

Ernst Denert Software Engineering Awards 2019



Michael Felderer, Wilhelm Hasselbring, Heiko Koziolak, Florian Matthes, Lutz Prechelt, Ralf Reussner, Bernhard Rumpe, and Ina Schaefer

Abstract The need to improve software engineering practices is continuously rising and software development practitioners are highly interested in improving their software systems and the methods to build them. And well, software engineering research has numerous success stories. The Ernst Denert Software Engineering Award specifically rewards researchers that value the practical impact of their work and aim to improve current software engineering practices. This chapter summarizes the awards history as well as the current reward process and criteria.

M. Felderer

University of Innsbruck, Innsbruck, Austria
e-mail: michael.felderer@uibk.ac.at

W. Hasselbring

Department of Computer Science, Kiel University, Kiel, Germany
e-mail: wha@informatik.uni-kiel.de

H. Koziolak

ABB, Ladenburg, Germany
e-mail: heiko.koziolak@de.abb.com

F. Matthes

Institute of Computer Science, Technical University Munich, Garching, Germany
e-mail: matthes@in.tum.de

L. Prechelt

Department of Mathematics and Computer Science, Freie Universität Berlin, Berlin, Germany
e-mail: prechelt@inf.fu-berlin.de

R. Reussner

Karlsruhe Institute of Technology, Karlsruhe, Germany
e-mail: reussner@kit.edu

B. Rumpe (✉)

RWTH Aachen University, Aachen, Germany
e-mail: rumpe@se-rwth.de

I. Schaefer

Technische Universität Braunschweig, Braunschweig, Germany
e-mail: i.schaefer@tu-bs.de

© The Author(s) 2020

M. Felderer et al. (eds.), *Ernst Denert Award for Software Engineering 2019*,
https://doi.org/10.1007/978-3-030-58617-1_1

1 Relevance of Software Engineering

The need to improve software engineering practices is continuously rising. While commercial software systems in the 1980s and 1990s typically comprised a few hundred thousand lines of code, Windows 10 today has more than 50 million lines of code. Google’s code base size is estimated at more than 2 billion lines of code, where thousands of software developers are contributing. There are more than 20 million professional software developers in the world, and in the USA they already make up 2.5% of the entire workforce. Recent technology advances in connectivity and artificial intelligence create a plethora of ever-growing software systems that are impossible to manage without sound software engineering practices.

Software engineering research had numerous success stories in the past. Design patterns experienced by practitioners were codified by academics and became an important communication tool for developers across the world. The Representational State Transfer (REST) architecture style captures key principles for designing web services and was a fundamental contribution to the world wide web. The ACM Impact Project traced back the design of many commercial middleware products to Ph.D. theses on distributed computing from the 1970s. Similar investigations were done for programming languages, configuration management, code walkthroughs, and exception handling. While it is still relatively rare that the output of a software engineering Ph.D. thesis is directly commercialized, the concepts and ideas sometimes influence commercial products indirectly, and the impact on software engineering practices is visible only much later.

Software development practitioners are highly interested in improving their software systems and the methods to build them. Microsoft Research interviewed 3000 Microsoft developers on how they viewed the practical relevance of academic works in the field. In general, the developers consider the best works in the field positive and are looking for ways to integrate them into their daily work. However, some research ideas are also characterized as “unwise,” for example tools that are not needed, empirical studies that produce no actionable outcome, methods that are tuned for very specific contexts, approaches that incur such high costs that they outweigh the benefits.

Therefore, the Ernst Denert Software Engineering Award specifically rewards researchers that value the practical impact of their work and aim to improve current software engineering practices. Creating tighter feedback loops between professional practitioners and academic researchers is essential to make research ideas ready for industry adoption. Researchers who demonstrate their proposed methods and tools on non-trivial systems under real-world conditions shall be supported, so that the gap between research and practice can be decreased.

2 History of the Ernst Denert Software Engineering Award

“Passion Software-Engineering.” This was the headline of a 2015 newspaper article about Ernst Denert. And they were absolutely right. Ernst Denert (Fig. 1) is really passionate about developing software with excellent quality in a predictable and systematic style. His seminal book on Software-Engineering in 2001,¹ the famous “Black Book on Software-Engineering” is depicted in Fig. 2. This book was the basis for software development projects at the company *sd&m* as well as teaching courses in practical application of software development techniques to industrial projects.

Fig. 1 Ernst Denert



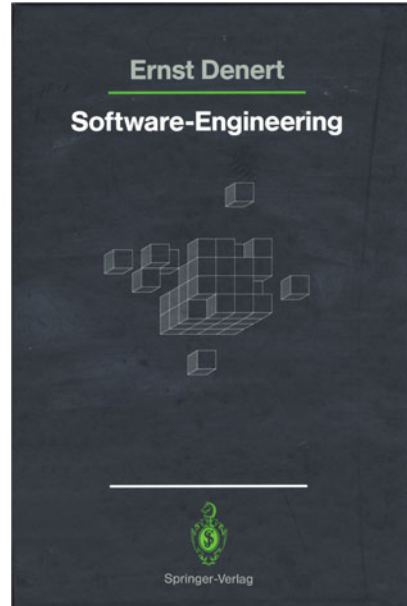
Ernst Denert furthermore is very much interested in encouraging young people to study computer science or at least to learn how programming and digitalization works, but also computer science students to concentrate on software engineering principles and software development. Starting with a degree in communications engineering he early moved to computer science in his Ph.D. thesis at TU Berlin, where software engineering was already an issue.

