-47	0	<code>boolean istVielfaches(int zahl, int vielfaches){</code>
		13.44 (0.00)	-47	0	
3	M	6.73			
4	S	17.62	-54	0	<code>boolean istVielfaches(int zahl, int vielfaches){ if (vielfaches % zahl == 0) return true; else {</code>
		17.62 (0.00)	-54	0	<code>return false;</code>

Ergebnisse zur dritten Forschungsfrage: Ableiten

Phase	Zeit in s	LS	G	Inhalt
5 M	2.50			
6 S	5.71	-33	0	<pre> public class Haus { private int nummer; private String </pre>
7 P	2.43			
8 S	3.01	-6	1	<pre> public class Haus { private int nummer; private String StraÙe; </pre>
9 P	2.43			
8.72 (4.86)		-39	1	
10 M	2.15			
11 P	2.19			
12 S	5.67	0	8	<pre> public class Haus { private int nummer; private String StfaraÙbe; </pre>

Ergebnisse zur dritten Forschungsfrage: Ableiten

Phase	Zeit in s	LS	G	Inhalt
12	M	0.94		
13	S	24.52	-32	3
				<pre> public class Haus{ private String farbe; private int nummer; public void Streiche(){ this.farbe = farbe; } </pre>
14	P	2.21		
15	S	1.65	-11	0
				<pre> public class Haus{ private String farbe; private int nummer; public void Streiche(String farbe){ this.farbe = farbe; } </pre>
16	P	2.66		



Ergebnisse zur dritten Forschungsfrage: Ableiten

Phase	Zeit in s	LS	G	Inhalt
1	M	14.68		
2	S	17.21	-50	1
				<pre>boolean istVielfaches (int zahl, int ↪ vielfaches){ if(</pre>
3	P	5.09		
4	S	17.71	-48	1
				<pre>boolean istVielfaches (int zahl, int ↪ vielfaches){ if(vielfaches%zahl==0) return true; else return false;</pre>

- **F1:** Welche Unterschiede zwischen ExpertInnen und NovizInnen der Programmierung sind mit Hilfe des entwickelten Messinstruments bezüglich
 - (a) ihrer Tippgeschwindigkeit,
 - (b) ihrer Fehlerrate,
 - (c) der Gesamtlänge der Merkphasen und
 - (d) der Anzahl der Merkphasen messbar?
- **F2:** Inwiefern erklären die Konstrukte Aufmerksamkeit, Arbeitsgedächtnis und Programmiererfahrung die Fähigkeit beim Memorieren und Reproduzieren von Quelltexten?
- **F3:** Welche Formen der Strukturierung wählen ExpertInnen und NovizInnen der Programmierung beim Übertragen und Memorieren von Quelltexten?



Fazit und Ausblick

Fazit und Ausblick

- Baustein hat sich als Messinstrument bewährt
- Verwendete Items müssen überarbeitet werden
- Erhebung der selbsteingeschätzten Programmiererfahrung möglicherweise nicht ideal
- Warum sind in den kognitiv komplexeren und längeren Items keine Unterschiede erkennbar? Variieren und Testen.
- Wie werden die Quelltexte wirklich im Arbeitsgedächtnis strukturiert? Lautes Denken.
- Können sich die Testpersonen an die übertragenen Quelltexte erinnern?
- Sind andere Varianten des Messinstruments besser geeignet, um Kompetenzunterschiede zu messen? Variation durch zeitliche Beschränkung oder Beschränkung der Anschauphasen.
- Können ExpertInnen der Programmiersprache Java auch Quelltext einer anderen unbekanntem objektorientierten Programmiersprache memorieren? Mögliche Hinweise auf eine Meta-Kompetenz



Vielen Dank für die Aufmerksamkeit!

Kontakt

Mike Barkmin
Didaktik der Informatik
Universität Duisburg-Essen

mike.barkmin@stud.uni-due.de

- Adelson, B. (1981). „Problem solving and the development of abstract categories in programming languages“. In: *Memory & Cognition* 9.4, S. 422–433.
- Amstad, T. (1978). *Wie verständlich sind unsere Zeitungen?*. Abhandlung: Philosophische Fakultät I. Zürich. 1977. Studenten-Schreib-Service.
- Amthauer, R. u. a. (2006). *Intelligenz-Struktur-Test 2000 R*. Bd. 2. Göttingen: Hogrefe.
- Bamberger, R. (1984). *Lesen - verstehen - lernen - schreiben : die Schwierigkeitsstufen von Texten in deutscher Sprache*. ger. Wien: Jugend und Volk [u.a.] ISBN: 3224152508.
- Barfield, W. (1986). „Expert-novice differences for software: implications for problem-solving and knowledge acquisition“. In: *Behaviour & Information Technology* 5.1, S. 15–29.

