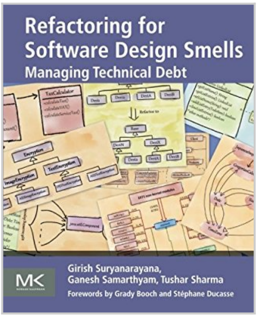
Refactoring for Software Design Smells
Managing Technical Debt
MK
Girish Suryanarayana, Ganesh Samarthyam, Tushar Sharma
Forewords by Grady Booch and Stéphane Ducasse

Refactoring for Software Design Smells: Managing Technical Debt

By Girish Suryanarayana, Ganesh Samarthyam, Tushar Sharma

Refactoring for Software Design Smells: Managing Technical Debt By Girish Suryanarayana, Ganesh Samarthyam, Tushar Sharma

Awareness of design smells – indicators of common design problems – helps developers or software engineers understand mistakes made while designing, what design principles were overlooked or misapplied, and what principles need to be applied properly to address those smells through refactoring. Developers and software engineers may "know" principles and patterns, but are not aware of the "smells" that exist in their design because of wrong or mis-application of principles or patterns. These smells tend to contribute heavily to technical debt – further time owed to fix projects thought to be complete – and need to be addressed via proper refactoring.

Refactoring for Software Design Smells presents 25 structural design smells, their role in identifying design issues, and potential refactoring solutions. Organized across common areas of software design, each smell is presented with diagrams and examples illustrating the poor design practices and the problems that result, creating a catalog of nuggets of readily usable information that developers or engineers can apply in their projects. The authors distill their research and experience as consultants and trainers, providing insights that have been used to improve refactoring and reduce the time and costs of managing software projects. Along the way they recount anecdotes from actual projects on which the relevant smell helped address a design issue.

- Contains a comprehensive catalog of 25 structural design smells (organized around four fundamental design principles) that contribute to technical debt in software projects
- Presents a unique naming scheme for smells that helps understand the cause of a smell as well as points toward its potential refactoring
- Includes illustrative examples that showcase the poor design practices underlying a smell and the problems that result
- Covers pragmatic techniques for refactoring design smells to manage technical debt and to create and maintain high-quality software in practice
- Presents insightful anecdotes and case studies drawn from the trenches of real-world projects

Refactoring for Software Design Smells: Managing Technical Debt

By Girish Suryanarayana, Ganesh Samarthyam, Tushar Sharma

Refactoring for Software Design Smells: Managing Technical Debt By Girish Suryanarayana, Ganesh Samarthyam, Tushar Sharma

Awareness of design smells – indicators of common design problems – helps developers or software engineers understand mistakes made while designing, what design principles were overlooked or misapplied, and what principles need to be applied properly to address those smells through refactoring. Developers and software engineers may "know" principles and patterns, but are not aware of the "smells" that exist in their design because of wrong or mis-application of principles or patterns. These smells tend to contribute heavily to technical debt – further time owed to fix projects thought to be complete – and need to be addressed via proper refactoring.

Refactoring for Software Design Smells presents 25 structural design smells, their role in identifying design issues, and potential refactoring solutions. Organized across common areas of software design, each smell is presented with diagrams and examples illustrating the poor design practices and the problems that result, creating a catalog of nuggets of readily usable information that developers or engineers can apply in their projects. The authors distill their research and experience as consultants and trainers, providing insights that have been used to improve refactoring and reduce the time and costs of managing software projects. Along the way they recount anecdotes from actual projects on which the relevant smell helped address a design issue.

- Contains a comprehensive catalog of 25 structural design smells (organized around four fundamental design principles) that contribute to technical debt in software projects
- Presents a unique naming scheme for smells that helps understand the cause of a smell as well as points toward its potential refactoring
- Includes illustrative examples that showcase the poor design practices underlying a smell and the problems that result
- Covers pragmatic techniques for refactoring design smells to manage technical debt and to create and maintain high-quality software in practice
- Presents insightful anecdotes and case studies drawn from the trenches of real-world projects

Refactoring for Software Design Smells: Managing Technical Debt By Girish Suryanarayana, Ganesh Samarthyam, Tushar Sharma Bibliography

- Sales Rank: #860154 in Books
- Published on: 2014-11-17
- Released on: 2014-11-03
- Original language: English
- Number of items: 1
- Dimensions: 9.25" h x .59" w x 7.50" l, 1.59 pounds
- Binding: Paperback

