



hochschule aschaffenburg
university of applied sciences



HOCHSCHULE COBURG



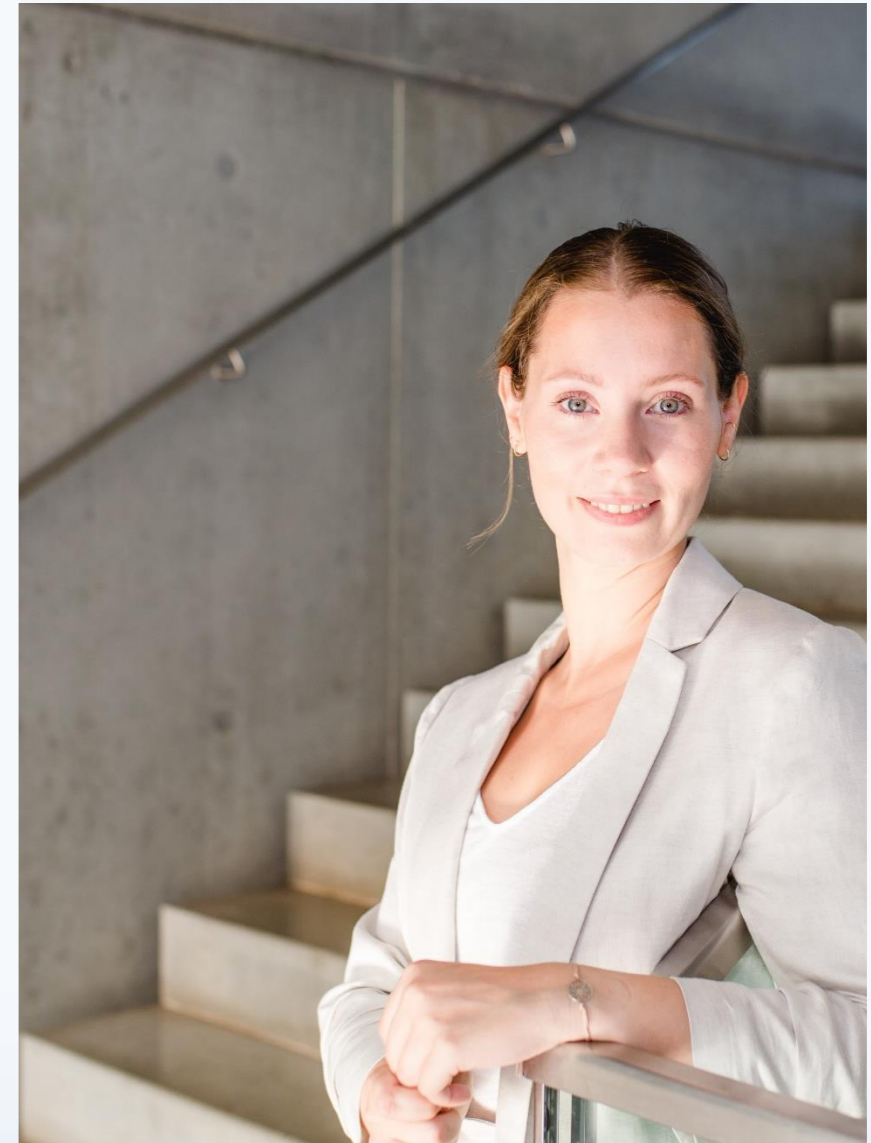
A SYSTEMATIC LITERATURE REVIEW ON MISCONCEPTIONS IN SOFTWARE ENGINEERING

Carolin Gold-Veerkamp
University of Applied Sciences Aschaffenburg
Nermin Saray
University of Applied Sciences Coburg

Presenter

Carolin Gold-Veerkamp

- Research associate in the project EVELIN (funded by the BMBF – German Federal Ministry of Education and Research) at the University of Applied Sciences Aschaffenburg in Germany as a PhD candidate
- studied Industrial Engineering (M.Sc.) at the Mannheim University of Applied Sciences, Germany (2012 - 2014)
- Mechatronics (B.Eng.) at the University of Applied Sciences in Aschaffenburg, Germany (2008 - 2012)



