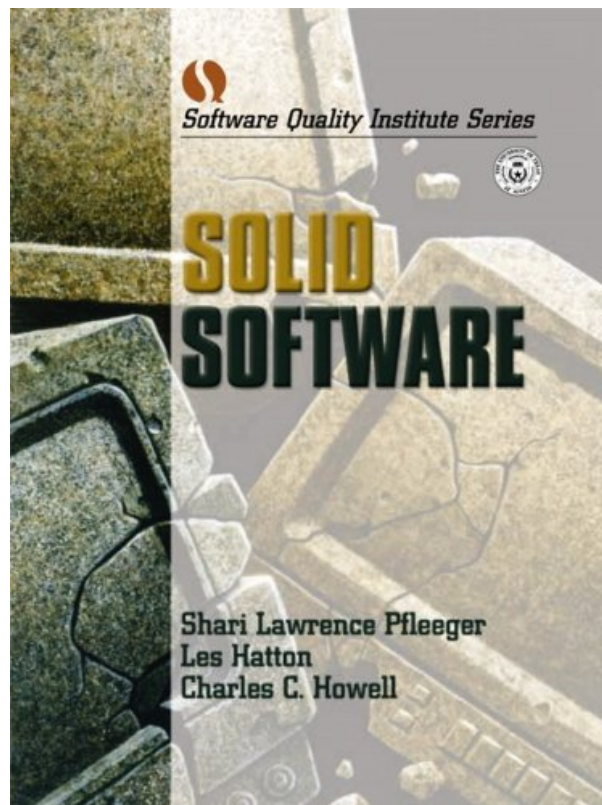
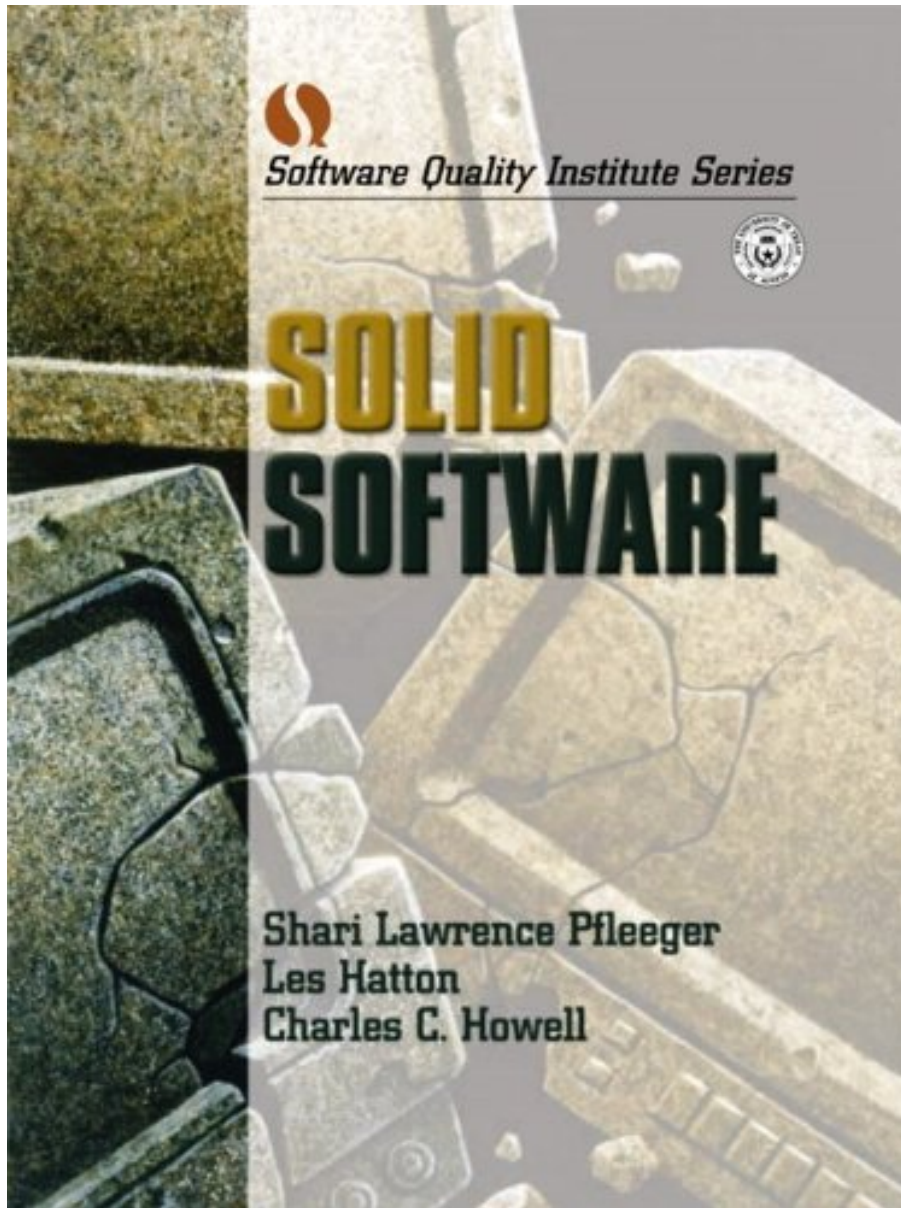


