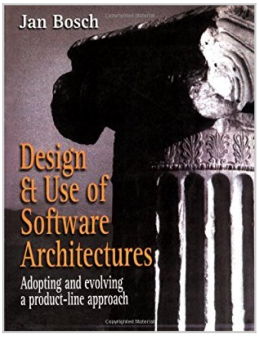

Design and Use of Software Architectures: Adopting and Evolving a Product-Line Approach

By Jan Bosch

Design and Use of Software Architectures: Adopting and Evolving a Product-Line Approach By Jan Bosch

A practical guide to designing and implementing software architectures.

Design and Use of Software Architectures: Adopting and Evolving a Product-Line Approach

By Jan Bosch

Design and Use of Software Architectures: Adopting and Evolving a Product-Line Approach By Jan Bosch

A practical guide to designing and implementing software architectures.

Design and Use of Software Architectures: Adopting and Evolving a Product-Line Approach By Jan Bosch Bibliography

- Sales Rank: #2440803 in Books
- Published on: 2000-05-29
- Original language: English
- Number of items: 1
- Dimensions: 9.20" h x .77" w x 7.38" l, 1.36 pounds
- Binding: Paperback
- 368 pages

Editorial Review

From the Back Cover

In the ever-changing world of software development, organizations are gradually realizing the need for an explicit software architecture for their systems. And software engineers are potentially faced with the challenge of developing multiple, inter-related systems whilst pressured by financial and lead-time constraints. It is thus essential to establish a well-defined design method in order to build high-quality, reliable, and easily maintainable individual or families of systems - the question is how to approach this?

This book provides an achievable answer. The author proposes a method for designing software architectures, and product line architectures, which is based on his experience in industry and research,

The first part of the book introduces the design method, examining its three key phases - functionality-based architecture design, architecture assessment and architecture transformation - in detail. The second half of the book explores the concept of software product lines, incorporating elements of architectural design, component-based software engineering and reuse. It takes you through the process of designing, developing and evolving a product-line approach, including the development of software architecture and components for the family, instantiation of family members and evolution of assets. It also examines the pros and cons of a number of organizational models that can be applied when putting a software product line approach in to practice.

If you are a software architect or engineer involved in designing software systems architectures, this book will give you the resources you need for success.

· features real-life case studies covering control and real-time systems, networking, and telecommunications industry examples to illustrate how the method and processes work in practice
· provides a systematic approach that employs both qualitative and quantitative techniques for assessments
· contains key chapters on approaches to component development, and use of object-oriented frameworks as components in software product lines
· includes sections that can be read as stand-alone, depending on your level of knowledge and experience and your specific area of interest.

0201674947B04062001

About the Author

Jan Bosch is Professor of Software Engineering at the University of Karlskrona/Ronneby, Sweden. He is a key member of the RISE and ARCS groups, through which he has been involved in extensive research in the software architecture area, in conjunction with a number of companies in industry. He has also authored many articles and conference proceedings and given workshops and tutorials at a number of international conferences

Preface

Software has entered virtually all parts of society, ranging from basic utilities such as electricity generation and distribution and telecommunications to personal devices such as mobile phones and cars. Whereas traditionally the competitiveness of a company was defined by its ability to construct and maintain mechanical systems, and later on hardware systems, currently it is the ability to develop and evolve software systems efficiently and effectively which is central.

These developments imply an increasing level of responsibility on the software engineering community. With the increase in integration between software systems, failures in one system may have effects which extend beyond the system itself and affect other systems, with possible consequences in the physical world that were not conceivable even a decade ago. On the other hand, software has allowed for unprecedented flexibility and agility in organizations and systems that have many positive effects on society, organizations and individuals.

Despite the success of software, software engineering still has many challenges to address. In particular, the following primary objectives can be identified: to drastically decrease the cost of developing and maintaining software and the time-to-market of new software products and to improve (and manage) the quality attributes of software products.

Although software systems have always had an architecture, during the last decade the notion of an explicit software architecture has been recognized as being important. One can identify three purposes for an explicitly defined software architecture. First, it allows for early assessment of and design for the quality attributes of a software system. Second, the software architecture represents a concrete artefact that can be used for discussions with and between stakeholders. Finally, it defines the architectural components and their interactions, which facilitates reuse in general and software product lines in particular.