Agenda

1. Introduction – What is a misconception?

2. Context

3. Systematic Literature Review (SLR)

a. Phase 1: Initial Planning

b. Phase 2: Examination Process

c. Phase 3: Results

i. Title & Abstract

ii. Full Text

iii. Misconceptions Found

4. Conclusion

Introduction – What is a „Misconception“?

is an individual conception that is "at odds with modern scientific theories" [1, p. 2]

the wording implies "the wrongness of a student's conception, and can thus be seen as critical of the holder of the concept" [1, p. 2]

On the basis of various concerns numerous „synonyms“ have developed:

- Preconceptions [1, p. 1f.; 4, p. 99; 5, p. 66; 6; 7, p. 61]
- Students' conceptions [1, p. 1f.; 3, p. 119]
- Alternative conceptions [4, p. 99; 8]
- Naïve conceptions [1, p. 1f.]
- Beliefs [9, p. 15]
- Naïve beliefs [3, p. 119]
- Alternative beliefs [3, p. 119; 6]
- Alternative frameworks [1, p. 1f.; 10, p. 7]
- Naïve theories [1, p. 1f.; 11]
- The „standard term“ as named by Smith et al. [3, p. 119] and others: Misconceptions [1, p. 1f.; 4, p. 99; 7, p. 61; 9, p. 15; 12, p. 325; 13, p. 60].

Context

- Constructivism: Learning is to be understood as an active, individual, situated, social, and cognitive psychological process
- Each individual builds own knowledge by combining new concepts based on previous knowledge
- This means, learners form conceptions and models to explain phenomena, processes, and artifacts before confronted with them in institutional learning
- These possibly alternate from scientific or expert perspectives
- Twofold significant impact on the learning process:
 - On the one hand, they can **serve as the basis for learning**
 - On the other, they can also **contradict the educational content** and thus hinder the learning process
- Therefore: Didactics should do justice to the learners' "points of departure" [1, p. 6]

➤ Question:

Which misconceptions in Software Engineering are already known?

SLR-Phase 1: Initial Planning

determine parameters that require previous definition in order to minimize bias in the SLR

1) Research Question(s):

To what extent does research on misconceptions in SE already exist?

Which misconceptions in SE are known/documentated?

2) Search Strategy:

- a) Language Selection: German and English
- b) Queries and Synonyms: (English Search Query covering a broad variety of synonyms of the term “misconception” combined with Software Engineering)

```

("software engineer*" OR "software development*" OR "software
process*")
AND
("preconception*" OR "conception*" OR "belief*" OR "misconception*"
OR
"naïve theor*" OR "alternative framework*" OR
"intuitive theor*" OR "intuitive science" OR "prior knowledge")
    
```

- c) Database: Electronic literature databases [28-37] are selected

SELECTION OF ELECTRONIC DATABASES FOR SLR BASED ON [25] [26]⁹

Source	IEEE	ACM	SD	SC	SL	ERIC	WoS	GS	arXiv	dblp
Information and Software Technology			X	X				X		
Journal of Systems and Software			X	X				X		
IEEE Transactions on SE	X							X		
IEEE Software	X							X		
Communications of the ACM		X								
ACM Computer Surveys		X								
ACM Transactions on SE		X								
Methodologies										
Software Practice and Experience								X		
Empirical SE Journal					X					
IEEE Proc. Software (now: IET	X							X		
Software)										
Proc. Int. Conference on SE	X	X						X		
Proc. Int. Symp. of Software Metrics	X	X						X		
Proc. Int. Symp. on Empirical SE	X	X						X		



SLR-Phase 1: Initial Planning

3) Selection Strategy: Controlled by predefined Inclusion (IC) and Exclusion Criteria (EC)

IC.1 The publication is written in **English or German** language.

IC.2 It is explicitly about the discipline **Software Engineering**.

IC.3 **Misconceptions** in SE are explicitly mentioned.

EC.1 The contribution is an abstract, workshop, or similar.

4) Quality Assessment: Determined Quality Criteria (QC)

QC.1 **Traceability:** How do the authors know this misconception?

QC.2 **Validation:** Has it been confirmed that it is a MISconception?

How did the authors validate the conception to be “at odds with modern scientific theories” [4, p. 2]?

QC.3 **Occurrence in the population:** Does this misconception exist in the population?

Did the authors test the misconception in a specific target group?

