1.

Record Nr.	UNINA9910337857603321
Autore	Reussner Ralf
Titolo	Managed Software Evolution / / edited by Ralf Reussner, Michael Goedicke, Wilhelm Hasselbring, Birgit Vogel-Heuser, Jan Keim, Lukas Märtin
Pubbl/distr/stampa	Cham : , : Springer International Publishing : , : Imprint : Springer, , 2019
ISBN	3-030-13499-7
Edizione	[1st ed. 2019.]
Descrizione fisica	1 online resource (XXII, 426 p. 160 illus., 79 illus. in color.)
Classificazione	BUS083000COM005000COM014000COM051230
Disciplina	005.1
Soggetti	Software engineering Software engineering - Management Electronic data processing - Management Computer science Software Engineering Software Management IT Operations Theory of Computation
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di contenuto	1 Introducing Managed Software Evolution -- 2 The Nature of Software Evolution -- 3 Addressed Challenges -- 4 Introduction to Case Studies -- 5 Tacit Knowledge in Software Evolution -- 6 Continuous Design Decision Support -- 7 Model-based Round-Trip Engineering and Testing of Evolving Software Product Lines -- 8 Performance Analysis Strategies for Software Variants and Versions -- 9 Maintaining Security in Software Evolution -- 10 Learning from Evolution for Evolution -- 11 Formal Verication of Evolutionary Changes -- 12 Case Studies for the Community -- 13 Lessons Learned -- 14 Future Research.
Sommario/riassunto	This open access book presents the outcomes of the “Design for Future – Managed Software Evolution” priority program 1593, which was launched by the German Research Foundation (“Deutsche Forschungsgemeinschaft (DFG)”) to develop new approaches to software engineering with a specific focus on long-lived software

systems. The different lifecycles of software and hardware platforms lead to interoperability problems in such systems. Instead of separating the development, adaptation and evolution of software and its platforms, as well as aspects like operation, monitoring and maintenance, they should all be integrated into one overarching process. Accordingly, the book is split into three major parts, the first of which includes an introduction to the nature of software evolution, followed by an overview of the specific challenges and a general introduction to the case studies used in the project. The second part of the book consists of the main chapters on knowledge carrying software, and cover tacit knowledge in software evolution, continuous design decision support, model-based round-trip engineering for software product lines, performance analysis strategies, maintaining security in software evolution, learning from evolution for evolution, and formal verification of evolutionary changes. In turn, the last part of the book presents key findings and spin-offs. The individual chapters there describe various case studies, along with their benefits, deliverables and the respective lessons learned. An overview of future research topics rounds out the coverage. The book was mainly written for scientific researchers and advanced professionals with an academic background. They will benefit from its comprehensive treatment of various topics related to problems that are now gaining in importance, given the higher costs for maintenance and evolution in comparison to the initial development, and the fact that today, most software is not developed from scratch, but as part of a continuum of former and future releases.