- 258 pages

Editorial Review

Review

"Given its practical orientation and the variety of real-world examples offered throughout the book, this is a must-have for any practicing software engineer, developer, software architect, or anyone else interested in software design."**--Computing Reviews**

"... a delightful, engaging, actionable read... you have in your hand a veritable field guide of smells... one of the more interesting and complex expositions of software smells you will ever find... The concept of technical debt is central to understanding the forces that weigh upon systems, for it often explains where and how and why a system is stressed. What delights me about this present book is its focus on technical debt and refactoring as the actionable means to attend to it." **--From the foreword by Grady Booch, IBM Fellow and Chief Scientist for Software Engineering, IBM Research**

"Evolving software inevitably accumulates technical debt, making maintenance increasingly painful and expensive. The authors, based on their extensive experience, categorize the major design problems (smells) that come up in software, and lucidly explain how these can be solved with appropriate refactoring." **--Diomidis Spinellis, Author of “Code Reading” and “Code Quality”, Addison-Wesley Professional**

"...the book I would have loved to write... *Refactoring for Software Design Smells* is an excellent book. It is another milestone that professionals will use... I’m sure that you will learn a lot from it and that you will enjoy it." **--From the foreword by Stéphane Ducasse, Co-author of Object-Oriented Reengineering Patterns, Morgan Kaufmann**

From the Back Cover

Refactoring for Software Design Smells presents 25 structural design smells, how they uncover mistakes made while designing, what design principles were overlooked or misapplied, and what principles need to be applied properly to address those smells through refactoring. Organized across common areas of software design, each smell is presented with diagrams and examples illustrating the poor design practices and the problems that result. The authors share insights that have been used to improve refactoring as well as reduce technical debt in software projects. Along the way they recount anecdotes from actual projects on which the relevant smell helped address a design issue.

About the Author

Girish Suryanarayana is currently a Senior Research Scientist at Research and Technology Center, Siemens Technology and Services Pvt. Ltd. Bangalore, India. At Siemens, he is involved in providing architectural guidance to software development teams, pursuing research in topics related to software architecture and cloud computing, and conducting internal software design and architecture training. Girish is a member of the IEEE Software Advisory Board and is the General Chair for the IEEE Software Experts Summit 2014 conference. He actively contributes to software engineering conferences. In 2013, he was on the program committee for the Software Engineering In Practice (SEIP) track at the 35th International Conference on Software Engineering (ICSE). Girish received a PhD in Information and Computer Science from the University of California, Irvine, in 2007. His research interests include software architecture, design patterns, design smells, refactoring, cloud computing, and reputation-based trust management systems. He is an IEEE-certified Software Engineering Certified Instructor (SECI) and regularly conducts training for the IEEE SWEBOK Certificate Program (SCP) and IEEE Certified Software Development Associate (CSDA) programs. He has also helped contribute course material for the IEEE's SWEBOK Certificate Program (SCP). He is regularly invited by local universities to deliver guest lectures on software architecture and design topics. Additionally, he is the General Chair for IEEE SES (Software Expert Summit) 2014 which is being organized by IEEE Software.

Ganesh Samarthyam has more than 11 years of experience in the IT industry. He is an independent consultant and corporate trainer based in Bangalore, India. Earlier, he worked for the Software Architecture and Development team in Siemens Corporate Research & Technologies in Bangalore. His work involved providing consultancy to development teams on code and design quality, pursuing applied research in the areas of software design, and conducting trainings on topics related to software design and refactoring and quality-driven development. Prior to Siemens, he worked in Hewlett-Packard's C++ compiler team for 5 years where he also represented HP in the ANSI/ISO C++ standardization committee. He is an IEEE-certified Software Engineering Certified Instructor (SECI) and regularly conducts training for the IEEE SWEBOK Certificate Program (SCP) and IEEE Certified Software Development Associate (CSDA) program. He has also helped contribute course material for the IEEE's SWEBOK Certificate Program (SCP). Ganesh has authored/co-authored several books on programming. His latest book titled "Oracle Certified Professional Java SE 7 Programmer Exams 1Z0-804 and 1Z0-805" (co-authored with Tushar Sharma) has been published by Apress in 2013. In his free time, he also regularly contributes to a column on software programming and design in “Open Source For You” magazine. His areas of interest include design smells and refactoring, bug patterns, software quality assessment methods, and programming languages. He can be reached at sgganesh@gmail.com.