SLR-Phase 2: Examination Process

1) Stage 1: Conducting the Automated Search

- If possible, use of **extended/advanced search functions**, wildcards (e.g., “misconception*”), and Boolean operators was made
- Search is limited to **document title and abstract**, as recommended by [27]

2) Stage 2: Applying the In-/Exclusion Criteria

- **Relevance of publication** is determined in a two-stage process:
 - i. **Title and abstract** are examined and evaluated on the basis of the predefined criteria.
 - ii. Papers included are then rechecked regarding the in-/exclusion criteria; this time considering the **full text**

3) Stage 3: Backward Snowballing

- “the references of the selected papers [are] reviewed and any missing candidate papers [are] assessed against the inclusion/exclusion criteria” [27, p. 2052] as well; this is referred to as ‘**backward snowballing**’

4) Stage 4: Data Analysis

- Assessment of quality using the predefined **quality criteria**

SLR-Phase 3: Results (Title & Abstract)

- Automated search completed between **April, 30th and May, 1st 2020**
- Search was not limited to a date range (review process timewise included every research found, covering papers as of 1970)
- Number of search matches: **n = 2,158**
- Excluding data sets that contained entire proceedings/compilations instead of contributions as well as duplicates: **n = 1,481**
- After applying the inclusion and exclusion criteria to title and abstract: **n = 128**

SUMMARY OF SLR RESULTS AFTER APPLYING IN-/EXCLUSION CRITERIA ON TITLE & ABSTRACT

	Search Engines										Sum
	IEEE	ACM	SD	SC	SL	ERIC	WoS	GS	arXiv	dblp	
Results of English SQ	250	410	93	847	0	29	4	87	257	41	2,018
Results of German SQ	16	54	7	46	0	3	0	7	2	5	140
Sum of Search Results	266	464	100	893	0	32	4	94	259	46	2,158
No Papers (e.g. Proc.)	2	2	0	53	0	2	0	34	6	0	99
Duplicates	15	85	18	383	0	10	4	16	18	29	578
Balance without Duplicates	249	377	82	457	0	20	0	44	235	17	1,481
IC.1a: English	249	352	81	442	0	20	0	40	231	10	1,425
IC.1b: German	0	0	0	5	0	0	0	1	0	0	6
IC.2: SE Discipline-Specific	223	253	65	381	0	18	0	34	162	7	1,143
IC.3: Misconceptions ⁹	30	60	4	43	0	8	0	6	5	2	158
EC.1: Contribution Type	0	1	0	0	0	1	0	0	0	0	2
EC: No Information	0	0	0	1	0	0	0	2	0	0	3
Paper Candidates	29	40	4	37	0	6	0	5	5	2	128

SLR-Phase 3: Results (Full Text Search)

- Inclusion and exclusion criteria are then applied to the paper candidates based on the full text of the contributions: **n = 15**
- Papers are excluded that cover the topic ‘misconception’, but did not explicitly mention at least one statement
- The subsequent backward snowball search (checked against the IC and ECs): **n = 18**

SUMMARY OF SLR RESULTS AFTER APPLYING IN-/EXCLUSION CRITERIA ON FULL TEXTS

	Search Engines										Sum
	IEEE	ACM	SD	SC	SL	ERIC	WoS	GS	arXiv	dblp	
Paper Candidates (see Table III)	29	40	4	37	0	6	0	5	5	2	128
IC.1a: English	29	40	4	37	0	6	0	5	5	2	128
IC.1b: German	0	0	0	0	0	0	0	0	0	0	0
IC.2: SE Discipline	29	40	4	37	0	6	0	4	5	2	127
IC.3: Mention Misconceptions	5	3	1	2	0	0	0	1	3	0	15
Papers Found	5	3	1	2	0	0	0	1	3	0	15
Backward Snowballing	27	5	0	2	0	0	0	1	4	0	39
Already Included in SLR	2	1	0	1	0	0	0	0	2	0	6
After Applying Selection Criteria	0	0	0	1	0	0	0	0	2	0	3
Result	5	3	1	3	0	0	0	1	5	0	18

Note: Through the selection process in Stage 2 and Backward Snowballing in Stage 3 as a whole, we double-checked the contributions by assessing each paper. As Kitchenham et al. suggest, publications are included if we cannot make a consensual decision [27, p. 2052].

