Software Architecture

Oliver Vogel • Ingo Arnold • Arif Chughtai
Timo Kehrer

Software Architecture

A Comprehensive Framework and Guide
for Practitioners

Springer

All Authors
authors@software-architecture-book.org

Oliver Vogel
oliver.vogel@software-architecture-book.
org

Ingo Arnold
ingo.arnold@software-architecture-book.
org

Arif Chughtai
arif.chughtai@software-architecture-
book.org

Timo Kehrer
timo.kehrer@software-architecture-book.
org

Translator
Tracey Duffy
TSD Translations
TraceyDuffy@tsdtranslations.org

Copyright © 2009 by Spektrum Akademischer Verlag, Heidelberg, Germany.
Title of the German original: Software-Architektur. Grundlagen - Konzepte - Praxis
ISBN: 978-3-8274-1933-0
All rights reserved.

ISBN 978-3-642-19735-2 e-ISBN 978-3-642-19736-9
DOI 10.1007/978-3-642-19736-9
Springer New York Dordrecht Heidelberg London

ACM Codes: D.2, K.6

Library of Congress Control Number: 2011933921

Cover Design Editor: KünkelLopka GmbH

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Foreword

“The architect should be equipped with knowledge of many branches of study and varied kinds of learning, for it is by his judgement that all work done by the other arts is put to test.” Thus opens Chapter I in Marcus Vitruvius Pollio’s seminal text, “*The Ten Books on Architecture*” [1]. Readers unfamiliar with Vitruvius’ work may find it surprising to learn that it was published in the first century B.C., long before anyone even dreamt of such a thing as software. It is, in fact, the oldest known engineering text. Yet, this nugget of wisdom from a two thousand-year-old text resonates fully today, spanning the full continuum of technological evolution and the growth of engineering knowledge up to our modern world of software.

Vitruvius goes on: “This knowledge is the child of practice and theory” and, yet further, “[i]t follows, therefore, that architects who have aimed at acquiring manual skill without scholarship have never been able to reach a position of authority to correspond to their pains, while those who relied only upon theories and scholarship were obviously hunting the shadow, not the substance.”

I cannot think of more apt set of quotations for introducing this book on software architecture. Architects, as Vitruvius tells us, must possess not only the requisite technical knowledge of their domain (i.e., the theory), but they must *understand* it, and, true understanding comes only with direct experience (i.e., practice). Moreover, architects must take a broad perspective that encompasses many varied facets of the systems they are designing: more than just the technical issues and solutions, but also the social, economic, and even psychological factors that are at play. It has been my experience in close to forty years of industrial software development that the primary difference between a competent software architect and a skilled software developer is that architects see beyond the technology. Architects perceive a software system not as a Java or C program or even as software, but as an integral part of a greater system that serves a particular business or technical purpose. Consequently, good software architects are individuals who care deeply about the system and recognize the value that it provides, which means that in the process of design they must learn to become domain experts, but ones distinguished by a deep understanding of computing technology and its capabilities.

The authors of this book are fully cognisant of what makes a true software architect—based on their long-term experience as practitioners. They teach us not only about the fundamental technical tricks of the trade (WITH WHAT) but also the equally important aspects (WHAT, WHERE, WHY, and WHO) and, last but not least, HOW all of these can be combined to produce a software design that hits the sweet spot. In this, the book distinguishes itself from numerous other books on software architecture—it covers the full spectrum of concerns facing an architect.

I think that we are fortunate to finally have such a comprehensive treatment of the topic at our disposal. For practicing architects, this book can serve as a handy reference—a convenient reminder and check list. For aspiring software architects, it will expose and demystify some of the less well-known but crucial aspects involved in the architectural practice and, perhaps, help identify the gaps they may need to fill to become *bona fide* heirs of Vitruvius' long-standing legacy of engineering excellence.

Bran Selic
Malina Software Corp., Ottawa, Canada

Reference

[1] Vitruvius, "The Ten Books on Architecture," (translated by Morris Hicky Morgan), Dover Publications, Inc., New York, 1960.

Foreword

For many years now I have been leading the *IT Architect Profession* program at IBM in Europe. It is my job to support the development of IT architects and to ensure that they keep their knowledge up-to-date. Increasing numbers of customers and competitors are interested in building up their own architecture skills. The Open Group, a technology-independent and provider-independent consortium, has been offering the *Open Group Information Technology Architect Certification Program* since 2006. Many of our customers and competitors already use it to evaluate the qualifications of their employees.

