

Teaching Formal Methods

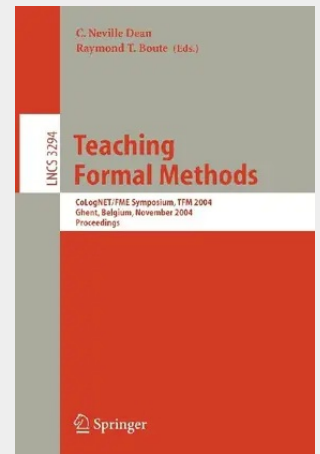
CoLogNET/FME Symposium, TFM 2004, Ghent, Belgium, November 18-19, 2004.
Proceedings

“Professional engineers can often be distinguished from other designers by the engineers’ ability to use mathematical models to describe and 1 analyze their products.” This observation by Parnas describes the de facto professional standards in all classical engineering disciplines (civil, mechanical, electrical, etc.). Unfortunately, it is in sharp contrast with current (industrial) practice in software design, where mathematical models are hardly used at all, even by those who, 2 in Holloway’s words “aspire to be engineers.” The rare exceptions are certain critical applications, where mathematical techniques are used under the general name formal methods.

Yet, the same characteristics that make formal methods a necessity in critical applications make them also advantageous in everyday software design at various levels from design efficiency to software quality. Why, then, is education failing with respect to formal methods? – failing to convince students, academics and practitioners alike that formal methods are truly pragmatic; – failing to overcome a phobia of formality and mathematics; – failing to provide students with the basic skills and understanding required to adopt more mathematical and logical approaches to software development. Until education takes these failings seriously, formal methods will be an obscure byway in software engineering, which in turn will remain severely impoverished as a result.

Professional engineers can often be distinguished from other designers by the engineers’ ability to use mathematical models to describe and 1 analyze their products. This observation by Parnas describes the de facto professional standards in all classical engineering disciplines (civil, mechanical, electrical, etc.). Unfortunately, it is in sharp contrast with current (industrial) practice in software design, where mathematical models are hardly used at all, even by those who, 2 in Holloway’s words “aspire to be engineers.” The rare exceptions are certain critical applications, where mathematical techniques are used under the general name formal methods.

Yet, the same characteristics that make formal methods a necessity in critical applications make them also advantageous in everyday software design at various levels from design efficiency to software quality. Why, then, is education failing with respect to formal methods? – failing to convince students, academics and practitioners alike that formal methods are truly pragmatic; – failing to overcome a phobia of formality and mathematics; – failing to provide students with the basic skills and understanding required to adopt more mathematical and logical approaches to software development. Until education takes these failings seriously, formal methods will be an obscure byway in software engineering, which in turn will remain severely impoverished as a result.



53,49 €

49,99 € (zzgl. MwSt.)

Lieferfrist: bis zu 10 Tage

Artikelnummer: 9783540236115

Medium: Buch

ISBN: 978-3-540-23611-5

Verlag: Springer

Erscheinungstermin: 17.11.2004

Sprache(n): Englisch

Auflage: 2004

Serie: Lecture Notes in Computer Science

Produktform: Kartoniert

Gewicht: 406 g

Seiten: 252

Format (B x H): 155 x 235 mm