SOLID SOFTWARE BY SHARI LAWRENCE PFLEEGER, LES HATTON, CHARLES C. HOWELL



**DOWNLOAD EBOOK : SOLID SOFTWARE BY SHARI LAWRENCE PFLEEGER,
LES HATTON, CHARLES C. HOWELL PDF**





Click link bellow and free register to download ebook:

**SOLID SOFTWARE BY SHARI LAWRENCE PFLEEGER, LES HATTON, CHARLES C.
HOWELL**

[DOWNLOAD FROM OUR ONLINE LIBRARY](#)

SOLID SOFTWARE BY SHARI LAWRENCE PFLEEGER, LES HATTON, CHARLES C. HOWELL PDF

Just for you today! Discover your preferred publication here by downloading and install as well as getting the soft data of guide **Solid Software By Shari Lawrence Pfleeger, Les Hatton, Charles C. Howell** This is not your time to typically likely to guide stores to buy a publication. Below, selections of e-book Solid Software By Shari Lawrence Pfleeger, Les Hatton, Charles C. Howell and also collections are readily available to download and install. Among them is this Solid Software By Shari Lawrence Pfleeger, Les Hatton, Charles C. Howell as your preferred publication. Getting this book Solid Software By Shari Lawrence Pfleeger, Les Hatton, Charles C. Howell by on the internet in this website can be recognized now by going to the link web page to download. It will be very easy. Why should be here?

From the Back Cover

The practical guide to evaluating and improving the quality of mission-critical software.

Large software systems will never be perfect, but decision makers need better ways to evaluate and enhance software quality—especially where software is mission critical or used in life-or-death environments. Solid Software presents realistic techniques for analyzing and improving the quality and robustness of any software system or software-intensive product.

Solid Software isn't theoretical: it's a relentlessly practical decision maker's guide to making intelligent, responsible trade-offs that lead to the best software at the best cost.

- Understand what levels of quality are reasonable to expect at every stage of the software life cycle, including development, deployment, and maintenance
- Discover the key "design leverage points" that lead to robust software
- Learn sophisticated new ways to predict software quality and assess systems in production
- Make the most of hazard analysis, testing, design analysis, reviews, static code analysis, and other techniques
- Choose the best tools—and use them more effectively

Solid Software draws upon dozens of real-world examples, based on the authors' extensive experience as software quality consultants, and interviews with key software decision makers worldwide. Whether you're a developer, project manager, architect, executive, manager, or regulator, it's your single source for improving software quality—in the real world.

About the Author

SHARI LAWRENCE PFLEEGER is President of Systems/Software, Inc., a leading software engineering consultancy. She has been founder/director of Howard University's Center for Research in Evaluating

Software Technology, visiting scientist at the City University (London) Centre for Software Reliability, principal scientist at MITRE Corporation's Software Engineering Center, and manager of the measurement program at Contel Technology Center.

Excerpt. © Reprinted by permission. All rights reserved.

Preface They constantly try to escape...by dreaming of systems so perfect that no one will need to be good. T. S. Eliot, Choruses from *The Rock*, VI

You're in charge. The buck or pound or peso stops with you. Your developers are to build a safety- or business-critical system, and you have a lot of questions to answer. How solid is the software supposed to be? How will you be able to demonstrate to the clients that it is as solid as they wish it? How will your developers be able to demonstrate to you that the software will be solid and (eventually) is solid, so that you can give assurances to your boss and to the clients? You know that there is (unfortunately) no easy solution to the challenges you face—no "eat all the cake you want and still lose weight diet" for developing critical software. But you can take advantage of the experience of others in a wide range of critical software projects.

