

1. Record Nr.	UNISA996466096503316
Titolo	Formal Methods and Software Development. Proceedings of the International Joint Conference on Theory and Practice of Software Development (TAPSOFT), Berlin, March 25-29, 1985 [[electronic resource]] : Volume 2: Colloquium on Software Engineering (CSE) // edited by Hartmut Ehrig, Christiane Floyd, Maurice Nivat, James Thatcher
Pubbl/distr/stampa	Berlin, Heidelberg : , : Springer Berlin Heidelberg : , : Imprint : Springer, , 1985
ISBN	3-540-39307-2
Edizione	[1st ed. 1985.]
Descrizione fisica	1 online resource (XVII, 459 p.)
Collana	Lecture Notes in Computer Science, , 0302-9743 ; ; 186
Disciplina	005.1
Soggetti	Software engineering Programming languages (Electronic computers) Computer logic Software Engineering/Programming and Operating Systems Software Engineering Programming Languages, Compilers, Interpreters Logics and Meanings of Programs
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Bibliographic Level Mode of Issuance: Monograph
Nota di contenuto	On the relevance of formal methods to software development -- Combining algebraic and predicative specifications in Larch -- The role of proof obligations in software design -- Functional semantics of modules -- Intuition in software development -- A rational design process: How and why to fake it -- Formalization in systems development -- Specifying and prototyping: Some thoughts on why they are successful -- A formal specification of line representations on graphics devices -- Experiences with the PSG — Programming System Generator -- Software construction using typed fragments -- Graph grammar engineering: A method used for the development of an integrated programming support environment -- Multidimensional tree-structured file spaces -- A theory of abstract data types for

program development: Bridging the gap? -- Program development and documentation by informal transformations and derivations -- ASSPEGIQUE: An integrated environment for algebraic specifications -- Application of PROLOG to test sets generation from algebraic specifications -- A PROLOG environment for developing and reasoning about data types -- Algebraic specification of synchronisation and errors: A telephonic example -- Modelling concurrent modules -- Synthesis of parallel programs invariants -- Analyzing safety and fault tolerance using Time Petri nets -- Algebraic specification of a communication scheduler -- The integration and distribution phase in the software life cycle -- Formalized software development in an industrial environment -- Object oriented concurrent programming and industrial software production -- Experience of introducing the Vienna development method into an industrial organisation -- EDP system development methodology: Auditability and control -- Experiences with object oriented programming.