Architecture skills are becoming increasingly important

In this context I am excited about the new edition of this book. It describes and explains very clearly and in a well-structured way what architects of IT systems do and what IT or software architecture is all about. The book therefore offers a good basis for familiarizing yourself with the topic and improving your architecture skills. It fits perfectly with the current trend that I see both in The Open Group and with our customers and competitors. It reflects the way of thinking that we have been promoting and demanding for many years at IBM.

This book helps you to build up and expand these skills

It is a very good time for IT architects. The trends in IT and technology are developing ever further and ever faster. A software architecture as the basis for the development of IT systems has become increasingly important in dealing with these rapid changes. Not least the whole discussion around the topic of service-oriented architecture (SOA) has made that more than clear.

The time is ripe to get into this exciting topic…

I can therefore highly recommend this book for anyone who has recognized the necessity of dealing with the topic of software architecture. It provides a comprehensive starting point for conscious architectural thinking.

…and develop an architectural awareness

Karin Dürmeyer
IBM Distinguished Engineer
IBM IOT Northeast IT Architect Profession Leader

Preface

As a term, architecture is ever-present …

In everyday IT work, the term “software architecture,” or “architecture” in general, has become ever-present, and due to its enormous relevance for project success, can no longer be ignored. Business cards show job titles such as Security Architect, Data Architect, System Architect, or even Enterprise Architect. We create documents with the title “Solution architecture” for customers, for example, or customers themselves request architecture from suppliers. Although the term “architecture” is used so frequently, on closer inspection, it is clear that architects, project leaders, developers, and other stakeholders do not share a common understanding of the term.

… and interpreted in lots of different ways …

For some of us, “architecture” is the selection and use of a technology; for others, “architecture” is a process; for many, “the architecture” is a folder with drawings containing geometrical figures connected to one another; for others again, “architecture” may be everything that “the architect” produces—whatever this may be. In its practical use, the term “architecture” covers quite a broad scope—that is, it is not defined or understood uniformly. This often makes it difficult for several people to work together and communicate efficiently in the architecture domain and in daily working life.

… initially even within the team of authors

When we decided to write a book about software architecture some years ago, we started our project by initially taking stock. We quickly learned that even within a strictly limited group of experienced software architects, it was not as easy to clearly define software architecture itself as we had expected. We realized that, even though we all had years of experience in designing, describing, or verifying software architectures, we did not have a uniform, precise understanding of the architecture domain.

Our desire for an architecture framework …

We became more and more aware of how important it was to develop a common understanding and vocabulary. An architecture framework that establishes a common, uniform terminology would allow us to look at and explain the architecture topic discriminatingly. This type of holistic framework was something we had always been looking for in our professional careers.

… and orientation

We looked back to the time when we ourselves were primarily software developers and were confronted with the term “software architecture” for the first time. At this point in time, “software architecture” was a very abstract term for us, and it was difficult for us to really grasp what it meant. There was no intuitive architecture framework available that would have enabled us to understand this important field of topics. Theory and practice concentrated primarily on individual aspects of architecture and did not allow a holistic understanding. We therefore tried to find order amongst the architecture chaos ourselves. For a long time, we had all been subconsciously or intuitively looking for a framework that covers the important dimensions of the architecture domain. At the beginning of our journey through the IT world, we needed a lot of technical and detailed knowledge. We therefore concentrated on acquiring knowledge about techniques and technologies, process models, methods, and organizations. In the course of our professional life and thus throughout our educational journey, each one of us, constantly and partly without being aware of it, derived *his* understanding of the architecture domain from this collection of isolated individual insights. With this book project, we had finally arrived at the point where we could reconcile our individual understandings, bring them together to formulate a common understanding, and make this the core of our book.

Our architectural thinking developed over time

We all knew that there is *no one* architecture examination that gives *the one* architecture certificate that you can pass or acquire in order to then be able to call yourself a certified architect. In the course of our lives as computer scientists, we had all already worked in lots of roles. As analysts, software developers, testers, project leaders, designers, or enterprise architects, we knew that architecture has many faces and that the architecture aspect is decisively important for many roles—not solely for the role of the architect. Our experience was also that, in addition to further technical education, we first had to gather sufficient practical experience before we could start to think “architecturally.”

Our book vision

The primary goal of our book is to give readers orientation in the architecture domain. In our view, many books about architecture focus too heavily on the topic of technology. Other books concentrate on architecture documentation and nomenclatures and their related techniques. Some other books look at solution patterns for architecture problems. And finally, relevant computer magazines regularly cover reports on project experiences in which the architecture aspect of a solution presented is very often the factor that gives the article its substance. However, in our opinion at least, hardly any of these works attempt to give the reader a comprehensive orientation in the topic of architecture. Most of the books we know concentrate only on selected sub-areas of architecture. And the few books that cover architecture more broadly still lack more or less a thorough structure that provides orientation, or rather, a book architecture.