There are many books for developers and much research about the theoretical ways to build software that does what it is supposed to do (and nothing more, like a virus or Trojan horse) and does it in a consistent, predictable, and safe way. There are theoretical books about how to evaluate the software before you field it or deliver it. But with safety-critical systems, many of which would need over 100,000 years of failure-free testing to confirm required reliability, theory is not enough. You need to know what is practical, what is available right now, and what can give you confidence in the quality of the requirements, design, code, and test procedures.

This is the book for you. In *Solid Software* we describe the problem and suggest what you can and cannot expect from your developers, their techniques and tools, and their software. We discuss what you should know about software quality—not just about the faults and failures but also how the quality affects your company's bottom line. Then we introduce eight techniques, one chapter at a time, that can help to increase your confidence—and that of your clients—in how the software will perform: Hazard analysis Testing Design analysis Prediction Reviews Static code analysis Configuration management and change control Tools

None of these techniques is foolproof, but each one helps you to manage the risks inherent in producing such critical code. When properly applied, each one gives you added confidence that you have addressed key points of vulnerability. When used in concert, these techniques stabilize the software, making it less likely to fail and more easy to change and expand.

SOLID SOFTWARE BY SHARI LAWRENCE PFLEEGER, LES HATTON, CHARLES C. HOWELL PDF

[Download: SOLID SOFTWARE BY SHARI LAWRENCE PFLEEGER, LES HATTON, CHARLES C. HOWELL PDF](#)

Reserve **Solid Software By Shari Lawrence Pfleeger, Les Hatton, Charles C. Howell** is one of the precious well worth that will make you consistently rich. It will not suggest as rich as the money offer you. When some people have lack to encounter the life, individuals with several publications in some cases will be wiser in doing the life. Why ought to be e-book Solid Software By Shari Lawrence Pfleeger, Les Hatton, Charles C. Howell It is in fact not meant that e-book Solid Software By Shari Lawrence Pfleeger, Les Hatton, Charles C. Howell will certainly provide you power to get to every little thing. The e-book is to read and what we suggested is guide that is checked out. You could likewise see how the book qualifies Solid Software By Shari Lawrence Pfleeger, Les Hatton, Charles C. Howell as well as varieties of book collections are giving here.

Yet right here, we will reveal you astonishing thing to be able consistently check out guide *Solid Software By Shari Lawrence Pfleeger, Les Hatton, Charles C. Howell* anywhere as well as whenever you happen as well as time. The publication Solid Software By Shari Lawrence Pfleeger, Les Hatton, Charles C. Howell by just can aid you to recognize having guide to review every time. It will not obligate you to constantly bring the thick publication wherever you go. You could merely keep them on the gizmo or on soft documents in your computer to always review the area at that time.

Yeah, hanging around to check out the book Solid Software By Shari Lawrence Pfleeger, Les Hatton, Charles C. Howell by online could also provide you favorable session. It will ease to communicate in whatever condition. This means can be more appealing to do and also much easier to review. Now, to obtain this Solid Software By Shari Lawrence Pfleeger, Les Hatton, Charles C. Howell, you can download in the web link that we provide. It will certainly aid you to obtain very easy way to download the book Solid Software By Shari Lawrence Pfleeger, Les Hatton, Charles C. Howell.

SOLID SOFTWARE BY SHARI LAWRENCE PFLEEGER, LES HATTON, CHARLES C. HOWELL PDF

Solid Software presents realistic techniques for analyzing and improving the quality and robustness of any software system or software-intensive product. Solid Software isn't theoretical: it's a relentlessly practical decision maker's guide to making intelligent, responsible trade-offs that lead to the best software at the best cost. Solid Software draws upon dozens of real-world examples, based on the author's extensive experience as software quality consultants, and interviews with key software decision makers worldwide. Whether you're a developer, project manager, architect, executive, manager, or regulator, it's your single source for improving software quality in the real world.

- Sales Rank: #4861535 in Books
- Color: Grey
- Published on: 2001-07-12
- Original language: English
- Number of items: 1
- Dimensions: 9.16" h x .91" w x 6.92" l, 1.41 pounds
- Binding: Paperback
- 336 pages

From the Back Cover

The practical guide to evaluating and improving the quality of mission-critical software.

Large software systems will never be perfect, but decision makers need better ways to evaluate and enhance software quality—especially where software is mission critical or used in life-or-death environments. Solid Software presents realistic techniques for analyzing and improving the quality and robustness of any software system or software-intensive product.

Solid Software isn't theoretical: it's a relentlessly practical decision maker's guide to making intelligent, responsible trade-offs that lead to the best software at the best cost.

