

Software Fundamentals: Collected Papers by David L. Parnas

By Daniel M. Hoffman, David M. Weiss

Software Fundamentals: Collected Papers by David L. Parnas By Daniel M. Hoffman, David M. Weiss

David L. Parnas is one of the grandmasters of software engineering. His academic research and industrial collaborations have exerted far-reaching influence on software design and development. His groundbreaking writings capture the essence of the innovations, controversies, challenges, and solutions of the software industry. Together, they constitute the foundation for modern software theory and practice. This book contains 33 of his most influential papers in various areas of software engineering. Leading thinkers in software engineering have contributed short introductions to each paper to provide the historical context surrounding each papers conception and writing. Software Fundamentals: Collected Papers by David L. Parnas is a practical guide to key software engineering concepts that belongs in the library of every software professional. It introduces and explains such seminal topics as: Relational and tabular documentation Information hiding as the basis for modular program construction Abstract interfaces that provide services without revealing implementation Program families for the efficient development of multiple software versions The status of software engineering as a

Software Fundamentals: Collected Papers by David L. Parnas By Daniel M. Hoffman, David M. Weiss
Bibliography

- Sales Rank: #1162725 in Books
- Published on: 2001-04-19
- Original language: English
- Number of items: 1
- Dimensions: 8.90" h x 1.50" w x 7.00" l, 2.60 pounds
- Binding: Paperback
- 688 pages

 [Download Software Fundamentals: Collected Papers by David L ...pdf](#)

 [Read Online Software Fundamentals: Collected Papers by David ...pdf](#)

Editorial Review

From the Inside Flap

Daniel M. Hoffman and David M. Weiss

Why Create a Book Around Dave Parnas's Work? It is sometimes said that progress in a scientific discipline can be measured by how quickly its founders are forgotten. Software development, sometimes called software engineering, is not a scientific discipline and is still young: Many of those who formulated fundamental principles in the field are still active in it. Unfortunately, we have the worst of both worlds: Our founders seem dimly remembered, and we are making little progress towards becoming a discipline. Fundamental ideas, such as information hiding and abstraction, are only vaguely understood by those who need them most and are constantly reinvented. Those who practice software development and those who teach software engineering seem uneducated in, and unaware of, the history of their profession. This book is our attempt to provide a view of the work of one of the grandmasters of our field, highlighting the fundamental ideas that he and his colleagues invented and expounded. We hope to provide a reference for those who teach and those who do, giving them both an historical record, a clear explanation of fundamental ideas that will help them in their work, and a set of examples to use and emulate. David L. Parnas is both a clear and creative thinker and an extraordinary expositor of seminal ideas. The issues that he addresses are at the heart of software engineering today; his explanations are still relevant and his solutions, trialed on real systems, transfer to today's software development organizations and environments. Do you need to understand how to organize your software into modules so it can be easily maintained and so that your modules are reusable, whether they are expressed as classes, packages, or other forms? Dave Parnas identified the information hiding principle and showed how to use it to construct workable, reusable modular structures that are stable over time. (See Chapters 2 and 16.) Are you struggling to create APIs to make your software useful to application programmers? Dave Parnas devised the idea (and coined the term) for abstract interfaces, and showed how to design interfaces that provide services without revealing their implementations. (See Chapter 15.) Languages like C++ and Java directly support this idea with abstract classes. Are you wondering how to create your software as a set of layers that define a hierarchical structure that meets your requirements, lets you build your system a few layers at a time, and lets others add to the structure that you have created? Dave Parnas clearly explained what a hierarchical structure is, what some of the important hierarchical structures that we use are, why people often confuse them, and how to create a layered structure that meets your needs. (See Chapter 8.) Do you know that your software is going to exist in many different versions, but are having difficulty designing your software not just to accommodate the different versions, but to take advantage of your situation to make your development process more efficient? Dave Parnas defined program families to help with just this situation and showed how to create them in a cost-effective way. (See Chapters 10 and 14.) Dave has been busy in more than just technical areas. His work includes commentary on the social responsibility of software engineers, both by exposition and by example. His stance on our inability to create trustworthy software for the Strategic Defense Initiative is represented (Chapters 26 and 27), as well as his thoughts on how to teach software engineering (Chapter 31 and 32), and how to make software engineering a profession (Chapters 28 and 33).