This book is concerned with two aspects of software architecture: the design of software architectures and software product lines. In Part I of the book, we present our software architectural design method. This method has been developed and refined through our involvement in a number of software architecture design projects. The three main projects are described as case studies in this book. In Part II of the book, we present the notion of software product lines and the

process of adopting and evolving a product-line approach to software development. Again, our approach has been shaped
by the co-operation projects that we have had with a number of companies which use software product lines. Consequently, the contents of Part II are shaped by the experiences from these co-operation projects.

Foreword

Our nice little puppy has really grow and now needs a doghouse. A few boards, some two-by-fours and leftover cedar
shingles, a handful of nails, and voilà! Our puppy has a brand new home. Now, let's say you didn't get it quite right at
first, so you made some adjustments along the way, then even more a week later. You could even imagine applying some
of these techniques to your house if, like me, you live in an area where most houses are made of wood, but you'd be hard
pressed to make a living at it. Some of the fixes along the way may have drastic consequences, or may violate the building
codes. Now try taking this 'build and fix' approach to a sky-scraper, and I'm afraid you'd have to completely rethink your
strategy.1

If you think of the parallel between construction and software development, the fact is that most software today is still
developed and built according to the 'code and fix' approach - by writing the code, summarily testing it, and shipping it
to the customer to use, or to finish the testing. Also in the software world, all doghouses and most log cabins have already
been built, and now companies are mostly tackling the sky scrapers. The 'code and fix' approach does not work and
development organizations realize that they need to engineer their software intensive systems.2 They go from developing
'one-off' systems, to multiple instances, to families of systems or product lines, to leveraging their efforts, as you cannot
build these sky-scrapers from scratch all the time.

This is where software architecture comes into play.

All software intensive systems have an architecture, but, unlike building architecture, this architecture is often hidden,
fuzzy, and seems to be produced more by black magic or by accident than by human intent or design. It was only five years
ago that Mary Shaw and David Garlan published their book Software Architecture: Perspectives on an Emerging
Discipline3 and although this discipline has made some progress, not much has been published since. It has been slow to
emerge as a mature software engineering discipline.

There are three main aspects where software architecture needs to make progress to establish itself:

n Architecture representation - By defining how to represent the architecture of software intensive systems

and reaching
some industry-wide consensus, we'll be able to communicate architectural designs or blueprints, to reason about them,
and to evaluate and compare architectures. This is where the future standard IEEE 1471 on architecture representation
will fill a void. This is also a place where the Unified Modeling Language (UML) has a role to play as a uniform notation
for architectural blueprints.

n Architectural process - By defining the methods to design and assess architectures, focusing on quality attributes (the
non-functional or 'afunctional' requirements), and addressing them in a systematic fashion. The architectural design
approaches need to be supported by a matching organization that takes architecture as a key function, and understands its
value and how it flows into other areas, such as planning, project management, product management, design or
deployment.

n Architectural assets - By collecting, cataloging, and presenting fragments of successful architectures, or even complete
architectures, whether they are called patterns, frameworks, components, mechanisms or standards, we will enable
software development organizations to design architecture without re-inventing the wheel. This will also foster better
communication across the industry, allowing practitioners to simply name the pattern rather than describing it completely.
However, as architectures are usually prized company assets, many companies are reluctant to exhibit their architectural
assets in a public forum. They'd rather patent them or carefully hide them in their products.

In this context, we really welcome this book Design and Use of Software Architectures, which represents a significant
step forward in this discipline. Jan Bosch actually contributes to all three aspects - representation, process, and assets
- with a good mix between an academic perspective and an industrial perspective.

Most of Jan Bosch's contribution is on the second point, though: the architectural design process, which is certainly the
topic closest to my interests and daily concerns. Method and process are definitely areas where we can say there are
many ways to achieve a goal, but at the same time areas where too little has been published yet to allow comparison and
evolution. Here Jan Bosch does an excellent job of dishing out engineering wisdom to people who have to design
architectures for families of related products, or product lines. This area is especially tough because it goes far beyond
the difficulty of getting the architecture right for one system - not an easy undertaking alone. It's compounded with the
challenge of getting it right for several systems, many as yet unspecified and, therefore, projecting oneself far

into the
future, often widely across an organization or even several organizations. The approach Jan develops takes the
engineering aspects very seriously as it dedicates a lot of attention to the assessment of the qualities of the architecture,
and even driving some design aspects from these qualities.