SLR-Phase 3: Results (Full Text Search)

- The misconceptions named in the papers (n = 18) are then checked against the QCs
- Misconceptions sufficiently meeting the QCs singularly are marked gray in the following Table

SUMMARY OF MISCONCEPTIONS FOUND IN THE FULL TEXTS USING THE QUALITY CRITERIA

	Papers Found																	Sum
	[38]	[39]	[40]	[41]	[42]	[43]	[44]	[45]	[46]	[47]	[48]	[49]	[50]	[51]	[53]	[54]	[55]	
Misconceptions explicitly named	16	12	12	7	6	4	5	12	4	7	7	10	36	4	10	21	4	167
QC.1 Traceability																		
- Study						4							36		21 (in [54])			51
- Reference(s)	15	6			6													37
- No Indication	1	6	12	7			5	12	4	7	7	10		4			4	79
QC.2 Validation																		
- Empirically Confirmed								12							8 (in [53])			20
- Empirically Rejected															2 (in [53])			2
- Reference(s)	6	6		5														17
- Only based on Explanation						4			4	7	7	10		4				36
- No Indication	10	6	12	2	6	4	5						36		11		4	92
QC.3 Occurrence																		
- Practitioners	16				6	4									21 (in [54])			37
- Undergraduates		12	12					12										36
- No Indication				7			5		4	7	7	10	36	4	10/21 (in [53])		4	94
Intersection (of rows marked)																		
								12							8			20

- The intersection of the QCs results in **n = 20** misconceptions found in the papers [45, 53, 54]
- Papers [54] and [53] only deal with the topic 'defect prediction'
- The authors of [45] look at SE covering the **software life cycle** more holistically; see thematic structuring on the next slide

Note: [45] would actually not be included in the intersection, as it is not explained where the misconceptions come from (QC.1). But the authors validated them (QC.2) and tested their occurrence concerning students (QC.3). Thus, the misconceptions listed are hypotheses, that have been empirically confirmed; thus, nevertheless, they are included in the intersection.

SLR-Phase 3: Misconceptions Found

LIST OF MISCONCEPTIONS MATCHING THE QUALITY CRITERIA

Topic(s) ¹¹								
P	PM	TS	R	I	De	Do	Misconception	Reference(s)
X							A defined software process is only important when you are working with people who are less skilled.	[45, (1)]
	X						A good software developer will often choose to work alone on a project in order to get it done faster.	[45, (2)]
X	X						When you have a team of good programmers who work well together, a software process will usually get in the way.	[45, (3)]
			X				My code should take advantage of the implementation details in other code.	[45, (4)]
		X					It is expected that clients will describe their requirements accurately before a team begins programming.	[45, (5)]
			X				As a software developer, most of my time will be spent designing and implementing new algorithms and data structures.	[45, (6)]
X							Most of the time when I start a new programming task in industry, I will be working on a new project.	[45, (7)]
	X						Developers do not need to know the high-level context of the system; this allows them to concentrate on their task.	[45, (8)]
				X			A software project is successful only if it ships with very few known defects.	[45, (9)]
					X		Software engineering is about producing lots of documentation on the requirements and implementation of the project.	[45, (10)]
X	X	X					Process, requirements, and team-management are important to business majors, not software developers.	[45, (11)]
			X				The majority of the cost of a successful software project will be the initial implementation effort.	[45, (12)]
				X			A file with a complex code change process tends to be buggy.	[53, (B1)], [54, (S2)]
				X			A file that is changed by more developers is more bug-prone.	[53, (B2)], [54, (S14)]
				X			A file with more added lines is more bug-prone.	[53, (B3)], [54, (S4)]
				X			Recently changed files tend to be buggy.	[53, (B4)], [54, (S7)]
				X			Recently bug-fixed files tend to be buggy.	[53, (B6)], [54, (S10)]
				X			A file with more fixed bugs tends to be more bug-prone.	[53, (B7)], [54, (S11)]
				X			A file with more commits is more bug-prone.	[53, (B8)], [54, (S12)]
				X			A file with more removed lines is more bug-prone.	[53, (B9)], [54, (S13)]

