

1. Record Nr.	UNINA9910463119703321
Autore	Schmidt Richard <1956->
Titolo	Software engineering [[electronic resource]] : architecture-driven software development / / Richard F. Schmidt
Pubbl/distr/stampa	Waltham, Mass., : Morgan Kaufmann, an imprint of Elsevier, 2013
ISBN	0-12-407878-8
Edizione	[1st edition]
Descrizione fisica	1 online resource (1 v.) : ill
Disciplina	005.1
Soggetti	Software engineering Software architecture Computer software - Development Electronic books.
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di bibliografia	Includes bibliographical references and index.
Nota di contenuto	Section 1. Software Engineering Foundations -- Introduction to Software Engineering -- Generic Software Development Framework -- The Software Architecture -- Coping with Project Complexity -- Integrated Product and Process Development -- Section 2. Specifying Software Requirements -- Understanding Software Requirements -- Software Requirements Analysis -- Managing Requirements -- Section 3. Developing the Functional Architecture -- Formulating the Functional Architecture -- Functional Analysis and Allocation Tasks -- Section 4. Developing the Physical Architecture -- Configuring the Physical Architecture -- Application Design Synthesis -- Section 5. Software Architectural Design Decisions -- Evaluating Architectural Alternatives -- Software Analysis Tasks -- Section 6. Managing the Software Architecture -- Software Control Tasks -- Software Verification and Validation Tasks -- Section 7. Software Integrated Product and Process Development -- Software Requirements Definition -- Preliminary Architecture Definition -- Detailed Architecture Definition -- Software Implementation.
Sommario/riassunto	Software Engineering: Architecture-driven Software Development is the first comprehensive guide to the underlying skills embodied in the IEEE's Software Engineering Body of Knowledge (SWEBOK) standard.

Standards expert Richard Schmidt explains the traditional software engineering practices recognized for developing projects for government or corporate systems. Software engineering education often lacks standardization, with many institutions focusing on implementation rather than design as it impacts product architecture. Many graduates join the workforce with incomplete skills, leading to software projects that either fail outright or run woefully over budget and behind schedule. Additionally, software engineers need to understand system engineering and architecture—the hardware and peripherals their programs will run on. This issue will only grow in importance as more programs leverage parallel computing, requiring an understanding of the parallel capabilities of processors and hardware. This book gives both software developers and system engineers key insights into how their skillsets support and complement each other. With a focus on these key knowledge areas, Software Engineering offers a set of best practices that can be applied to any industry or domain involved in developing software products. A thorough, integrated compilation on the engineering of software products, addressing the majority of the standard knowledge areas and topics Offers best practices focused on those key skills common to many industries and domains that develop software Learn how software engineering relates to systems engineering for better communication with other engineering professionals within a project environment
