Modernizing Legacy Systems: Software Technologies, Engineering Processes and Business Practices


This book isn’t likely to become the most popular technical book of the year but for the material covered may become one of the most enduringly useful.

The target audience is primarily system architects for large and complex information systems in transition from nearly obsolete technologies to more contemporary solutions. The case study, analysis and recommendations cover a variety of modern technologies (XML, Java, etc.) but component developers and sub-system engineers will find little to work with. Non-technical project managers should also look elsewhere. The focus is on architectural, integration and migration challenges. The authors do not assume any particular technical skills from the reader but general knowledge of requirements management, systems analysis and design, testing and software development are necessary to understand most of the material covered.

The first three chapters cover material essential for those to follow including the case study, the modernization approach, and some core risk management principles. The case study sets the scale of challenges to be addressed - 90 processing locations, 25 thousand users, 1.8 million lines of code and millions of dollars in annual operating costs. Subsequent chapters are organized according to the activities in the modernization approach and each begins with a “where are we” section to keep new material in context. The modernization approach on its own is a useful blueprint for other projects but is fairly general so the topics covered within this framework should apply to most modernization efforts. Themes are developed throughout the book and it makes a great read from beginning to end but most chapters can be read independently. Early chapters deal with the general topics such as the business case for modernization and gaining insight into legacy systems. Middle chapters deal with architectural issues such as transaction management and system integration. Finally, there is coverage of specific topics such as code/data migration, evaluating architectural alternatives and how to plan the modernization effort.

Discussion of specific products and technologies is included but serves primarily to illustrate principles, techniques and recommendations that would apply to a majority of modernization efforts. For example, the chapter on analysis of alternatives covers solutions based upon Solaris, Oracle and BEA Tuxedo but the real value is as an illustrative example of evaluating architectural alternatives.

About a third of the chapters are written with additional authors but the case study is followed throughout, the modernization approach used consistently to organize ideas and conceptual integrity of the main themes sustained across chapters. The editing of the text across chapters is very good. The diagrams are one of the few areas that could be improved. While diagrams are plentiful and generally informative, the notation used varies from section to section and the mix of styles may be confusing. For example, the sequence diagrams do not consistently include focus of control and it isn’t always clear if arrow styles convey synchronicity. Some diagrams use the Unified Modeling Language (UML), some do not, and others use a mix of UML and non-UML notations – the last the most likely to confuse.

Two of the topics addressed are particularly valuable – architectural patterns and transaction management. The architectural patterns discussed include the general capabilities that are required of any enterprise system – external data access, reporting, ad hoc queries, transaction processing, etc. Each is discussed with regard to its impact on the system architecture and a description of the chosen, modernized, solution. This would be an invaluable template for most projects – the specific solution described may or may not suit your project requirement but the motivations for each pattern are nearly universal. The evolution of the as-is to the to-be architecture for these patterns is one of the themes that emerges throughout the book. The material devoted to transaction processing is also very strong and covers this issue at several levels of abstractions and from several perspectives. Sections are devoted to transaction characterization, managing transaction processing code and data, distribution, messaging, and data migration. Each section uses the case study as an illustrative, concrete example and draws from these examples techniques and principles of general applicability.

Modernization efforts often require years of effort and cost millions of dollars. This book provides informative discussion, valuable insights, and convincing recommendations. It merits a careful read by almost any systems architect working on a legacy modernization project. The team at the SEI has once again put forward the essentials without which so many projects will fail and many millions of dollars will be wasted.