

<b>1. Record Nr.</b>	UNINA9910349394003321
<b>Titolo</b>	Engineering Trustworthy Software Systems : Third International School, SETSS 2017, Chongqing, China, April 17–22, 2017, Tutorial Lectures // edited by Jonathan P. Bowen, Zhiming Liu, Zili Zhang
<b>Pubbl/distr/stampa</b>	Cham : , : Springer International Publishing : , : Imprint : Springer, , 2018
<b>ISBN</b>	3-030-02928-X
<b>Edizione</b>	[1st ed. 2018.]
<b>Descrizione fisica</b>	1 online resource (XVII, 219 p. 127 illus., 31 illus. in color.)
<b>Collana</b>	Programming and Software Engineering ; ; 11174
<b>Disciplina</b>	005.1
<b>Soggetti</b>	Software engineering Programming languages (Electronic computers) Computer programming Computer logic Computers Computer system failures Software Engineering Programming Languages, Compilers, Interpreters Programming Techniques Logics and Meanings of Programs The Computing Profession System Performance and Evaluation
<b>Lingua di pubblicazione</b>	Inglese
<b>Formato</b>	Materiale a stampa
<b>Livello bibliografico</b>	Monografia
<b>Note generali</b>	Includes index.
<b>Nota di contenuto</b>	A Guide to Rely/Guarantee Thinking -- An Illustrated Guide to the Model Theory of Supertype Abstraction and Behavioral Subtyping -- Formalizing Hoare Logic in PVS -- Modeling Concurrency in Dafny -- Software Is Not Soft: Challenges and Approaches to Dynamic Software Update -- Challenges for Formal Semantic Description: Responses from the Main Approaches. .
<b>Sommario/riassunto</b>	This volume contains a record of some of the lectures and seminars delivered at the Third International School on Engineering Trustworthy Software Systems (SETSS 2017), held in April 2017 at Southwest

University in Chongqing, China. The six contributions included in this volume provide an overview of leading-edge research in methods and tools for use in computer system engineering. They have been distilled from six original courses delivered at the school on topics such as: rely/guarantee thinking; Hoare-style specification and verification of object-oriented programs with JML; logic, specification, verification, and interactive proof; software model checking with Automizer; writing programs and proofs; engineering self-adaptive software-intensive systems; and with an additional contribution on the challenges for formal semantic description. The material is useful for postgraduate students, researchers, academics, and industrial engineers, who are interested in the theory and practice of methods and tools for the design and programming of trustworthy software systems. .

---