- Bateson, A. G., R. A. Alexander und M. D. Murphy (1987). „Cognitive processing differences between novice and expert computer programmers“. In: *International Journal of Man-Machine Studies* 26.6, S. 649–660.
- Björnsson, C. H. (1968). *Läsbarhet*. Stockholm: Liber.
- Brickenkamp, R. (1967). *Test d2: Aufmerksamkeits-Belastungs-Test*. Bd. 2. Göttingen: Hogrefe.
- Brunetti, R., C. Del Gatto und F. Delogu (2014). „eCorsi: Implementation and testing of the Corsi block-tapping task for digital tablets“. In: *Frontiers in Psychology* 5.
- Chase, W. G. und H. A. Simon (1973). „Perception in chess“. In: *Cognitive Psychology* 4.1, S. 55–81.
- Corsi, P. M. (1972). *Human Memory and the Medial Temporal Region of the Brain*. McGill theses.
- Gobet, F. und H. A. Simon (1996). „Recall of rapidly presented random chess positions is a function of skill“. In: *Psychonomic Bulletin and Review* 3.2. cited By 98, S. 159–163.

Literaturverzeichnis

- Guerin, B. und A. Matthews (1990). „The Effects of Semantic Complexity on Expert and Novice Computer Program Recall and Comprehension“. In: *The Journal of General Psychology* 117.4. PMID: 28142341, S. 379–389.
- Kellogg, R. T. (1996). „A model of working memory in writing“. In: Torrance, Mark und Gaynor C Jeffery. *The Cognitive demands of writing : processing capacity and working memory in text production*. Amsterdam University Press, S. 57–71.
- Kramer, M., P. Hubwieser und T. Brinda (2016). „A Competency Structure Model of Object-Oriented Programming“. In: *2016 International Conference on Learning and Teaching in Computing and Engineering (LaTICE)*, S. 1–8.
- Magliaro, S. und J. K. Burtin (1987). „Adolescents' Chunking of Computer Programs“. In: *Comput. Sch.* 4.3-4, S. 129–138.



- Misra, S. und K. I. Akman (2008). „Weighted Class Complexity: A Measure of Complexity for Object Oriented System“. In: *Journal of Information Science and Engineering*.
- Schmidt, A. L. (1986). „Effects of experience and comprehension on reading time and memory for computer programs“. In: *International Journal of Man-Machine Studies* 25.4, S. 399–409.
- Shneiderman, B. (1976). „Exploratory experiments in programmer behavior“. In: *International Journal of Computer & Information Sciences* 5.2, S. 123–143.
- Siegmund, J. u. a. (2014). „Measuring and Modeling Programming Experience“. In: *Empirical Softw. Engg.* 19.5, S. 1299–1334.

Die Rolle des Arbeitsgedächtnisses beim Übertragen von Texten

Basic Process	Spatial	Central Executive	Verbal
Planning	X	X	
Translating		X	X
Programming		X*	
Executing			
Reading		X	X
Editing	X		

Tabelle: Kellogg (1996): Model of the Role of Working Memory in Writing

Note: X indicates use of component.

X*. For highly practiced motor skills, these demands are small if not negligible.

Lesbarkeitsindizes

■ Flesch-Reading-Ease-DE nach Amstad (1978)

$$\text{FRE}_{DE} = 180 \cdot dSI - dSa \cdot 58.5$$

■ dSI: durchschnittliche Satzlänge, dSa: durchschnittliche Silbenanzahl

■ über 80 anspruchslos, 71 - 80 sehr einfach, 61 - 70 einfach, 46 - 60 durchschnittlich, 36 - 45 schwierig, unter 35 sehr schwierig

■ Wiener-Sachtext-Formel nach Bamberger (1984)

$$\text{WST} = 0.1935 \cdot MS + 0.1672 - SI + 0.1297 \cdot IW - 0.0327 \cdot ES - 0.875$$

■ MS: Prozentanteil der Wörter mit drei oder mehr Silben, SI: mittlere Satzlänge in Wörtern, IW: Prozentanteil der Wörter mit mehr als sechs Buchstaben, ES: Prozentanteil der einsilbigen Wörter

■ Angabe in Schulstufen von 4 bis 15.