Then he went to industry and relatively soon started together with colleagues the already mentioned software development company “*sd&m*” (Software Design and Management) in 1982. He headed the company until 2001, when he left for a new endeavor, namely CEO of the IVU Traffic Technologies AG. He thus moved from individual software development projects to a software product house.

Ernst Denert was always passionate on Software Engineering. That was why he also was active in teaching at the Munich University of Technology in 1991, where he also received an Honorary Professorship position and also founded the Ernst

¹Ernst Denert: Software-Engineering, Springer Berlin, 1991.

Fig. 2 Software Engineering
by Ernst Denert²



Denert Software Engineering price back in 1989. The price was at first funded by the Ernst Denert Stiftung and now is funded by the Gerlind & Ernst Denert Stiftung.

The software engineering community greatly thanks the Gerlind & Ernst Denert Stiftung for their continuous support of the community.

As said: It was always an important topic for Professor Denert to push forward the field of Software Engineering. This was therefore logical to run a special conference on Software Engineering to bring together the most influential Software Engineering Pioneers and also publish a proceedings of that happening (Fig. 3).⁴

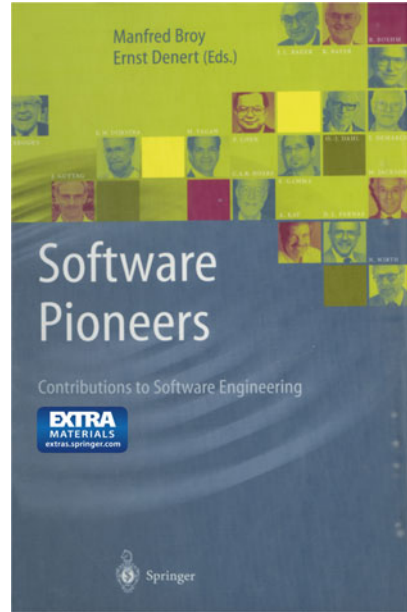
Currently, Ernst Denert is also funding and working on a project called “Vitruv,”⁵ where we can expect a book on software architecture to emerge. We are looking forward to see many more interesting things to come.

²From <https://www.springer.com/de/book/9783642843440>.

³From <https://www.springer.com/de/book/9783540430810>.

⁴Software Pioneers. Contributions to Software Engineering. Published by Manfred Broy and Ernst Denert, Springer, 2002.

⁵<https://software-architecture.org/>.

Fig. 3 Software Pioneers³

3 Award Submission Process

For the 2019 and forthcoming awards, the criteria have changed slightly, the organization was taken over by the GI, and the form of presentation of the nominated works has been extended by this book. The call for submissions to the 2019 awards was roughly as follows.

The Ernst Denert Software Engineering Award for outstanding contributions to the Software Engineering discipline is awarded by the

- Fachbereich Softwaretechnik of the German Society for Informatics (GI), the
- Suisse Society for Informatics (SI), and the
- Austrian Computing Society (OCG).

The award is given annually and endowed with 5,000 Euros. We gratefully thank the *Gerlind & Ernst Denert-Stiftung* for the kind donation of the price and takeover of the financial costs for the colloquium.

The prize is awarded for excellent work within the discipline of Software Engineering, which includes methods, tools, and procedures for better and efficient development of high quality software that was created in the area of SI, OCG, or GI.

An essential requirement is applicability and usability in industrial practice. The practical applicability must be demonstrated. Open source software or results published as open data are welcome, but commercial products are excluded.

Proposals can be sent by (1) scientific universities in Germany, Austria, and Switzerland that have the right to confer doctorates in computer science. Univer-

sities may propose dissertations and habilitations. All work with a final examination in the period from September 1 to August 31 is permitted. (2) Proposals can also be sent by all software engineers—these are typically members of one of the Software Engineering divisions within the GI, SI, or OCG. They may propose academic or non-academic work that they consider worthwhile. This particularly includes work in the area of open source or open data with relevance to the Software Engineering field.

Deadline for a proposal in 2019 was December 15.

For dissertations and postdoctoral theses the following documents have been submitted: dissertation (resp. habilitation), justification of the proposal by the submitting software engineer/university, the reports of the reviewing experts, whereby at least one review was prepared by an independent expert, a curriculum vitae of the Doctoral candidate including a list of publications, and a link to the corresponding doctoral regulations of the university.

For open source and open data projects the submission had to include a project description, justification of the proposal by the submitting software engineer, a laudatio of at least one independent expert, a curriculum vitae of the project members involved, a list of publications referring to the project, links on the sources of the published results (e.g., an open data directory), and installation and usage instructions.