Thematical Structuring

P = Project

PM = Process Models

TS = Team Skills

R = Requirements

I = Implementation

De = Defects

Do = Documentation

Conclusion

- Paper's purpose: **Identify and analyse known misconceptions in SE**
 - Overall goal: To use these insights in higher education
 - Predefined search queries have been applied to search **10 databases**
 - Out of **2,158 publications**, 18 could be identified as appropriate for the selection criteria.
 - These contain **167 statements**, which the authors of these papers refer to as misconceptions.
 - **20** of them met the quality criteria specified; i.e. only 3 publications cover valuable data.
 - To conclude, the results show...
 - ... currently evidence-based research on misconceptions in SE is limited
 - ... there is not enough research on evidence-based misconceptions in SE to use these insights
- **So, in addition a primary study to identify misconceptions in SE is indispensable before addressing them.**



**Thank you very much
for your attention.**

PROJEKT EVELIN
Experimentelle Verbesserung
des Lernens von Software Engineering

GEFÖRDERT VOM



Bundesministerium
für Bildung
und Forschung



References

- [1] R. Duit, "Science education research internationally: Conceptions, research methods, domains of research," EURASIA Journal of Mathematics, Science & Technology Education, vol. 3, no. 1, pp. 3–15, 2007.
- [2] R. Duit, H. Gropengießer, U. Kattmann, M. Komorek, and I. Parchmann, "The model of educational reconstruction," in Science Education Research and Practice, D. Jorde and J. Dillon, Eds. Springer, 2012, pp. 13–37.
- [3] J. P. Smith, A. A. diSessa, and J. Roschelle, "Misconceptions reconceived: A constructivist analysis of knowledge in transition," Journal of the Learning Sciences, vol. 3, no. 2, pp. 115–163, 1994.
- [4] J. R. Read, "Children's misconceptions and conceptual change in science education," 2004, [retrieved: 09, 2020]. [Online]. Available: <http://acell.chem.usyd.edu.au/Conceptual-Change.cfm>
- [5] I. Diethelm and S. Zumbrägel, "An investigation of secondary school students' conceptions on how the internet works," in Koli Calling International Conference on Computing Education Research, M.-J. Laakso and R. McCartney, Eds. ACM Press, 2012, pp. 67–73.
- [6] S. Todtenhaupt, To develop an understanding of chemistry in schoolchildren: An investigation into the redox topic at a high school. (Original title: "Zur Entwicklung des Chemieverständnisses bei Schülern: Eine Untersuchung zur Redox-Thematik an einem Gymnasium" [German]). Frankfurt a.M.: Lang, 1995.
- [7] A. Taylor Kujawski, and P. Kowalski, "Naive psychological science: the prevalence, strength, and source of misconceptions," The Psychological Record, vol. 54, pp. 15–25, 2004.
- [8] R. Reuter, F. Hauser, C. Gold-Veerkamp, J. Mottok, and J. Abke, "Towards a definition and identification of learning obstacles in higher software engineering education," in Annual International Conference on Education and New Learning Technologies (EDULEARN), IATED, Ed., 2017, pp. 10 259–10 267.
- [9] R. Duit, "STCSE: Students' and teachers' conceptions and science education," Bibliography, 2009, [retrieved: 09, 2020]. [Online]. Available: http://archiv.ipn.uni-kiel.de/stcse/download_stcse.html
- [10] D. N. Perkins and R. Simmons, "Patterns of misunderstanding: An integrative model of misconceptions in science, math and programming," in 2. Int. Seminar Misconceptions and Educational Strategies in Science and Mathematics: Vol. I, J. D. Novak, Ed., 1987, pp. 381–395.
- [11] —, "Patterns of misunderstanding: An integrative model for science, math, and programming," Review of Educational Research, vol. 58, no. 3, pp. 303–326, 1988.
- [12] L. Louca and Z. C. Zacharia, "The use of computer-based programming environments as computer modelling tools in early science education: The cases of textual and graphical program languages," International Journal of Science Education, vol. 30, no. 3, pp. 287–324, 2008.