- Understand what levels of quality are reasonable to expect at every stage of the software life cycle, including development, deployment, and maintenance
- Discover the key "design leverage points" that lead to robust software
- Learn sophisticated new ways to predict software quality and assess systems in production
- Make the most of hazard analysis, testing, design analysis, reviews, static code analysis, and other techniques
- Choose the best tools—and use them more effectively

Solid Software draws upon dozens of real-world examples, based on the authors' extensive experience as software quality consultants, and interviews with key software decision makers worldwide. Whether you're a developer, project manager, architect, executive, manager, or regulator, it's your single source for improving software quality—in the real world.

About the Author

SHARI LAWRENCE PFLEEGER is President of Systems/Software, Inc., a leading software engineering consultancy. She has been founder/director of Howard University's Center for Research in Evaluating Software Technology, visiting scientist at the City University (London) Centre for Software Reliability, principal scientist at MITRE Corporation's Software Engineering Center, and manager of the measurement program at Contel Technology Center.

Excerpt. © Reprinted by permission. All rights reserved.

Preface They constantly try to escape...by dreaming of systems so perfect that no one will need to be good. T. S. Eliot, Choruses from *The Rock*, VI

You're in charge. The buck or pound or peso stops with you. Your developers are to build a safety- or business-critical system, and you have a lot of questions to answer. How solid is the software supposed to be? How will you be able to demonstrate to the clients that it is as solid as they wish it? How will your developers be able to demonstrate to you that the software will be solid and (eventually) is solid, so that you can give assurances to your boss and to the clients? You know that there is (unfortunately) no easy solution to the challenges you face—no "eat all the cake you want and still lose weight diet" for developing critical software. But you can take advantage of the experience of others in a wide range of critical software projects.

There are many books for developers and much research about the theoretical ways to build software that does what it is supposed to do (and nothing more, like a virus or Trojan horse) and does it in a consistent, predictable, and safe way. There are theoretical books about how to evaluate the software before you field it or deliver it. But with safety-critical systems, many of which would need over 100,000 years of failure-free testing to confirm required reliability, theory is not enough. You need to know what is practical, what is available right now, and what can give you confidence in the quality of the requirements, design, code, and test procedures.

This is the book for you. In *Solid Software* we describe the problem and suggest what you can and cannot expect from your developers, their techniques and tools, and their software. We discuss what you should know about software quality—not just about the faults and failures but also how the quality affects your company's bottom line. Then we introduce eight techniques, one chapter at a time, that can help to increase your confidence—and that of your clients—in how the software will perform: Hazard analysis Testing Design analysis Prediction Reviews Static code analysis Configuration management and change control Tools

None of these techniques is foolproof, but each one helps you to manage the risks inherent in producing such critical code. When properly applied, each one gives you added confidence that you have addressed key points of vulnerability. When used in concert, these techniques stabilize the software, making it less likely to fail and more easy to change and expand.

Most helpful customer reviews

7 of 7 people found the following review helpful.

Solid Software, mushy in the middle

By R. Dameron

I teach software engineering at the University of Colorado. I have begun a 3 course graduate series and am using *Solid Software* as one of two texts in the first course, *Software Engineering of Stand Alone Programs*. (Subsequent courses cover multiprogramming and distributed systems.) I have found *Solid Software* to cover the right topics but not to the right level of detail for my purposes. It is hard to find a general book on software engineering that covers adequately enough development factors that contribute to robustness such

as requirements and design reviews, static and dynamic testing, etc. I was pleased when I read through the table of contents and saw the 9 areas addressed by the book. However, it is written to, say, first level managers of programming teams, not to the programmers themselves. On many topics, there are excerpts from books or papers that give a high level "hit". The good news is that the reference list at the end of each chapter includes excellent references. I think it's fair to say that my students' (all with industry experience) reaction is that it's not quite detailed enough to really understand. They are not expecting a how-to guide but ... more than this level. But you *can* follow the reference trail to get to more substance. If you ARE a 1st level manager, wondering what more can be done, what is reasonable to ask for, then this may be a great book for you.

See all 1 customer reviews...