■ LIX nach Björnsson (1968)

$$\text{LIX} = A/B \cdot C/A \cdot 100$$

■ A: Anzahl der Wörter, B: Anzahl der Punkte, Kommata, Semikolons und Großbuchstaben, C: Anzahl der Wörter mit sechs oder mehr Zeichen

■ unter 40: Kinder- und Jugendliteratur, 40 bis 50: Belletristik, 50 bis 60: Sachliteratur, über 60: Fachliteratur



Abschreiben

Zeichen: 290, FRE_{DE}: 48, WST: 10

Frühling ist eine deutsche Filmreihe, die seit Anfang 2010 produziert wird. Durch ihr Talent, den Menschen zuzuhören, wird die Dorfhelferin dabei mehr zu einer Art Sozialarbeiterin, die es versteht, den Frühling in Familien zu tragen, die einen Schicksalsschlag erleben oder erlitten haben.

Zeichen: 292, TWCC: 9

```
public int euklid(int zahl1, int zahl2) {
    if (zahl1 == 0) {
        return zahl2;
    } else {
        while (zahl2 != 0) {
            if (zahl1 > zahl2) {
                zahl1 -= zahl2;
            } else {
                zahl2 -= zahl1;
            }
        }
    }
    return zahl1;
}
```



Übertragen 1

Zeichen: 147, TWCC: 3

```
public class Haus {
    private int nummer;
    private String farbe;

    public void streiche(
        ↪ String farbe) {
        this.farbe = farbe;
    }
}
```

Zeichen: 149, FRE_{DE}: 79,

WST: 5

Lettland ist ein Staat im Osten von Europa. Er liegt an der Ostsee und gehört zu den baltischen Staaten. Die beiden anderen sind Estland und Litauen.

Zeichen: 146, TWCC: 2

```
boolean istVielfaches(int
    ↪ zahl, int
    ↪ vielfaches) {
    if (vielfaches % zahl == 0)
        ↪ {
        return true;
    } else {
        return false;
    }
}
```

Übertragen 2

Zeichen: 219, TWCC: 5

```
public String einruecken(
    ↪ String text, int
    ↪ anzahl, String
    ↪ symbol) {
    if(breite < 0) {
        return null;
    }
    for(int index = anzahl;
        ↪ index > 0; index
        ↪ —) {
        text = symbol + text;
    }
    return text;
}
```

Zeichen: 206, FRE_{DE}: 69,

WST: 8

Meerschweinchen heißen „Schweinchen“, weil sie wie Schweine quieken. „Meer“ kommt daher, dass sie von Südamerika aus über den Atlantik nach Europa gebracht worden sind. In Europa sind sie beliebte Haustiere.

Zeichen: 214, TWCC: 4

```
class Datei {
    String name;
    String pfad;

    Datei(String name, String
        ↪ pfad) {
        this.name = name;
        this.pfad = pfad;
    }

    void umbenennen(String
        ↪ name) {
        this.name = name;
    }
}
```



Übertragen 3

Zeichen: 237, TWCC: 7

```

public class Dreieck
    ↪ extends Vieleck
    ↪ {
public float a, b, c = 1;

public Dreieck() {
super(3);
}

public class Vieleck {
private int ecken;

public Vieleck(int ecken)
    ↪ {
this.ecken = ecken;
}
}

```

Zeichen: 221, FRE_{DE}: 49,

WST: 10

Allerheiligen ist ein Fest. Möglicherweise haben die Christen im frühen Mittelalter das Fest von Nichtchristen übernommen. Viele Menschen hatten das Bedürfnis, wenigstens einmal im Jahr gemeinsam an Verstorbene zu denken.

Zeichen: 244, TWCC: 8

```

void istPrimzahl(int zahl)
    ↪ {
boolean primzahl = true;
for(int p = 2; p < zahl; p
    ↪ ++){
if((i % p) == 0) {
primzahl = false;
break;
}
}
if(primzahl) {
System.out.println(i);
}
}

```