Many challenges still lie before us in this rather young field of software architecture. One can easily say that there is a lot
more we still don't know and can't achieve yet in this domain than what we do and can. This book is an important
contribution, a rock placed on the cairn, that one cannot remove nor dismiss easily. I hope that you will enjoy it and learn
from it as much as I have. It is a big book, and fortunately there are several paths through it. It is not a boring theoretical
manual, as it remains hooked into reality, and is well illustrated with real-life examples - something not very easy to do
in the architectural domain; all too often the examples presented on architecture are either too small or too trivial. After
all, who needs a blueprint for a doghouse?

This book has also convinced me that we are slowly, but surely, coming to an era where we will assemble systems one
component at a time rather than program them again and again, one line at a time, just changing programming language
every five years.

Happy reading!

Philippe Kruchten, P. Eng.

Director of Process Development

Rational Software Canada, Vancouver, B.C.

0201674947P04062001

Users Review

From reader reviews:

Pearlie Henry:

What do you consider book? It is just for students because they're still students or it for all people in the world, the particular best subject for that? Only you can be answered for that concern above. Every person

has various personality and hobby for each other. Don't to be forced someone or something that they don't want do that. You must know how great and also important the book Design and Use of Software Architectures: Adopting and Evolving a Product-Line Approach. All type of book would you see on many resources. You can look for the internet methods or other social media.

Jeffrey Paolucci:

In this 21st centuries, people become competitive in most way. By being competitive now, people have do something to make these survives, being in the middle of often the crowded place and notice through surrounding. One thing that oftentimes many people have underestimated the idea for a while is reading. Sure, by reading a reserve your ability to survive raise then having chance to endure than other is high. For yourself who want to start reading a new book, we give you this specific Design and Use of Software Architectures: Adopting and Evolving a Product-Line Approach book as basic and daily reading reserve. Why, because this book is usually more than just a book.

Jason Valladares:

A lot of people always spent their free time to vacation or even go to the outside with them family members or their friend. Are you aware? Many a lot of people spent they free time just watching TV, or playing video games all day long. If you would like try to find a new activity that is look different you can read a book. It is really fun for you. If you enjoy the book that you read you can spent 24 hours a day to reading a e-book. The book Design and Use of Software Architectures: Adopting and Evolving a Product-Line Approach it is very good to read. There are a lot of people that recommended this book. These were enjoying reading this book. In case you did not have enough space to develop this book you can buy the particular e-book. You can m0ore very easily to read this book through your smart phone. The price is not to fund but this book has high quality.

Michael Mantz:

Your reading sixth sense will not betray you actually, why because this Design and Use of Software Architectures: Adopting and Evolving a Product-Line Approach e-book written by well-known writer we are excited for well how to make book that may be understand by anyone who also read the book. Written throughout good manner for you, still dripping wet every ideas and publishing skill only for eliminate your hunger then you still question Design and Use of Software Architectures: Adopting and Evolving a Product-Line Approach as good book not just by the cover but also from the content. This is one e-book that can break don't ascertain book by its deal with, so do you still needing another sixth sense to pick this!? Oh come on your examining sixth sense already told you so why you have to listening to another sixth sense.

Jan Bosch #AW1VUM0IX39

Read Design and Use of Software Architectures: Adopting and Evolving a Product-Line Approach By Jan Bosch for online ebook

Design and Use of Software Architectures: Adopting and Evolving a Product-Line Approach By Jan Bosch 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 Design and Use of Software Architectures: Adopting and Evolving a Product-Line Approach By Jan Bosch books to read online.

Online Design and Use of Software Architectures: Adopting and Evolving a Product-Line Approach By Jan Bosch ebook PDF download

Design and Use of Software Architectures: Adopting and Evolving a Product-Line Approach By Jan Bosch Doc

Design and Use of Software Architectures: Adopting and Evolving a Product-Line Approach By Jan Bosch Mobipocket

Design and Use of Software Architectures: Adopting and Evolving a Product-Line Approach By Jan Bosch EPub