

1. Record Nr.	UNISA996465870803316
Autore	Liblit Ben
Titolo	Cooperative Bug Isolation [[electronic resource]] : Winning Thesis of the 2005 ACM Doctoral Dissertation Competition // by Ben Liblit
Pubbl/distr/stampa	Berlin, Heidelberg : , : Springer Berlin Heidelberg : , : Imprint : Springer, , 2007
ISBN	1-280-86572-5 9786610865727 3-540-71878-8
Edizione	[1st ed. 2007.]
Descrizione fisica	1 online resource (112 p.)
Collana	Programming and Software Engineering ; ; 4440
Disciplina	005.1068
Soggetti	Software engineering Computer logic Algorithms Software Engineering/Programming and Operating Systems Software Engineering Logics and Meanings of Programs Algorithm Analysis and Problem Complexity
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Revised version of the author's thesis (Ph.D.)--University of California, Berkeley, 2004. "Association for Computing Machinery"--Cover.
Nota di bibliografia	Includes bibliographical references (pages [97]-101) and index.
Nota di contenuto	Instrumentation Framework -- Practical Considerations -- Techniques for Statistical Debugging -- Related Work -- Conclusion.
Sommario/riassunto	Efforts to understand and predict the behavior of software date back to the earliest days of computer programming, over half a century ago. In the intervening decades, the need for effective methods of understanding software has only increased; software has spread to become the underpinning of much of modern society, and the potentially disastrous consequences of broken or poorly understood software have become all too apparent. Ben Liblit's work reconsiders two common assumptions about how we should analyze software and it arrives at some striking new results. In principle, understanding software is not such a hard problem. Certainly a computer

scientist studying programs appears to be in a much stronger position than, say, a biologist trying to understand a living organism or an economist trying to understand the behavior of markets, because the biologist and the economist must rely on indirect observation of the basic processes they wish to understand. A computer scientist, however, starts with a complete, precise description of the behavior of software—the program itself! Of course, the story turns out not to be so straightforward, because despite having a perfect description, programs are sufficiently complex that it is usually difficult or even impossible to answer many simple questions about them.