Why Did We Pick These Papers? The preceding are just a few examples of the ideas described in the papers that constitute this book. Out of the more than two hundred papers that Dave has published, we selected thirty-two, plus one special one that he did not write, but strongly influenced. We picked technical papers that expressed fundamental ideas that were groundbreaking when they were published, that have an enduring message, and that are models of exposition, and nontechnical papers that had an influence on the opinions of

the time. Some were controversial when published and remain so. An outstanding aspect of Dave's career is his insistence that his ideas be tested on real problems, where one cannot define away the complexity of the world in the interest of devising an elegant solution. Perhaps the best known examples are the operational flight program (OFP) for the U.S. Navy's A-7E aircraft and the shut-down software for the Darlington nuclear power plant. The A-7E project, also known as the Software Cost Reduction (SCR) project, was conducted by Dave and colleagues at the U.S. Naval Research Laboratory (NRL). It was a demonstration of how to apply ideas such as information hiding, abstraction, cooperating sequential processes, deterministic scheduling, program families, formal specification, hierarchical structuring, and undesired event handling to the design of a hard-real-time system. Many of the same approaches now appear in modern designs and modern languages under different names; a few diverse examples are exception handling (Chapter 12) and the observer pattern (Chapter 22). Several years of Dave's time and effort were directed at making the SCR software and its documentation an engineering model of how to develop and document software. The papers derived from the project that appeared in the research literature; such as Chapters 6, 12, 15, 16, 17, 18, and 22, only tell part of the story. The complete set of requirements and design documentation (including what we now term architecture), was published as technical reports by NRL and serve as detailed guides and templates for those wishing to use the ideas.

How Is the Book Organized? This book contains thirty-three papers divided into four sections. Dave has written a short introduction to each section and we have invited a guest author to write an introduction to each paper. Specification and Description contains six papers, focusing on the most important kinds of software engineering documentation and the roles that they play. Relational and tabular documentation are presented in depth, including both the underlying mathematical basis and practical notations suitable for use by working programmers. Design contains thirteen papers, covering the principles and techniques that have been central to Dave's work for the past three decades. Information hiding is emphasized, including the role of information hiding in abstract interfaces, its application in complex systems, and its implications in the design of program families. Concurrency and Scheduling contains two early papers on the use of semaphores and two more recent papers on new approaches to synchronization and scheduling. The latter focus on achieving both good performance and a module structure that supports maintainability and comprehensibility. Finally, Commentary contains ten papers on a wide variety of topics including education, social issues, the role of the engineer, and the status of software engineering as an engineering profession. In the interests of preserving the historical record and of leaving Dave's writing style unperturbed, we have tampered as little as possible with the papers that appear here, only correcting a few typographical errors in most papers.

Why Have Guest Introductions? The papers span the period from the 1970s through the 1990s. Some use old examples and notations that may not seem relevant to today's Internet world. We asked leading members of our field to write short introductions to the papers to explain the papers' historical and modern relevance. Right from the start, we knew that the introductions must be fun to read and worth reading. They must tell the reader something worth knowing that is not in the paper or is not obvious from reading the paper. We were most fortunate in gathering an impressive collection of authors. Some have been involved with Dave since his work at NRL and earlier. Others participated in the SCR Workshops that continued the NRL work. Some have never directly collaborated with Dave. All are excellent writers with special insights about the significance of the papers both at the time of writing and today. All wrote with enthusiasm and skill. The thirty-three paper introductions are an important contribution in their own right. The fact that these people were all willing, indeed eager, to contribute speaks highly of Dave's work. Dave collaborated with us on the selection of the papers in this book. On several occasions he commented that we were likely to get people angry once again. That is the nature of the man and his ideas: insightful, creative, stimulating, provocative. We hope you find that the papers in this book have the same qualities. It is our present to Dave on his sixtieth birthday.

Acknowledgments We would like to say that we had the idea for this book on our own, but it actually originated with Brad Appleton. Thanks, Brad, for giving us the chance to carry out the idea. Organizational and production details for a book of this sort can get quickly out of hand without an experienced professional editor to guide you. Debbie Lafferty at Addison-Wesley has been a cheerful, steadfast guide for us, appreciating the idea for the book from the first, and working with us to make it happen. During the course of production, all of the papers contained herein were retyped. Dorene Brummel happily took on the job of proofreading them, for which we are very grateful. Joanne Glazer Weiss showed outstanding forbearance and support when her husband plunged into this project immediately after finishing his first book. He thanks and loves her. Duck Bay, British Columbia September, 2000

0201703696P04182001

From the Back Cover

David L. Parnas is one of the grandmasters of software engineering. His academic research and industrial collaborations have exerted far-reaching influence on software design and development. His groundbreaking writings capture the essence of the innovations, controversies, challenges, and solutions of the software industry. Together, they constitute the foundation for modern software theory and practice.

This book contains thirty-three of his most influential papers in various areas of software engineering. Leading thinkers in software engineering have contributed short introductions to each paper to provide the historical context surrounding each paper's conception and writing.

Software Fundamentals: Collected Papers by David L. Parnas is a practical guide to key software engineering concepts that belongs in the library of every software professional. It introduces and explains such seminal topics as:

- Relational and tabular documentation
- Information hiding as the basis for modular program construction
- Abstract interfaces that provide services without revealing implementation
- Program families for the efficient development of multiple software versions
- The status of software engineering as a profession
- Why complex software, such as for the Strategic Defense Initiative, is unlikely to work the first time that it is used in the field

As a celebration of one of the fathers of modern software engineering, and as a practical guide to the key concepts underlying software development, ***Software Fundamentals*** is valuable for professionals, especially those who are interested in teaching the fundamentals of software.