Our challenges

We thus faced two great challenges. The first challenge was to design a book structure that addressed the aspects of orientation, theory, and practice—for us, all of these aspects are equally important. Our second challenge was to develop and describe a software architecture model that then allowed us to work through the multi-dimensional nature of this topic appropriately and to use it as a stable core for our book. The result of this initial and fundamental work was the architecture of the book itself. We describe this in detail in Chap. 2 and it is structured as follows:

> Explanation of the architecture dimensions (e.g., requirements in the context of architecture) based on a holistic architecture framework.
> Presentation of the parts of the individual architecture dimensions relevant in practice.
> Practical application of the architecture contents covered in the book.

Our book

This book is thus the result of our desire for a work that structures the topics around architecture sensibly, is based on practice, and that conveys corresponding practical experience. In particular, the book is independent of any specific technology and is timeless. For us therefore, this book belongs to that group of fundamental works that provides you with a stable and future-proof reference system that goes beyond current technological trends. The task that we set ourselves with writing this book was not easy—it required all of the authors to look at the topic of architecture intensively and in great depth beyond the otherwise usual level of considering different aspects in isolation. In the time in which we produced this book, we learned a lot. We discussed and debated with one another. As a result of working together on this book, we gained a lot of new and valuable knowledge and a common understanding of architecture.

You now hold our understanding of architecture in your hands. We hope that our claim of arranging and explaining the topic of architecture for you, and anchoring it in practical examples, will help you in your dealings with this interesting and important area of your working life or your studies.

History of the book and the English version

The first edition of this book appeared on the German-speaking market in autumn 2005. In our view, the great success that the first edition enjoyed was connected to the fact that at this time, conceptual, planning, educational, or organizational contributions in IT had gained importance to the extent that specialized technical knowledge was outsourced to countries with pay structures and an expert basis that further encouraged this trend. From then on, the role of the architect, with its holistic and integrative view of the IT challenges, formed the spearhead of a new generation of training profiles within computer science and neighboring domains. This had a corresponding positive effect on the sales of our fundamental work.

The high demand for the first edition of our book meant that we were able to offer our German-speaking readers a revised and updated second edition of the book in 2008.

In the meantime, we received numerous requests from non-German-speaking colleagues to provide an English translation of our book. All of the authors work in an international, primarily English-speaking environment, and, thanks to presentations at IT conferences or university contacts, have regular exchanges with English-speaking colleagues. We therefore quickly agreed when we received a request from Springer for a further revised version of our book—this time in English. We used the opportunity of producing an English translation to improve the contents further based on reader feedback, our practical experience, and current IT developments, such as cloud computing.

Although the translation and the repeated revision of this third edition cost our translator and us as authors many hours of our free time, we are all happy that we took advantage of this opportunity. In particular, we are delighted to finally be able to offer our book to a global audience.

Our thanks

At this point we would like to thank everyone who gave us the freedom to work on this project and who supported us. This includes our partners and children, our friends and colleagues, our employers and superiors. We would like to thank all of those who gave up their time for us and constantly gave us new strength.

Our sincere thanks also go to our translator, Tracey Duffy. With her extremely professional and team-oriented approach and her great talent for technical translation, she provided us with continuous support in realizing this translation project. Her assistance enabled us to meet our high quality standards, and to do so highly efficiently and right on schedule.

Finally, we would like to thank Ralf Gerstner at Springer, who provided us with continuous and professional support in producing this third edition of our book, and who did so with great patience.

Contents

About the Authors

Oliver Vogel is a certified Enterprise and IT Architect with IBM Switzerland. He leads, coaches, and acts as consultant for international projects in architecture topics such as architectural design, implementation, evaluation, and governance. He is also the worldwide IBM Enterprise Architecture Education leader.

Ingo Arnold is a Global Enterprise Architect with Novartis, Switzerland. In addition to his role at Novartis, Ingo is an Associate Professor at the Universities of Basel, Switzerland, and Lörrach, Germany. He is also a well-known speaker at IT conferences, where he holds presentations on topics such as SOA, IT Security, and IT Governance for international audiences.

Arif Chughtai has been a successful freelance IT consultant and IT trainer for more than 10 years. His specialist fields include software architecture, service-oriented architectures, object-oriented software development, and model-driven software development. He regularly shares his expert knowledge in lectures, presentations, and technical articles.

Timo Kehrer is a scientific employee at the Software Engineering Group of the University of Siegen, Germany. He is currently researching model-based software development, model comparison, model version management, and model evolution.