4 Selection Criteria

The jury discussed extensively the criteria to be applied to the submissions and agreed on the following list of considerations:

1. Above all is the potential to contribute to a better software engineering practice. Practical applicability of the results is therefore a *conditio sine qua non*.
2. This applicability needs to be demonstrated (or, if long timescales are involved, at least demonstrated to be likely).
3. Contributions may aim to be useful directly, e.g., via a tool, or indirectly via theoretical or empirical insights.
4. They may target the software product or the software development process.
5. They may pertain to “hot” topics or to eternal problems of software development.
6. They may promise large improvements (“tenfold”) in narrow domains of applicability or incremental improvements in broad domains. Domains of applicability can be narrow or broad with respect to technology assumptions, development process assumptions, assumptions regarding the target software application domain, or the set of software development tasks being addressed.
7. These criteria will lead to the need of comparing apples to oranges. When comparing dissimilar contributions in a neck-and-neck race, the jury will favor more credible packages over more impressive ones.
8. The best submissions are those that will be viewed as important steps forward even 15 years from now.

9. This is not a Ph.D. dissertation price. Submissions can come from individuals or groups, can take any form, may cover longer or shorter time spans, and do not even need to be framed as research.

5 Selection Process

From the submitted proposal the jury, which is identical to the editors of this book, selected an appropriate subset of candidates.

The selected “*Ernst Denert Software Engineering Award Nominees*,” were asked to present their results and further empirical findings in the form of a colloquium that took place in January 15th to 17th, 2020 in the Leibniz Center for Computer Science at Schloss Dagstuhl (Fig. 4).

Fig. 4 The nominees and the committee in Dagstuhl⁶



All nominated candidates and especially the winning person (or group) are invited to summarize their work in this book.

Further information can be found on the webpage of the Software Engineering community of the GI.⁷

6 The Award Nominees and the Winner

Based on the nominations that have been received, the following list of nominees was selected and invited to a two day Dagstuhl seminar. Each of the candidates had the possibility to present and defend her or his thesis in front of the committee (in alphabetical order):

⁶From <https://www.dagstuhl.de/20033>.

⁷<https://fb-swt.gi.de/ernst-denert-se-preis>.

- Sebastian Baltes, Universität Trier
was nominated by his university with his doctoral thesis on *Software Developers' Work Habits and Expertise*. His doctor father was Stephan Diehl, his second examiner Stefan Wagner.
- Timo Greifenberg, RWTH Aachen
was nominated by his university with his doctoral thesis on *Artefaktbasierte Analyse modellgetriebener Softwareentwicklungsprojekte*. His doctor father was Bernhard Rumpel, his second examiner Steffen Becker.
- Marco Konersmann, Universität Duisburg-Essen
was nominated by his university with his doctoral thesis on *Explicitly Integrated Architecture*. His doctor father was Michael Goedicke, his second examiner Ralf Reussner.
- Marija Selakovic, Technischen Universität Darmstadt
was nominated by her university with her doctoral thesis on *Actionable Program Analyses for Improving Software Performance*. Her doctor father was Michael Pradel, her second examiner Frank Tip.
- Johannes Späth, Paderborn University
was nominated by his university with his doctoral thesis on *Synchronized Pushdown Systems for Pointer and Data-Flow Analysis*. His doctor father was Eric Bodden, his second examiner Karim Ali.

We congratulate *Johannes Späth*, his doctoral supervisor *Eric Bodden*, and Paderborn University for winning the Ernst Denert Software Engineering Award 2019.

Johannes had the possibility to present the most important results of his thesis called *Synchronized Pushdown Systems for Pointer and Data-Flow Analysis* during his talk at the software engineering conference SE 2020 from February 24 to 28, 2020, Innsbruck, Austria and of course also in Chapter 3 of this book.



Fig. 5 Award ceremony in Innsbruck: Professor Denert and Dr. Späth

Figure 5 shows the laudator Professor Denert handing over the award certificate to the award winner Johannes Späth after the presentation.

7 Contents of the Book

All the nominees were given the chance to present in the following chapters

- the key findings of the work,
- their relevance and applicability to practice and industrial software engineering projects, and
- additional information and findings that have only been discovered afterwards, e.g., when applying the results in industry.

Software Engineering projects are teamwork. In practice outstanding accident research is also teamwork. This somewhat conflicts with the requirement that a doctoral thesis is a monograph. To reflect the teamwork idea, quite like Professor Denert is discussing in Chap. 2, we decided that the following papers, describing this research can (or should even) be written with co-authors that somehow contributed to the findings during the thesis development or continue to work with the results.

7.1 *Thanks*

We gratefully thank Professor Ernst Denert for all his help in the process and the *Gerlind & Ernst Denert Stiftung* for the kind donation of the price and takeover of the financial costs for the colloquium. We also thank all the people helping in the organization, including Katrin Hölldobler for managing the submission and the selection process, the colloquium, and the book organization. We thank the organization team of Castle Dagstuhl, whose customary overwhelming hospitality allowed us to run the colloquium. Furthermore, we thank the organization team of the software engineering conference SE 2020 to host the award ceremony in Innsbruck. Finally, we thank the GI, the OCG, and the SI computer science societies and especially their Software Engineering divisions for their support in making this price a success.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