SOLID SOFTWARE BY SHARI LAWRENCE PFLEEGER, LES HATTON, CHARLES C. HOWELL PDF

The books Solid Software By Shari Lawrence Pfleeger, Les Hatton, Charles C. Howell, from straightforward to complicated one will be a really helpful works that you could require to change your life. It will certainly not provide you unfavorable declaration unless you don't get the significance. This is definitely to do in checking out a book to get rid of the meaning. Commonly, this e-book qualified Solid Software By Shari Lawrence Pfleeger, Les Hatton, Charles C. Howell is read considering that you actually similar to this sort of e-book. So, you can get simpler to recognize the impression and also definition. Once again to always keep in mind is by reviewing this e-book **Solid Software By Shari Lawrence Pfleeger, Les Hatton, Charles C. Howell**, you could fulfil hat your interest begin by completing this reading book.

From the Back Cover

The practical guide to evaluating and improving the quality of mission-critical software.

Large software systems will never be perfect, but decision makers need better ways to evaluate and enhance software quality—especially where software is mission critical or used in life-or-death environments. Solid Software presents realistic techniques for analyzing and improving the quality and robustness of any software system or software-intensive product.

Solid Software isn't theoretical: it's a relentlessly practical decision maker's guide to making intelligent, responsible trade-offs that lead to the best software at the best cost.

- Understand what levels of quality are reasonable to expect at every stage of the software life cycle, including development, deployment, and maintenance
- Discover the key "design leverage points" that lead to robust software
- Learn sophisticated new ways to predict software quality and assess systems in production
- Make the most of hazard analysis, testing, design analysis, reviews, static code analysis, and other techniques
- Choose the best tools—and use them more effectively

Solid Software draws upon dozens of real-world examples, based on the authors' extensive experience as software quality consultants, and interviews with key software decision makers worldwide. Whether you're a developer, project manager, architect, executive, manager, or regulator, it's your single source for improving software quality—in the real world.

About the Author

SHARI LAWRENCE PFLEEGER is President of Systems/Software, Inc., a leading software engineering consultancy. She has been founder/director of Howard University's Center for Research in Evaluating Software Technology, visiting scientist at the City University (London) Centre for Software Reliability, principal scientist at MITRE Corporation's Software Engineering Center, and manager of the measurement program at Contel Technology Center.

Excerpt. © Reprinted by permission. All rights reserved.

Preface They constantly try to escape...by dreaming of systems so perfect that no one will need to be good. T. S. Eliot, Choruses from *The Rock*, VI

You're in charge. The buck or pound or peso stops with you. Your developers are to build a safety- or business-critical system, and you have a lot of questions to answer. How solid is the software supposed to be? How will you be able to demonstrate to the clients that it is as solid as they wish it? How will your developers be able to demonstrate to you that the software will be solid and (eventually) is solid, so that you can give assurances to your boss and to the clients? You know that there is (unfortunately) no easy solution to the challenges you face—no "eat all the cake you want and still lose weight diet" for developing critical software. But you can take advantage of the experience of others in a wide range of critical software projects.

There are many books for developers and much research about the theoretical ways to build software that does what it is supposed to do (and nothing more, like a virus or Trojan horse) and does it in a consistent, predictable, and safe way. There are theoretical books about how to evaluate the software before you field it or deliver it. But with safety-critical systems, many of which would need over 100,000 years of failure-free testing to confirm required reliability, theory is not enough. You need to know what is practical, what is available right now, and what can give you confidence in the quality of the requirements, design, code, and test procedures.

This is the book for you. In *Solid Software* we describe the problem and suggest what you can and cannot expect from your developers, their techniques and tools, and their software. We discuss what you should know about software quality—not just about the faults and failures but also how the quality affects your company's bottom line. Then we introduce eight techniques, one chapter at a time, that can help to increase your confidence—and that of your clients—in how the software will perform: Hazard analysis Testing Design analysis Prediction Reviews Static code analysis Configuration management and change control Tools

None of these techniques is foolproof, but each one helps you to manage the risks inherent in producing such critical code. When properly applied, each one gives you added confidence that you have addressed key points of vulnerability. When used in concert, these techniques stabilize the software, making it less likely to fail and more easy to change and expand.

Just for you today! Discover your preferred publication here by downloading and install as well as getting the soft data of guide **Solid Software By Shari Lawrence Pfleeger, Les Hatton, Charles C. Howell** This is not your time to typically likely to guide stores to buy a publication. Below, selections of e-book *Solid Software By Shari Lawrence Pfleeger, Les Hatton, Charles C. Howell* and also collections are readily available to download and install. Among them is this *Solid Software By Shari Lawrence Pfleeger, Les Hatton, Charles C. Howell* as your preferred publication. Getting this book *Solid Software By Shari Lawrence Pfleeger, Les Hatton, Charles C. Howell* by on the internet in this website can be recognized now by going to the link web page to download. It will be very easy. Why should be here?