References

- [13] J. Confrey, “Misconceptions across subject matter: science, mathematics, programming,” in 2. Int. Seminar Misconceptions and Educational Strategies in Science and Mathematics: Vol. I, J. D. Novak, Ed., 1987, pp. 81–106.
- [14] N. Taylor and G. Corrigan, “New South Wales primary school teachers’ perceptions of the role of ICT in the primary science curriculum – A rural and regional perspective,” *International Journal of Science and Mathematics Education*, vol. 5, no. 1, pp. 85–109, 2007.
- [15] R. D. Pea, “Language-independent conceptual “bugs” in novice programming,” *Journal educational computing research*, vol. 2, no. 1, pp. 25–36, 1986.
- [16] J. Sorva, “Visual program simulation in introductory programming education.” Dissertation, Espoo, Aalto Univ., Finland, 2012.
- [17] A. Swidan, F. Hermans, and M. Smit, “Programming misconceptions for school students,” in *Conference on International Computing Education Research (ICER)*. ACM, 2018, pp. 151–159.
- [18] Z. Zanko, M. Mladenovic, and I. Boljat, “Misconceptions about variables at the K-12 level,” *Education and Information Technologies*, vol. 24, no. 2, pp. 1251–1268, 2019.
- [19] R. Kelter, M. Kramer, and T. Brinda, “Statistical frequency-analysis of misconceptions in object-oriented-programming: Regularized PCR models for frequency analysis across oop concepts and related factors,” in *Koli Calling International Conference on Computing Education Research*, M. Joy and P. Ihantola, Eds. ACM, 2018, pp. 6:1–6:10.
- [20] S. Holland, R. Griffiths, and M. Woodman, “Avoiding object misconceptions,” in *SIGCSE technical symposium on Computer science education*, C. M. White, C. Erickson, B. Klein, and J. E. Miller, Eds. ACM, 1997, pp. 131–134.
- [21] M. T. Rucker and N. Pinkwart, ““How else should it work?” A grounded theory of pre-college students’ understanding of computing devices,” *ACM Transactions on Computing Education*, vol. 19, no. 1, pp. 2:1–2:23, 2018.
- [22] I. Diethelm, P. Hubwieser, and R. Klaus, “Students, teachers and phenomena: Educational reconstruction for computer science education,” in *Koli Calling International Conference on Computing Education Research*, M.-J. Laakso and R. McCartney, Eds. ACM, 2012, pp. 164–173.
- [23] B. A. Kitchenham and S. Charters, “Guidelines for performing systematic literature reviews in software engineering: Version 2.3,” *EBSE Technical Report (EBSE-2007-01)*, Keele University & University of Durham, 2007.
- [24] P. O. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, “Lessons from applying the systematic literature review process within the software engineering domain,” *Journal of Systems and Software*, vol. 80, no. 4, pp. 571–583, 2007.
- [25] B. A. Kitchenham, et al., “Systematic literature reviews in software engineering – a systematic literature review,” *Information and Software Technology*, vol. 51, no. 1, pp. 7–15, 2009.
- [26] A. Bartel, “Conception and development of a DSM-based gamification authoring system to support university teaching. (Original title: “Konzeption und Entwicklung eines DSM-basierten Gamification Authoring Systems zur Unterstützung hochschulischer Lehre” [German]),” Dissertation, Universität Regensburg, 2018.

