

# Towards a Theory of Software Development Expertise

Sebastian Baltes,<sup>1</sup> Stephan Diehl<sup>2</sup>

**Abstract:** Software development includes diverse tasks such as implementing new features, analyzing requirements, and fixing bugs. Being an expert in those tasks requires a certain set of skills, knowledge, and experience. Several studies investigated individual aspects of software development expertise, but what is missing is a comprehensive theory. We present a first conceptual theory of software development expertise that is grounded in data from a mixed-methods survey with 335 software developers and in literature on expertise and expert performance. Our theory currently focuses on programming, but already provides valuable insights for researchers, developers, and employers. The theory describes important properties of software development expertise and which factors foster or hinder its formation, including how developers' performance may decline over time. Moreover, our quantitative results show that developers' expertise self-assessments are context-dependent and that experience is not necessarily related to expertise.

**Keywords:** software engineering; expertise; theory; psychology

## 1 Introduction

An *expert* is, according to Merriam-Webster, someone “having or showing special skill or knowledge because of what [s/he has] been taught or what [s/he has] experienced”. K. Anders Ericsson, a psychologist and expertise researcher, defines *expertise* as “the characteristics, skills, and knowledge that distinguish experts from novices and less experienced people”. For some areas, such as playing chess, there exist representative tasks and objective criteria for identifying experts. In software development, however, it is more difficult to find objective measures for quantifying expert performance. Aside from programming, software development involves many other tasks such as requirements engineering, testing, and debugging, at which a software development expert is expected to be good.

In the past, researchers investigated certain aspects of software development expertise such as the influence of programming experience, desired attributes of software engineers, or the time it takes for developers to become “fluent” in software projects. However, there is currently no theory combining those individual aspects. Such a theory could help structuring existing knowledge about software development expertise in a concise and precise way and hence facilitate its communication.

---

<sup>1</sup> University of Trier, Software Engineering, 54296 Trier, Germany, [research@sbaltes.com](mailto:research@sbaltes.com)

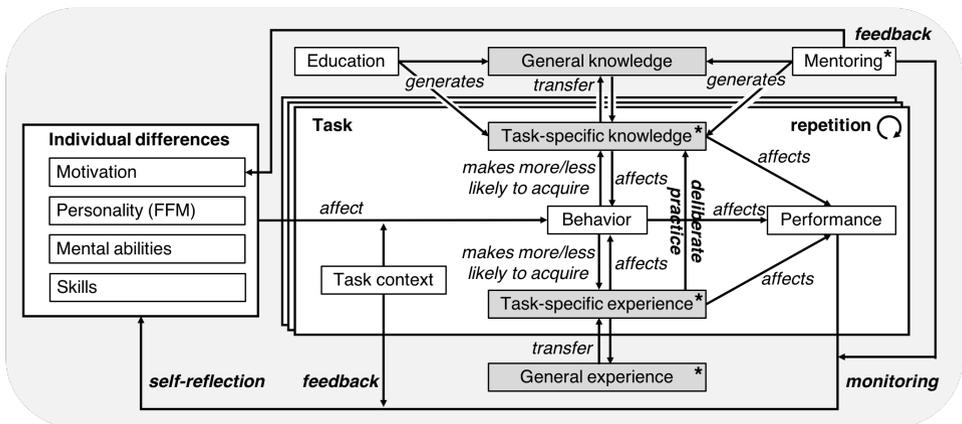
<sup>2</sup> University of Trier, Software Engineering, 54296 Trier, Germany, [diehl@uni-trier.de](mailto:diehl@uni-trier.de)

With our paper, we contribute a theory that describes *central properties* of software development expertise and *important factors* influencing its *formation*. The theory is grounded in data from a mixed-methods survey with 335 participants and in literature on expertise and expert performance. Our expertise model is task-specific, but includes the notion of transferable knowledge and experience from related fields or tasks. On a conceptual level, the theory focuses on factors influencing the formation of software development expertise over time.

The theory can help researchers, software developers as well as employers. *Researchers* can use it to design studies related to expertise and expert performance, and in particular to reflect on the complex relationship between experience and expertise, which is relevant for many self-report studies. *Software developers* can learn which properties are distinctive for experts in their field, which behaviors may lead to becoming a better software developer, and which contextual factors could affect expertise development. If they are already “senior”, they can learn what other developers expect from good mentors or which effects age-related performance decline may have on them. Finally, *employers* can learn what typical reasons for demotivation among their employees are, hindering developers to improve, and how they can build a work environment supporting expertise development of their staff.

## 2 Conceptual Theory

The following figure shows the central concepts and relationships of our theory. More details on our research design and the resulting theory can be found in the full paper [BD18].



## References

- [BD18] Baltes, Sebastian; Diehl, Stephan: Towards a Theory of Software Development Expertise. In: 26th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2018). ACM, pp. 187–200, 2018.