David Parnas is highly regarded for his many valuable contributions to software engineering. He developed and applied cutting-edge software technology to the U.S. Navy's A-7E aircraft, and he advised the Atomic Energy Control Board of Canada on the use of safety-critical, real-time software. During his career, he has contributed more than 200 papers to ACM, IEEE, and ICSE publications. He won an ACM "Best Paper" award, two "Most Influential Paper" awards from ICSE, and the 1998 "Outstanding Researcher" award from ACM SIGSOFT. In May 2001, Dr. Parnas was recognized at the International Conference on Software Engineering for his lifetime of outstanding achievements.

About the editors:

Daniel Hoffman is an Associate Professor of Computer Science at the University of Victoria in British Columbia. **David Weiss** is the Director of the Software Technology Research Department at Avaya

Laboratories.

0201703696B04062001

About the Author

Daniel Hoffman is an Associate Professor of Computer Science at the University of Victoria in British Columbia.

David M. Weiss is the Director of the Software Production Research Department at Avaya Laboratories. His technical work has evolved into the invention of processes that incorporate ideas from families, design for change, measurement, precise specification, and technology transfer. The result has been a software production process based on family-oriented abstraction, specification, and translation, known as FAST.

0201703696AB04062001

Users Review

From reader reviews:

Wanda Crane:

This Software Fundamentals: Collected Papers by David L. Parnas book is simply not ordinary book, you have it then the world is in your hands. The benefit you receive by reading this book is definitely information inside this e-book incredible fresh, you will get details which is getting deeper an individual read a lot of information you will get. That Software Fundamentals: Collected Papers by David L. Parnas without we realize teach the one who studying it become critical in considering and analyzing. Don't become worry Software Fundamentals: Collected Papers by David L. Parnas can bring whenever you are and not make your carrier space or bookshelves' turn into full because you can have it inside your lovely laptop even cellphone. This Software Fundamentals: Collected Papers by David L. Parnas having good arrangement in word in addition to layout, so you will not experience uninterested in reading.

Goldie Oleary:

A lot of people always spent their very own free time to vacation or even go to the outside with them friends and family or their friend. Did you know? Many a lot of people spent they free time just watching TV, or even playing video games all day long. In order to try to find a new activity that's look different you can read some sort of book. It is really fun to suit your needs. If you enjoy the book which you read you can spent all day long to reading a guide. The book Software Fundamentals: Collected Papers by David L. Parnas it is very good to read. There are a lot of people who recommended this book. They were enjoying reading this book. When you did not have enough space to develop this book you can buy the e-book. You can m0ore quickly to read this book from your smart phone. The price is not to cover but this book offers high quality.

Rosemary Lafleur:

Playing with family in a park, coming to see the sea world or hanging out with pals is thing that usually you have done when you have spare time, after that why you don't try thing that really opposite from that. One activity that make you not feeling tired but still relaxing, trilling like on roller coaster you have been ride on and with addition info. Even you love Software Fundamentals: Collected Papers by David L. Parnas, you are able to enjoy both. It is great combination right, you still need to miss it? What kind of hangout type is it? Oh can occur its mind hangout fellas. What? Still don't understand it, oh come on its identified as reading friends.

Lily Spivey:

Guide is one of source of information. We can add our understanding from it. Not only for students but native or citizen want book to know the upgrade information of year to year. As we know those guides have many advantages. Beside we add our knowledge, can bring us to around the world. By book Software Fundamentals: Collected Papers by David L. Parnas we can consider more advantage. Don't that you be creative people? For being creative person must love to read a book. Merely choose the best book that suitable with your aim. Don't become doubt to change your life with that book Software Fundamentals: Collected Papers by David L. Parnas. You can more pleasing than now.

**Download and Read Online Software Fundamentals: Collected
Papers by David L. Parnas By Daniel M. Hoffman, David M. Weiss
#PYQKRF0W15G**

Read Software Fundamentals: Collected Papers by David L. Parnas By Daniel M. Hoffman, David M. Weiss for online ebook

Software Fundamentals: Collected Papers by David L. Parnas By Daniel M. Hoffman, David M. Weiss Free PDF d0wnl0ad, audio books, books to read, good books to read, cheap books, good books, online books, books online, book reviews epub, read books online, books to read online, online library, greatbooks to read, PDF best books to read, top books to read Software Fundamentals: Collected Papers by David L. Parnas By Daniel M. Hoffman, David M. Weiss books to read online.

Online Software Fundamentals: Collected Papers by David L. Parnas By Daniel M. Hoffman, David M. Weiss ebook PDF download

Software Fundamentals: Collected Papers by David L. Parnas By Daniel M. Hoffman, David M. Weiss Doc

Software Fundamentals: Collected Papers by David L. Parnas By Daniel M. Hoffman, David M. Weiss Mobipocket

Software Fundamentals: Collected Papers by David L. Parnas By Daniel M. Hoffman, David M. Weiss EPub