Tushar Sharma is currently a Technical Expert in the Software Architecture and Development group at the Research and Technology Center, Siemens Technology and Services Pvt. Ltd., Bangalore, India. His work at Siemens involves researching and providing consultation on topics related to software design, refactoring, testability, design patterns, and change impact analysis. He has a Masters degree in Computer Science from the Indian Institute of Technology-Madras (IIT-M), Chennai, India, where he specialized in design patterns and refactoring. Tushar conducts several internal training sessions on topics related to software design and refactoring. He has been actively pursuing his research interests that resulted in several patents, research papers, and tools related to software design and refactoring. He is also the co-author of the book “Oracle Certified Professional Java SE 7 Programmer Exams 1Z0-804 and 1Z0-805" published by Apress in 2013. Additionally, he is the Local Publicity Chair for IEEE SES (Software Expert Summit) 2014.

Users Review

From reader reviews:

Norman Williams:

Book is to be different for each grade. Book for children until adult are different content. We all know that that book is very important normally. The book Refactoring for Software Design Smells: Managing Technical Debt had been making you to know about other knowledge and of course you can take more information. It is rather advantages for you. The reserve Refactoring for Software Design Smells: Managing Technical Debt is not only giving you far more new information but also to get your friend when you feel bored. You can spend your current spend time to read your reserve. Try to make relationship with the book Refactoring for Software Design Smells: Managing Technical Debt. You never truly feel lose out for everything when you read some books.

Donald Andrews:

Often the book Refactoring for Software Design Smells: Managing Technical Debt will bring someone to the new experience of reading a new book. The author style to clarify the idea is very unique. In the event you try to find new book to read, this book very suited to you. The book Refactoring for Software Design Smells: Managing Technical Debt is much recommended to you to see. You can also get the e-book in the official web site, so you can more readily to read the book.

Jennie Groth:

Playing with family inside a park, coming to see the water world or hanging out with buddies is thing that usually you may have done when you have spare time, after that why you don't try point that really opposite from that. 1 activity that make you not experience tired but still relaxing, trilling like on roller coaster you are ride on and with addition associated with. Even you love Refactoring for Software Design Smells: Managing Technical Debt, you are able to enjoy both. It is great combination right, you still would like to miss it? What kind of hangout type is it? Oh come on its mind hangout folks. What? Still don't buy it, oh come on its known as reading friends.

Clifford Roselli:

It is possible to spend your free time to read this book this guide. This Refactoring for Software Design Smells: Managing Technical Debt is simple bringing you can read it in the park your car, in the beach, train and also soon. If you did not include much space to bring the actual printed book, you can buy often the e-book. It is make you better to read it. You can save typically the book in your smart phone. Consequently there are a lot of benefits that you will get when you buy this book.

Download and Read Online Refactoring for Software Design Smells: Managing Technical Debt By Girish Suryanarayana, Ganesh Samarthyam, Tushar Sharma #BHPCASQR5M3

Read Refactoring for Software Design Smells: Managing Technical Debt By Girish Suryanarayana, Ganesh Samarthyam, Tushar Sharma for online ebook

Refactoring for Software Design Smells: Managing Technical Debt By Girish Suryanarayana, Ganesh Samarthyam, Tushar Sharma 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 Refactoring for Software Design Smells: Managing Technical Debt By Girish Suryanarayana, Ganesh Samarthyam, Tushar Sharma books to read online.

Online Refactoring for Software Design Smells: Managing Technical Debt By Girish Suryanarayana, Ganesh Samarthyam, Tushar Sharma ebook PDF download

Refactoring for Software Design Smells: Managing Technical Debt By Girish Suryanarayana, Ganesh Samarthyam, Tushar Sharma Doc

Refactoring for Software Design Smells: Managing Technical Debt By Girish Suryanarayana, Ganesh Samarthyam, Tushar Sharma Mobipocket

Refactoring for Software Design Smells: Managing Technical Debt By Girish Suryanarayana, Ganesh Samarthyam, Tushar Sharma EPub

BHPCASQR5M3: Refactoring for Software Design Smells: Managing Technical Debt By Girish Suryanarayana, Ganesh Samarthyam, Tushar Sharma