References

- [27] B. A. Kitchenham and P. O. Brereton, "A systematic review of systematic review process research in software engineering," *Information and Software Technology*, vol. 55, no. 12, pp. 2049–2075, 2013.
- [28] IEEE, "IEEE Xplore Digital Library," 2020. [Online]. Available: <https://ieeexplore.ieee.org/Xplore/home.jsp>
- [29] ACM, "ACM Digital Library," 2020. [Online]. Available: <http://dl.acm.org/>
- [30] Springer Nature Switzerland AG, "SpringerLink," 2020. [Online]. Available: <https://link.springer.com/>
- [31] Elsevier B.V., "Scopus," 2020. [Online]. Available: <http://www.scopus.com/>
- [32] —, "ScienceDirect," 2020. [Online]. Available: <https://www.sciencedirect.com/>
- [33] Institute of Education Sciences of the US Department of Education, "Eric – education resources information center," 2020. [Online]. Available: <https://eric.ed.gov/>
- [34] Clarivate Analytics, "Web of Science," 2020. [Online]. Available: <http://www.webofknowledge.com>
- [35] Google LLC, "Google Scholar," 2020. [Online]. Available: <http://scholar.google.de/>
- [36] Cornell University, "arXiv.org: e-Print archive," 2020. [Online]. Available: <https://arxiv.org/>
- [37] Schloss Dagstuhl and Universität Trier, "dblp: computer science bibliography," 2020. [Online]. Available: <https://dblp.uni-trier.de/>
- [38] P. Devanbu, T. Zimmermann, and C. Bird, "Belief evidence in empirical software engineering," in *International Conference on Software Engineering (ICSE)*, 2016, pp. 108–119.
- [39] M. M. Inuwa and A. Varol, "Intensity of misconception in software engineering," in *International Informatics and Software Engineering Conference (UBMYK)*, 2019, pp. 1–6.
- [40] J. Ivins, B. R. von Kinsky, D. Cooper, and M. Robey, "Software engineers and engineering: A survey of undergraduate preconceptions," in *Frontiers in Education (FIE)*, 2006, pp. MIF–6–11.
- [41] B. Özkan and O. Demirors, "On the seven misconceptions about functional size measurement," in *Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement (IWSM-MENSURA)*, 2016, pp. 45–52.
- [42] J. S. van der Ven and J. Bosch, "Busting software architecture beliefs: A survey on success factors in architecture decision making," in *Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, 2016, pp. 42–49.
- [43] A. Begel and B. Simon, "Struggles of new college graduates in their first software development job," *SIGCSE Bull*, vol. 40, no. 1, pp. 226–230, 2008.
- [44] D. DeMarco Brown, "Five agile UX myths," *Journal of Usability Studies*, vol. 8, no. 3, pp. 55–60, 2013.
- [45] L. A. Sudol and C. Jaspán, "Analyzing the strength of undergraduate misconceptions about software engineering," in *International Computing Education Research (ICER)*. ACM, 2010, pp. 31–39.

References

- [46] R. H. Wilcox, "Behavioral misconceptions facing the software engineer," in Computer and Information Sciences, ser. SEN Report Series Software Engineering, J. T. Tou, Ed. Elsevier, 1971, vol. 2, pp. 285–287.
- [47] J. P. Bowen and M. G. Hinchey, "Seven more myths of formal methods: Dispelling industrial prejudices," in FME'94: Industrial Benefit of Formal Methods, ser. LNCS 873. Springer, 1994, pp. 105–117.
- [48] A. Hall, "Seven myths of formal methods," IEEE Software, vol. 7, no. 5, pp. 11–19, 1990.
- [49] D. Carlson, "Debunking agile myths," CrossTalk, vol. 30, no. 3, pp. 32–36, 2017.
- [50] S. Jahn, C. Gold-Veerkamp, R. Reuter, J. Mottok, and J. Abke, "Secure software engineering in academic education: Students' preconceptions of it security," in International Conference of Education, Research and Innovation (ICERI), IATED, Ed., 2019, pp. 6825–6834.
- [51] P. Ralph and B. J. Oates, "The dangerous dogmas of software engineering," 2018, [retrieved: 09, 2020]. [Online]. Available: arXiv: 1802.06321v1
- [52] Shrikanth N. C. and T. Menzies, "Assessing developer beliefs: A reply to "perceptions, expectations, and challenges in defect prediction"," 2019, [retrieved: 09, 2020]. [Online]. Available: arXiv: 1904.05794v1
- [53] —, "Assessing practitioner beliefs about software defect prediction," 2020, accepted at ICSE'20, [retrieved: 09, 2020]. [Online]. Available: arXiv: 1912.10093v3
- [54] Z. Wan et al., "Perceptions, expectations, and challenges in defect prediction," IEEE Transactions on Software Engineering, pp. 1–26, 2018.
- [55] D. Rombach and F. Seelisch, "Formalism in software engineering: Myths versus empirical facts," in Balancing Agility and Formalism in Software Engineering, D. Hutchison, et al., Ed. Springer, 2008, pp. 13–25.
- [56] R. Duit, "Schülervorstellungen – von Lerndefiziten zu neuen Unterrichtsansätzen [German]," in Schülervorstellungen in der Physik, R. Müller, R. Wodzinski, and M. Hopf, Eds. Köln: Aulis, 2011, pp. 8–14.
- [57] C. Schulte and J. Magenheimer, "Novices' expectations and prior knowledge of software development," in First international Workshop on Computing education research, R. Anderson, S. A. Fincher, and M. Guzdial, Eds. ACM Press, 2005, pp. 143–153.